Robot trajectory planning using OLP and structured light 3D machine vision

RODRIGUES, Marcos <http://orcid.org/0000-0002-6083-1303>, KORMANN, Mariza, SCHUHLER, C and TOMEK, P

Citation:

Copyright and re-use policy

Robot Trajectory Planning using OLP and Structured Light 3D Machine Vision

M. Rodrigues[1], M. Kormann[1], C. Schuhler[2], and P. Tomek[3]

[1] Sheffield Hallam University, Sheffield, UK
[2] TWI – The Welding Institute, Cambridge, UK
[3] MFKK Invention and Research Services Center Ltd, Hungary

Abstract. This paper proposes a new methodology for robotic offline programming (OLP) addressing the issue of automatic program generation directly from 3D CAD models and verification through online 3D reconstruction. Limitations of current OLP include manufacturing tolerances between CAD and workpieces and inaccuracies in workpiece placement and modelled work cell. These issues are addressed and demonstrated through surface scanning, registration, and global and local error estimation. The method allows the robot to adjust the welding path designed from the CAD model to the actual workpiece. Alternatively, for non-repetitive tasks and where a CAD model is not available, it is possible to interactively define the path online over the scanned surface.

1 Introduction

Welding represents one of the single largest applications of robots in manufacturing engineering, as approximately a quarter of all industrial robots are being used in connection to welding tasks [1]. The development of flexible automation systems that can be set up quickly and switched over to another product line are essential to increase productivity and profitability while maintaining product quality within prescribed tolerances. The challenge is that small and medium sized enterprises (SMEs) normally do not have the resources to invest in expensive technologies and extensive human training [2]. In particular, robot programming is a demanding specialised activity that, for non-repetitive tasks, can take several hundred times longer than the actual robot execution time [3].

There are two methods of robot programming, namely online and offline (OLP) programming. Online is normally carried out by skilled operators guiding the robot through a sequence of locations in space [4]. Although conceptually simple, for complex geometries it becomes difficult, very tedious and time consuming. Attempts have been made to improve online programming by the addition of sensors and additional calibration [5,6]. However, the process has to be repeated again for a workpiece with a slight different design. Despite these issues, online programming is the programming of choice for most SMEs.

OLP methods utilise 3D CAD data to generate and test robot programs and are widely used in automation systems with large product volumes [3]. Once the workpiece and robot cell are modelled, the operator can simulate the

program and test for collisions. The programs are then downloaded to the robot for execution. The main advantage is that OLP does not require the actual robot so it does not adversely affect utilisation time. Some limitations and open issues of OLP [7–9] can be summarised as follows:

– manufacturing tolerances between real and ideal CAD workpieces,
– inaccurate placement of the workpiece within the robot cell,
– inaccuracies between physical and modelled work cell,
– thermal effects during welding,
– lack of a methodology to deal with complex features.

The MARWIN project [10] develops a cognitive welding robot interface where welding tasks and parameters are intuitively selected by the end-user directly from a library of CAD models. No knowledge of robot programming is required, as robot trajectories are automatically calculated from the CAD models and validated through fast 3D scanning of the welding scene. The role of the user is limited to high level specification of the welding task and to the confirmation and/or changing of welding parameters and sequences as suggested by the control program. MARWIN uses a 3D structured light scanner where the light source and camera are in a parallel arrangement; its development and mathematical formulation have been described in [2, 11].

The focus of this paper is on the problem of automatic program generation in OLP as this is a perceived gap and a requirement, as no system exist in the market which implements the complete offline programming chain, although many separate *ad hoc* solutions exist [3]. The approach is to incorporate a fast area 3D scanner and propose a methodology for OLP that addresses most of the issues above, and can include pre- and post-verification of welding quality. Furthermore, the proposed method is also suitable for non-repetitive workpieces (for which CAD models may or may not be available) through a combination of sensor-guided online and offline programming.

2   Methodology

The method described here deals with interactive definition of control points defining a robotic welding path including the tooltip orientation and its translation to the actual workpiece – which may be slightly different from its ideal CAD model. This adaptive translation is the method's main novelty as it can deal with uncertainties between CAD descriptions and real world workpieces in the robotic cell. Most of the limitations highlighted in the previous section are thus addressed in the following steps.

1. **CAD model generation**. Here it is assumed that a CAD model is available and can be loaded into the 3D modelling environment.
2. **Control points and tag creation**. This involves the definition of robot position tags from 3D CAD data with specific tool centre point. This paper proposes an interactive 3D method in which the user selects the path control

points directly on the model surface. For each selected control point, the solution will automatically generate the approach and retreat locations and the tooltip orientation.

3. **3D surface scanning**. A fast structured light 3D reconstruction method is used to scan workpiece surfaces that include the welding path. There is no need to scan the entire workpiece.

4. **3D registration with CAD models and error checking**. Automatic registration of CAD and scanned surface is performed based on point visibility constraints. A global and local root mean square error (RMSE) measures between CAD model and their nearest points on the scanned surface give an indication to the user whether welding should proceed or not.

5. **Translation of control points from CAD to scanned surface**. If the global and local RMSEs are within set thresholds, each control point defined on the CAD model is translated to the scanned surface. This will guarantee that the welding path will be adapted to the actual workpiece thus, minimising path uncertainty.

6. **Trajectory planning**. The inverse kinematics of industrial robots usually yields multiple solutions in Cartesian space. Here, the control points together with derived approach, retreat and orientation information from step 2 are used to generate a unique solution. This step can easily be achieved by standard OLP software from the robot manufacturers. In the MARWIN project, this is achieved using the ABB RobotStudio which can deal with issues such as reachability, transitions, collision avoidance, and so on.

Steps 1 and 6 are outside the scope of this paper; in what follows we describe and demonstrate steps 2–5.
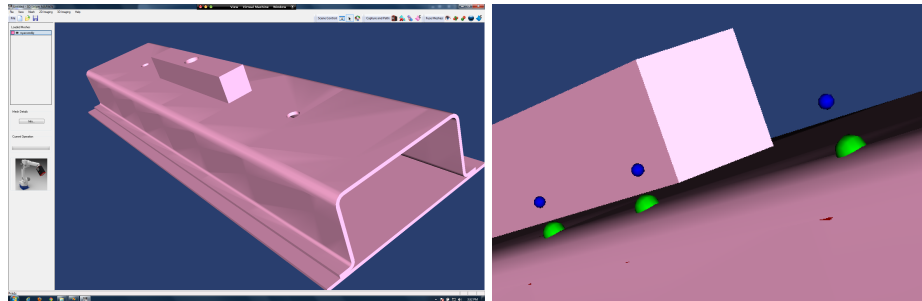
3   Control Points and Tag Creation

Fig. 1. Left: the 3D Striper MARWIN interface with a loaded CAD model. Right, welding control points selected through mouse clicks.

To demonstrate the concepts, an interactive modelling application named *3D Striper MARWIN* has been developed, whose interface is shown in Fig. 1

(left) with a loaded CAD model. In order to define a sequence of points for a welding path including approach points and tooltip orientation, an aligned bounding box with the (x, y, z) axes is estimated. This is simply defined by 6 bounds $(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$. When the user marks a point on the surface of the model, a ray-tracing algorithm is used to find the intersection with the bounding box and the intersection with the mesh. This is a problem of line-plane intersection where it is assumed that the line has a starting point S and direction $\mathbf{c}$. The intersection line is given by

$$L(t) = S + \mathbf{c}t \tag{1}$$

The solution only involves finding the intersection point with the generic plane [12]. The generic plane is the xy-plane or $z = 0$. The line $S + \mathbf{c}t$ intersects the generic plane when $S_z + \mathbf{c}_z t_i = 0$ where t_i is t "intersection":

$$t_i = -S_z/c_z \tag{2}$$

From equation (2), the line hits the plane at point $\mathbf{p}_i$:

$$\mathbf{p}_i = S - \mathbf{c}(S_z/c_z) \tag{3}$$

Thus, for every ray, i.e. for every point marked on the surface of the model by the user through a mouse click in Fig. 2 (right) two points are obtained $\mathbf{p}_1$ (blue sphere) and $\mathbf{p}_2$ (green). These points define the welding sequence where $\mathbf{p}_1$ is the intersection with the bounding box and $\mathbf{p}_2$ is the intersection with the mesh. Note that it is unlikely that the intersection on the mesh will rest on a vertex. More likely, it will intersect on a polygon's face somewhere between vertices. A good approximation then is to find the three vertices on the mesh that are the nearest to the intersection line. Such vertices define a plane and it then becomes straightforward to determine the exact intersection point through Equation (3).

The tooltip alignment is determined by Euler's theorem [12]. Defining vectors $\mathbf{u}$ and $\mathbf{q}$ as

$$\mathbf{u} = \mathbf{p}_2^k - \mathbf{p}_1^k, \quad \mathbf{q} = \mathbf{p}_1^{k+1} - \mathbf{p}_1^k \tag{4}$$

where k is the index of each pair of points $(\mathbf{p}_1, \mathbf{p}_2)$. The desired alignment is that the tooltip x-axis is aligned with vector $\mathbf{u}$ and e y-axis is aligned with $\mathbf{q}$. To perform the alignment, the required rotation is decomposed into a sequence of known steps:

1. Perform two rotations around the y and z axes by angles θ and ϕ so that the x-axis becomes aligned with $\mathbf{u}$.
2. Perform a z-roll of angle β around the newly rotated x-axis such that the y-axis becomes aligned with $\mathbf{q}$.

The above transformation requires the multiplication of 3 matrices:

$$\mathbf{R_u}(\beta) = \mathbf{R}_y(-\theta)\mathbf{R}_z(\phi)\mathbf{R}_x(\beta) \tag{5}$$

Figure 2 shows the result of such alignment where the rotated x-axis is shown in yellow, the y-axis in red, and the z-axis in green. The approach (and retreat)
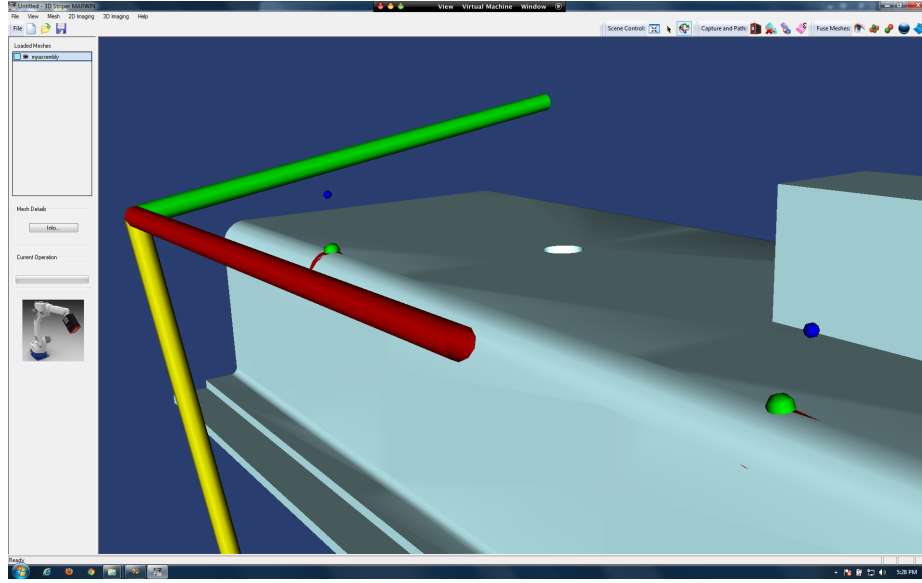
Fig. 2. Euler's theorem is used to determine the rotation matrix for tooltip alignment.

position is $\mathbf{p}_1$ and the welding position is $\mathbf{p}_2$ which are achieved by a rotation around $\mathbf{u}$ defined by equation (5) followed by a translation $\mathbf{t}$ from $\mathbf{p}$ (any position in the work space) to $\mathbf{p}_1$ and $\mathbf{p}_2$:

$$\mathbf{p}_i = \mathbf{R}\mathbf{p} + \mathbf{t}, \quad \text{where} \quad i = 1, 2. \tag{6}$$

The approach path from a generic position $\mathbf{p}$ to $\mathbf{p}_1$ can be seen as a midpoint approach while from $\mathbf{p}_1$ to $\mathbf{p}_2$ is the final positioning of the tooltip.

4   3D Surface Scanning

The principle of operation of structure light scanning is to project patterns of light onto the target surface whose image is recorded by a camera [13]. The shape of the captured pattern is combined with the spatial relationship between the light source and the camera, to determine the 3D position of the surface along the pattern. The main advantages of the method are speed and accuracy; a surface can be scanned from a single 2D image and processed into 3D in 40ms [14, 15].

The expressions to compute the coordinates (x, y, z) of a surface point from a pixel location (v, h) on stripe n (mapping to a point on the surface of the scanned object) is defined as [2, 11]:

$$x = D_p - \frac{D_p D_s}{vPD_p + W_n}, \quad y = \frac{hPD_p D_s}{vPD_p + W_n}, \quad z = \frac{W_n D_s}{vPD_p + W_n} \tag{7}$$

where D_s is the constant vertical distance between the camera and the projector (Fig. 3, left), D_p is the constant distance between the projector and the system origin (or calibration plane), W is the constant width between successive light planes (or stripes) in the calibration plane, P is the pixel size in the sensor plane of the camera. The mathematical formulation of such arrangement is simpler than of those of standard scanners which results in less computing cycles thus, making the parallel design appropriate for 3D real-time processing.
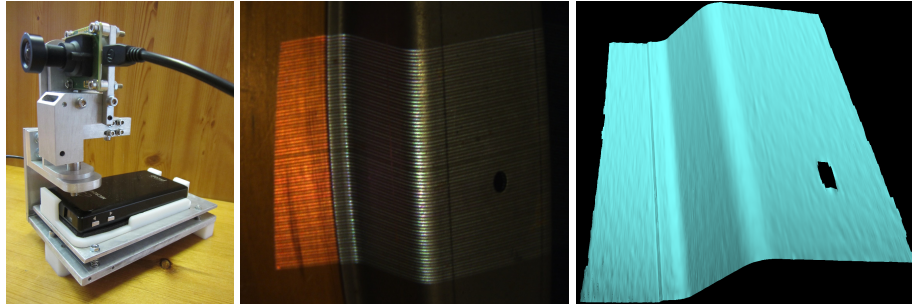

Fig. 3. Left: GMPR scanner; middle: 2D image; right: 3D reconstructed surface.

Fig. 3 depicts the GMPR scanner developed by the MARWIN project which is capable of processing a single 2D image (Fig.3 middle) into a surface patch (right). The scanner is attached to the robot tooltip such that each patch is incrementally registered until the desired workpiece surface is fully scanned. All scanned surfaces are thus, described in robot coordinate system.

5   3D Registration with CAD Models and Error Checking

In order to ensure correct calculation of robot trajectories based on the scanned surface, it is necessary to register the CAD model of the welding assembly to its scanned 3D model. This is to verify whether or not the scanned scene matches its CAD description and, if so, translate the control points from the CAD to the scanned surface. It is stressed that only translated points to the scanned surface will be used for trajectory calculation. The ICP (Iterative Closest Point) estimation algorithm [16] is used with the additional constraint of point visibility. The closest points in the ICP are found by calculating the Euclidean distances between a point $\mathbf{p}$ in the first frame (the CAD model) and a point $\mathbf{q}$ in the second frame (the scanned surface S) given by

$$d(\mathbf{p}, S_k) = \min_{j \in (1, \dots, N_k)} d(\mathbf{p}, \mathbf{q}_{k,j}) \qquad (8)$$

Equation (8) means that every point in the CAD model needs to be checked against every point in the scanned surface. Once the closest points are estimated,

the two sets of points $\mathbf{p}_i$ and $\mathbf{q}_i$ are paired to each other. The registration goal is to estimate the parameters $(\mathbf{R}, \mathbf{t})$ rotation matrix and translation vector by minimising the following objective function:

$$F(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{m}\sum_{j=1}^{Ni} p_{i,j} d^2(\mathbf{R}p_{i,j} + \mathbf{t}, S_k) + \sum_{k=1}^{n}\sum_{l=1}^{Nk} q_{k,l} d^2(\mathbf{R}^T p_{k,l}^{'} - \mathbf{R}^T \mathbf{t}, S_i) \quad (9)$$

From the objective function in (9) the distance minimisation between the two sets of points is performed in a least squares sense:

$$f(\mathbf{R}, \mathbf{t}) = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{R}\mathbf{p}_i + \mathbf{t}, \mathbf{q}_i||^2 \quad (10)$$

When the transformation model $(\mathbf{R}, \mathbf{t})$ has been estimated, transform every point in the CAD model. This iteration is repeated until convergence to a minimum set threshold or when a predefined number of iterations is reached.
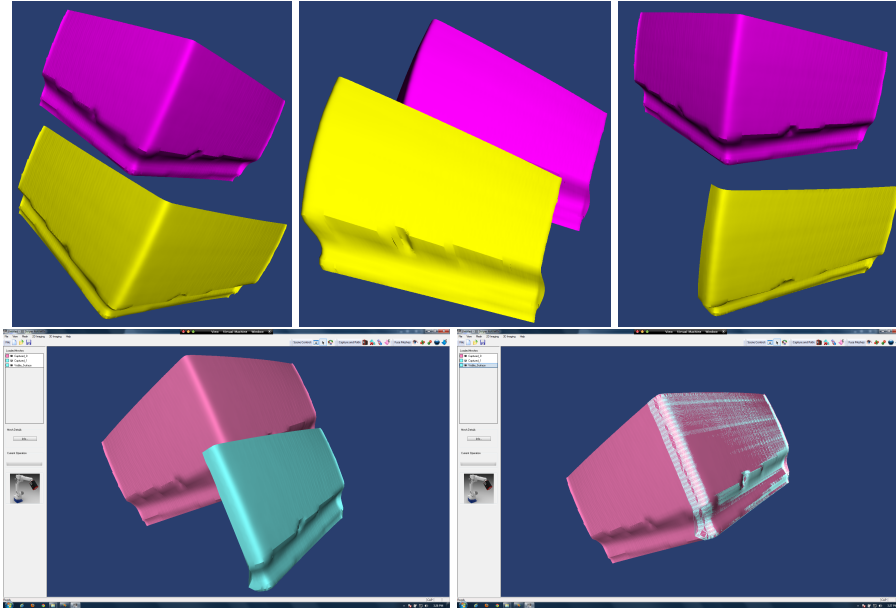

Fig. 4. Visibility constraints and registration. Top row (left and middle): model is reoriented to display desired surface; (right): hidden surfaces are removed from a selected model. Bottom row (left): initial position; (right): after registration.

The proposed visibility constraints are necessary for partial registration, as the ICP is guaranteed to fail if one tries to register both sets of data without adequate constraints. The method is that the user will pre-orient the CAD model

such that its visible surface is in a similar orientation as the scanned surface, which is always looking down the workpiece from the robot's coordinate x-axis. Once this step is completed, the method proposed here is the removal of hidden surfaces based on the depth buffer as depicted in Fig. 4 (top row). Each vertex $P = (P_x, P_y, P_z)$ of a face is available in the viewport as a scaled and shifted version of

$$(x, y, z) = \left(\frac{P_x}{-P_z}, \frac{P_y}{-P_z}, \frac{aP_z + b}{-P_z} \right) \tag{11}$$

The constants a and b are chosen such that the third component in (11) equals zero if P lies in the near plane and unity if P lies in the far plane. Successful registration is then performed with the visible and scanned surfaces, as shown in Fig. 4 (bottom row) through Equations (9) and (10).

6   Translation of Control Points from CAD to Scanned Surface

Upon registration convergence, each point $\mathbf{p}_2$ that was originally defined on the surface of the CAD model needs to be translated to the surface of the scanned model as depicted in Fig. 5. This is achieved by finding the intersection of the vector $\mathbf{u}$ defined by Equation (4) with the scanned mesh using Equations (2) and (3).
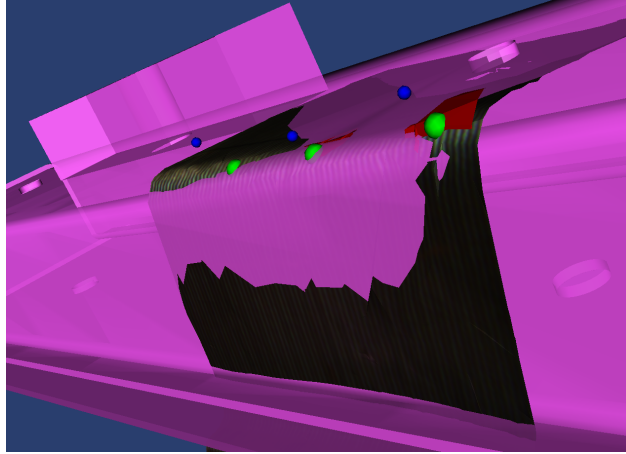

Fig. 5. Control points originally defined on the CAD model (magenta) are translated along the vector $\mathbf{u}$ to the surface of the scanned model.

In order to decide whether or not to proceed to the generation of the welding path, the root mean square error (RMSE) is calculated both globally and locally. The global RMSE considers all visible points and their nearest points on the

scanned mesh. Defining $\mathbf{p}$ a set of points in the CAD model and $\hat{\mathbf{p}}$ the nearest points in the scanned surface, the global RMSE is evaluated as:

$$\text{RMSE}(\hat{\mathbf{p}}) = \frac{1}{N} \sqrt{\sum_{k=1}^{N} (\hat{\mathbf{p}}_k - \mathbf{p}_k)^2} \tag{12}$$

where k is the number of points in the visible surface. The local RMSE only considers the control points lying on the path. Equation (12) is also used for local RMSE but in this case k is the index of the control points, $\mathbf{p}$ is the set of control points in the CAD model and $\hat{\mathbf{p}}$ the translated points along vector $\mathbf{u}$.

If both global and local RMSE are smaller than a set minimum threshold, then the set of control points and approach points are saved to an XML file. This information will be used by the ABB RobotStudio to generate the robot trajectory as described in Step 6 of the methodology in Section 2. Otherwise, an error condition is flagged to the user who may need to adjust the workpiece(s), proceed to a new scanning, or re-estimate visibility.

7   Conclusion

This paper has proposed and demonstrated a new methodology for robot offline programming in welding tasks using a combination of 3D scanning and 3D model manipulation through an interactive interface. The method is based on defining control points on a CAD model that represent the welding path and then verifying whether or not the CAD model matches the actual workpiece through online scanning and registration. Registration is optimised by defining visibility constraints. Two measures of discrepancy between CAD and scanned surface are performed based on global error and local path error. If the scanned surface matches its CAD model within given error thresholds, then specification of the welding task can proceed.

The ability to automatically calculate robot trajectories and welding sequences directly from CAD models and then verifying these online through 3D reconstruction satisfies a principal aim of the MARWIN project which is to provide a user-centred, cognitive system for robotic welding tasks. Furthermore, the proposed methods address open issues in OLP concerning discrepancies between CAD and manufactured workpieces, inaccurate placement of workpieces in the robot cell, and inaccuracies between physical and modelled work cell.

While the main focus of the paper has been to address OLP methodology, the techniques developed here are also suitable for online programming of non-repetitive workpieces where a CAD model may not be available. In this case, the user would select the control points directly on the scanned surface and proceed to generate robot trajectories. The next stage of the MARWIN project is to integrate the developed methods and software routines onto an OTC robot control system whose results will be reported in the near future.

References

[1]   European Commission Report (2003). "Smart, Sustainable Manufacturing: EU Research Sparks Revolution in Machine Tools, Robots and Automation", *Research and Innovation Report*, Brussels, 2003.

[2]   M. Rodrigues, M. Kormann, C. Shuhler, P. Tomek (2013). "An Intelligent Real Time 3D Vision System for Robotic Welding Tasks", *IEEE 9th Int Symposium on Mechatronics and its Applications* (ISMA13), Jordan, Amman, 9–11 April 2013.

[3]   Z. Pan, J. Polden, N. Larkin, S. van Duin, J. Norrish, (2012). "Recent progress on programming methods for industrial robots". *Robotics and Computer Integrated Manufacturing*, 28 (2), 87-94.

[4]   M.H. Junior, L. Wei, L.S. Yong (2000). "An industrial application of control of dynamic behaviour of robots – a walk-through programmed welding robot", *Proc 2000 IEEE Int Conf on Robotics and Automation*, San Francisco, CA, April 2000.

[5]   R.D. Schraft, C. Meyer (2006). "The need for an intuitive teaching method for small and medium enterprises", ISR-Robotik, Munich, Germany, 15–18 May 2006.

[6]   Z. Pan, H. Zhang (2007). "Robotic programming for manufacturing industry", *Proc Int Conf on Mechanical Eng and Mechanics*, Wuxi, China, 5–7 Nov 2007.

[7]   W. Dai, M. Kampker (2000). "User Oriented Integration of Sensor Operations in a Offline Programming System for Welding Robots", *Proc IEEE Conf on Robotics and Automation*, San Francisco, CA, April 2000.

[8]   A.M. Bi, S.Y.T Lang (2007). "A Framework for CAD- and Sensor-Based Robotic Coating Automation", *IEEE Transactions on Industrial Informatics*, Volume 3, Issue 1, Page(s):84 91, Feb. 2007.

[9]   P. Neto, N. Mendes and J. N. Pires (2010). "CAD-Based Robot Programming: the role of Fuzzy-PI Force Control in Unstructured Environments", *6th IEEE Conf on Automation Science and Engineering*, Toronto, Canada, 21–24 August 2010.

[10]   MARWIN: Decision Making and Augmented Reality Support for Automatic Welding Installations (2013). Project funded by the EC through Grant Agreement 286284 *Research for the Benefit of SMEs*. http://www.marwin-welding.eu/

[11]   M. Rodrigues, M. Kormann, C. Shuhler, P. Tomek (2013). "Structured Light Techniques for 3D Surface Reconstruction in Robotic Tasks", *Proc. 8th Int Conf on Computer Recognition Systems* (CORES 2013), Milkow, Poland, 27–29 May 2013.

[12]   F.S. Hill Jr (2001). *Computer Graphics Using OpenGL*, 2nd edition, Prentice-Hall Inc, 922pp.

[13]   A. Robinson, L. Alboul, M. Rodrigues, "Methods for Indexing Stripes in Uncoded Structured Light Scanning Systems", Journal of WSCG, Vol.12, No.1-3, ISSN 1213–6972.

[14]   M. Rodrigues, A. Robinson (2010). "Novel methods for real-time 3D facial recognition", In: SARRAFZADEH, Majid and PETRATOS, Panagiotis, (eds.) *Strategic Advantage of Computing Information Systems in Enterprise Management*. Athens, Greece, ATINER, 169–180.

[15]   M. Rodrigues, A. Robinson (2011). "Real-time 3D Face Recognition using Line Projection and Mesh Sampling", In: *EG 3DOR 2011 - Eurographics 2011 Workshop on 3D Object Retrieval*, Llandudno, UK, 10th April 2011, 9–16.

[16]   P. Besl and N. McKay (1992). "A method for Registration of 3-D Shapes", *IEEE Trans on Pattern Analysis and Machine Intelligence* (PAMI), 14(2):239–256.