

Robot Arms with 3D Vision Capabilities

Theodor Borangiu and Alexandru Dumitrache
*Politehnica University of Bucharest
Romania*

1. Introduction

The use of industrial robots in production started a rapid expansion in the 1980s, since they had the possibility of improving productivity, being able to work for extended periods of time with good repeatability, therefore making the quality of products stable. However, since the first robots worked “blindly”, on a pre-programmed trajectory, dedicated equipment had to be prepared only for supplying the workpieces to the robots (Inaba & Sakakibara, 2009). Also, human operators had to manually align the workpieces before the robot was able to manipulate them.

The *intelligent robots* appeared later in 2001 in order to solve this problem. An intelligent industrial robot is not a humanoid robot that walks and talks like a human, but one that performs complex tasks, similar to a skilled worker. This is achieved with sensors (vision, force, temperature etc) and artificial intelligence techniques.

Today, solutions to problems like picking parts placed randomly in a bin (*bin picking*), which were considered difficult a few years ago, are now considered mature: (Hardin, 2008) and (Iversen, 2006). A similar problem is *auto racking*, where robots have to pick parts which are presented one at a time, although the exact location and 3D orientation varies. These applications are made possible using 3D vision sensors.

1.1 3D vision sensors

Two types of vision sensors are used on the factory floor: two-dimensional (2D) and three-dimensional (3D). The 2D sensors are usually similar to a digital photographic camera, being able to capture an image of the workpiece and obtain the position and rotation angle of the object. This works well for parts that can sit on a flat surface, and enables them to be picked by a SCARA robot, for example.

There are two major methods for 3D vision sensors:

- Structured light
- Stereoscopic vision

1.2 Structured light sensors

Structured light sensors have a common principle: projecting a narrow band of light onto a three-dimensionally shaped surface produces a line of illumination that appears distorted when viewed from other perspectives than that of the projector. The shape of the line of

illumination can be captured by a 2D camera, allowing the exact geometric reconstruction of the surface shape using the *triangulation* method.

The simplest sensor using the triangulation principle is the *range finder*, which measures the distance to the closest reflective object (Fig. 1). A pulse of light (laser, regular visible light or infrared) is emitted and then reflected back on a linear CCD array. The position of the reflected light on the CCD array can be used to compute the distance to the closest object. This kind of sensor is also affordable to robot hobbyists (Palmisano, 2007).

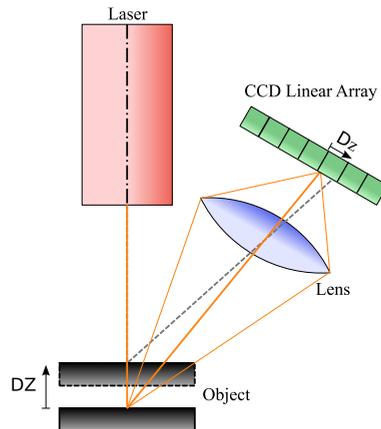


Fig. 1. Triangulation-based laser range finder

A more complex sensor is the *profile scanner*, which projects a narrow stripe of laser light onto the surface being digitized. A 2D camera, placed at a known angle with respect to the laser plane, records the image of the laser stripe and computes the local geometrical shape of the surface. For reconstructing a full 3D model of the workpiece, the profile scanner has to be swept around the part. The most precise way is to use a coordinate measuring machine (CMM). The sensors can also be mounted on industrial robots, which are more flexible in positioning and orienting the sensor, but also less accurate. Laser-based vision systems can generate very accurate 3D surface maps of the digitized parts, depending on the quality of the components used, but can be slow, since the sensor has to be moved continuously around the workpiece.

A faster method is projecting a pattern consisting of more light stripes. A typical setup for 3D measuring has a stripe projector, which is similar to a video projector, and at least one camera. Common setups include two cameras in opposite sides of the projector.

The depth can be reconstructed by analyzing the stripe patterns recorded by the camera, and several algorithms are available. The displacement of any single stripe can be converted into 3D coordinates; this involves identifying the stripes, either by tracing or counting stripes; ambiguities may appear when the workpiece contains sharp vertical walls. Another method involves alternating stripe patterns, resulting in binary sequences. Depth may also be computed by variations in the stripe width along the surface, and also by frequency / phase analysis of the stripe pattern by means of Fourier or Wavelet transforms. Practical implementation combine these methods in order to reconstruct a complete and unambiguous model of the surface.

1.3 Stereoscopic Vision

Stereoscopic vision attempts to compute the third dimension (the depth) in a way similar to the human brain. A 3D binocular stereo vision system uses two cameras which take images of the same scene from different positions (Fig. 2), and then computes the 3D coordinates for each pixel by comparing the parallax shifts between the two images.

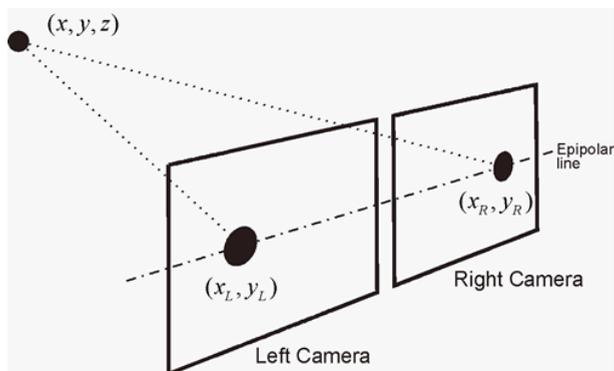


Fig. 2. Stereo vision principle: two cameras, which view the same scene, detect a common 3D point on different 2D locations

The main process in stereoscopic vision is the *stereo correspondence* between the two images, which is used to estimate *disparities* (differences in image locations of an object recorded by the two cameras) Calin & Roda (2007). The disparity is negatively correlated with the distance from the cameras: as the distance from the camera increases, the disparity decreases. Using geometry and algebra, the points that appear in the 2D stereo images can be mapped as coordinates in 3D space.

The stereo matching problem was, and continues to be, one of the most active research areas in computer vision. Several algorithms were developed; a classification and comparative benchmark for dense two-frame correspondence algorithms is presented in (Scharstein & Szeliski, 2002).

Aside from depth perception, stereo matching is also used in mobile robotics, where comparing two successive images taken with the same camera leads to estimation of the motion of the robot.

For robotic applications, which require 6-DOF part localization in 3D space, stereo vision is considered also a mature technology: (Hardin, 2008) and (Iversen, 2006). 3D part localization is also possible even with a single 2D camera, using a trained model of the part, and the approach is already used in production (Iversen, 2006). An experiment showing the integration of a binocular stereo vision system with an industrial robot is presented by Cheng & Chen (2008).

The main advantage of stereo vision techniques is that they do not require additional light sources, and therefore, these techniques are non-invasive with respect to the surrounding environment.

2. 3D reconstruction system

An application involving a 6-DOF robot arm and a profile scanning device is presented in this section. The structure of the system can be seen in Fig. 3(a). The main components are:

- Short range laser probe, able to measure distances between 100 and 200 mm with 30 μm accuracy, using one laser beam and two CCD cameras;
- 6-DOF vertical robot arm, with 650 mm reach and 20 μm repeatability;
- 1-DOF rotary table, for holding the workpiece being scanned;
- 4-axis CNC milling machine, for reproduction of the scanned parts.

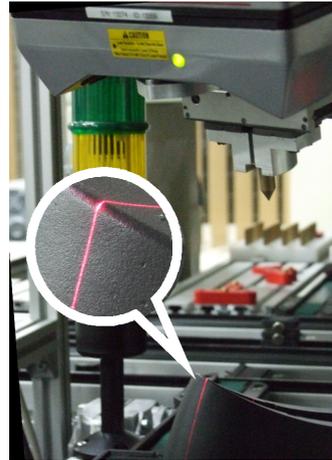
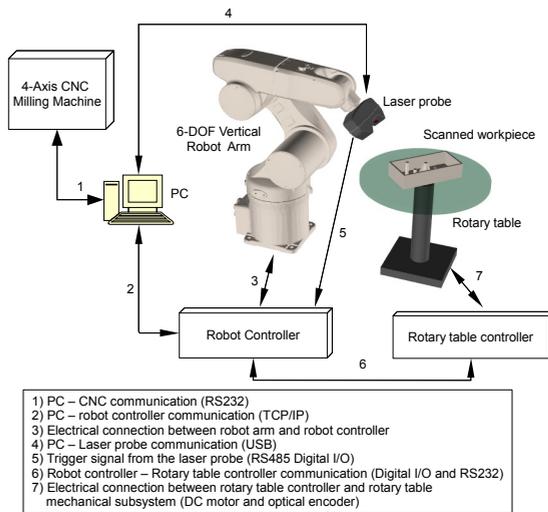


Fig. 3. (a) Overview of the 3D scanning system; (b) Laser sensor scanning a dark surface

2.1 Simulation platform

Before installing the physical hardware, a software simulation platform was developed in order to test the scanning strategies, develop the motion planning algorithms and analyze the laser sensor behavior in controlled situations.

The simulator, presented in detail in (Borangiu et al., 2008a), has two components:

- Robot motion simulation, which uses a 7-DOF kinematic model based on Denavit-Hartenberg convention (Spong et al., 2005) and renders individual rigid meshes for each DOF of the system;
- Optical simulation of the laser sensor in a virtual 3D world, using raytracing.

The simulator has two modes of operation: static and dynamic simulation. In the static mode, the robot maintains the position of the laser sensor fixed, and the two CCD image sensors show a simulated image. In dynamic mode, the user can specify complex scanning trajectories which will be followed. The result from the laser sensor is analyzed and a point cloud model,

representing the scanned virtual part, is created. The simulator is also capable of exporting animations with the scanning system following a predefined program.

The profile scanner emits a laser beam focused into a plane, which, projected on a surface, is reflected on the image sensor as a line or a curve. This laser beam may be modelled as a point light source, which is constrained to pass to a narrow opening (Fig. 4). In POV-Ray, a freeware raytracing software package (POV-Ray, 2003), the laser beam can be simulated with the code from Fig. 4 (b). Here, the laser rays start from origin, are projected in the positive direction of the Z axis and the laser rays are going to be focused in YZ plane. The narrow opening has the dimensions L (large edge) and W (small edge) and is located at a distance D from the origin.

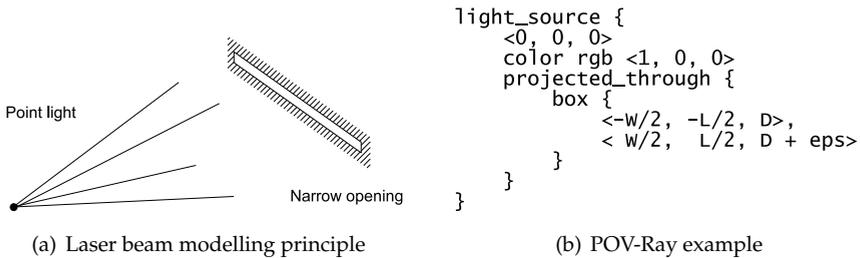


Fig. 4. Laser beam modelling using POV-Ray

The cameras used in the laser probe are modeled as two standard perspective cameras, which may be implemented in POV-Ray by entering their parameters such as position, orientation and focal length. In the following text, only one of the two cameras will be described, as the other one is identical and symmetrical to the first one. For converting the image data into 3D coordinates, a pinhole camera model (Peng & Gupta, 2007) is used.

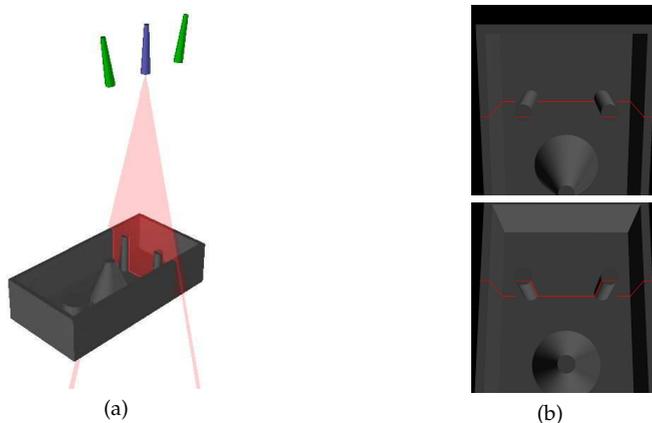


Fig. 5. Laser sensor simulation: (a) planar laser beam, two cameras and a virtual workpiece; (b) Simulated images obtained from the two cameras using raytracing

Let XYZ be the reference frame of the laser probe (Fig. 6(b) and 6(c)), and let xyz be the reference frame of the CCD array from the camera (Fig. 6(a) and 6(b)). Referring to Fig. 6(b), the camera position and orientation with respect to the laser device is given by three scalar parameters: a , b and ϕ .

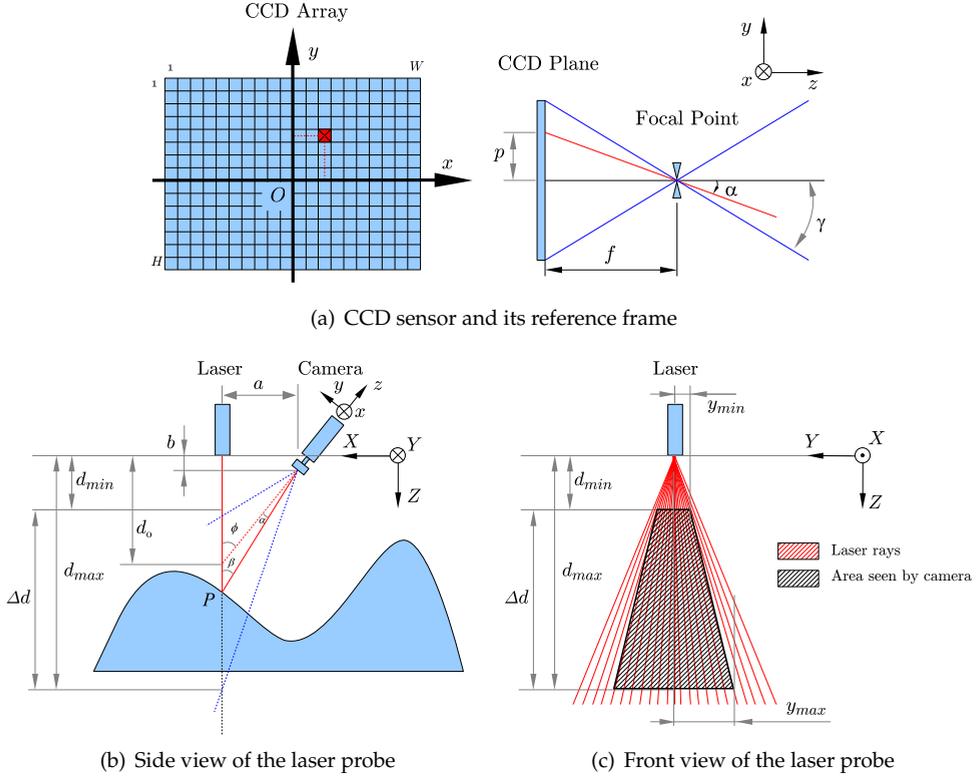


Fig. 6. Triangulation

Using these notations, let $P = (P_X; P_Y; P_Z)$ the point of reflection of a laser ray, in the XYZ reference frame, and let $p = (p_x; p_y)$ be the coordinate of the pixel at which the ray was detected on the CCD matrix, in xy reference frame. Knowing the 2D pixel coordinates p , the location of the 2D point P can be expressed using the triangulation equations (1):

$$P_X = 0 \quad P_Y = \frac{a}{f \sin \left(\phi - \arctan \frac{p_y}{f} \right)} p_x \quad P_Z = \frac{a}{f \tan \left(\phi - \arctan \frac{p_y}{f} \right)} + b \quad (1)$$

where $f = \frac{H}{2 \tan \gamma}$ if the unit length is considered to be 1 pixel, i.e. the distance between two adjacent pixels on the CCD array.

These equations are valid only under ideal conditions, i.e. when the camera and the laser sensor are perfectly aligned and there are no optical distortions from the camera lens. In practice, the transformation for converting the 2D pixel coordinates into 3D data expressed in millimetres is obtained using a calibration procedure. A look-up table model accounts for any nonlinear errors, especially lens distortion, and physical alignment between the camera and the laser is tuned using scalar offset parameters.

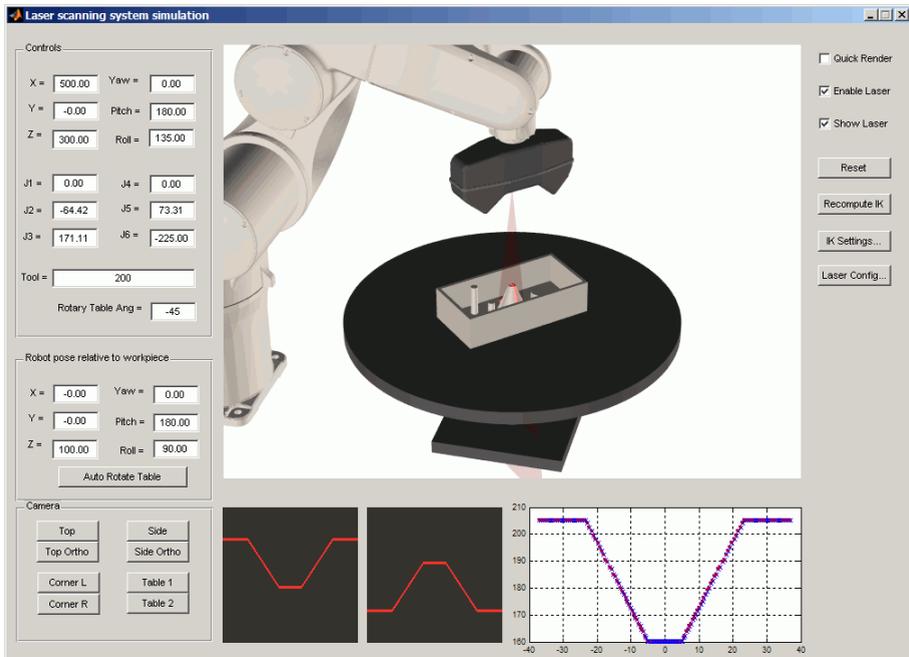


Fig. 7. Screenshot of the laser scanning system simulator

2.2 Integration Issues

A 3D point cloud model of the scanned part can be obtained by combining the measurements from the sensor with the instantaneous position of the robot. Aside from mechanical and electrical connections between the sensor and the laser probe, the two devices have to be synchronized. There are two operating modes supported by the sensor:

- Stop and look
- Buffered synchronization

With the first method, the measurements from the sensor are read only when the robot is not moving, and has reached its programmed destination. The method is the easiest to implement, does not need any synchronization signals between the vision sensor and the robot, but it is also the slowest, being limited at around 1 or 2 sensor readings per second. It is used only for debugging purposes.

The second method requires a trigger signal, which in the current implementation is sent from the vision sensor in the middle of the exposure period, from the sensor to the robot. When

receiving the signal, the robot latches its instantaneous position, and stores it in a buffer. The trigger signal may be reversed, so the robot activates the vision sensor. The data from the robot and the sensor is collected on the PC and processed at a later time. This method allows sensor readings to be taken while the robot is still in motion, and close-spaced measurements can be taken at much higher rates, e.g. 50 readings/second. However, there may be a significant delay from the of data acquisition until the data is processed by the PC. In the system used here, the bottleneck is the Ethernet link between the robot and the PC, and the delay is usually 0.2 – 0.3 seconds, and could reach 1 second. The buffers ensure that the data is matched properly even when high delays occur in communication.

Since the system also has a 7th degree of freedom, the rotary table, its controller has to be also synchronized with the robot. If the table does not rotate while the sensor is actually scanning, there is no need for additional synchronization. If the table would have to rotate while scanning takes place, the rotary table controller would have to also listen to the trigger signal.

Another issue in integrating the three components (sensor, robot and turntable) is the calibration. For accurate 3D reconstruction, the system has to know, at every moment, the position of the table and the position of the sensor, both relative to robot base. Calibration issues are discussed in detail in (Borangiu et al., 2008b) and (Borangiu et al., 2009a).

2.3 Surface Reconstruction from Point Cloud

The raw output from the laser scanning system is a *point cloud* model, consisting of a huge and disorganised set of 3D points (X , Y and Z coordinates). This format is rarely used in practice; other models can be derived from it, such as the depth map or the polygon mesh.

An example of 3D reconstruction is given in Fig. 8, when the scanned part was a small decorative object having 40 mm height. The scanning procedure was done in 16 passes, i.e. 8 passes looking at the part from above and 8 passes from below. In each scan pass, the laser sensor was moved only in translation. Between two scan passes, the turntable was rotated in 45 degree increments in order to get a complete 3D representation of the part surface.

From each scan pass, the point cloud was transformed into a depth map using a straightforward approach, mapping the farthest point to black and the closest point to white. From the depth map it was possible to obtain a mesh by taking 4 adjacent pixels and forming a quadrilateral. The meshes from the 16 scans were stitched in MeshLab, an open source package for processing and editing large and unstructured 3D triangular meshes Cignoni (2008).

3. Automatic 3D Contour Following

A second application, described in (Borangiu et al., 2009b), uses the same profile sensor for teaching a complex 3D path which follows an edge of an workpiece, without the need to have a CAD model of the respective part. The 3D contour is identified by its 2D profile, and the robot is able to learn a sequence of points along the edge of the part. After teaching, the robot is able to follow the same path using a physical tool, in order to perform various technological operations, for example, edge deburring or sealant dispensing. For the experiment, a sharp tool was used, and the robot had to follow the contour as precisely as possible. Using the laser sensor, the robot was able to teach and follow the 3D path with a tracking error of less than 0.1 millimetres.

The method requires two tool transformations to be learned on the robot arm (Fig. 10(b)). The first one, T_L , sets the robot tool center point in the middle of the field of view of the laser sensor, and also aligns the coordinate systems between the sensor and the robot arm. Using

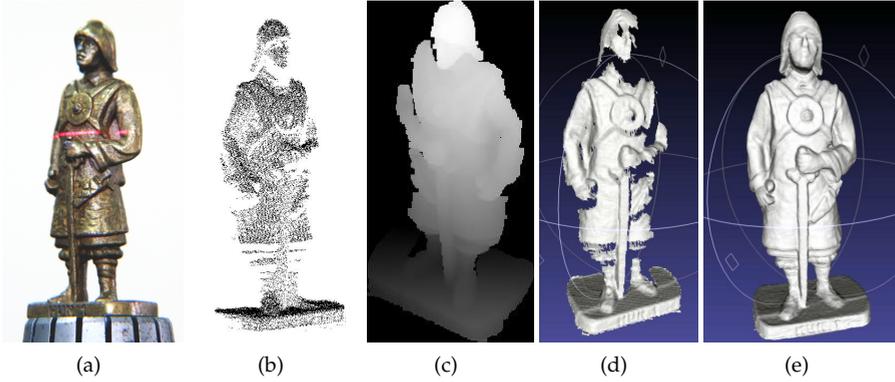


Fig. 8. 3D reconstruction example: (a) Photography of a decorative object, along with the laser stripe; (b) Point cloud from one scan pass; (c) Depth map computed from the point cloud; (d) Mesh model obtained from the depth map; (e) Complete 3D model, postprocessed in MeshLab

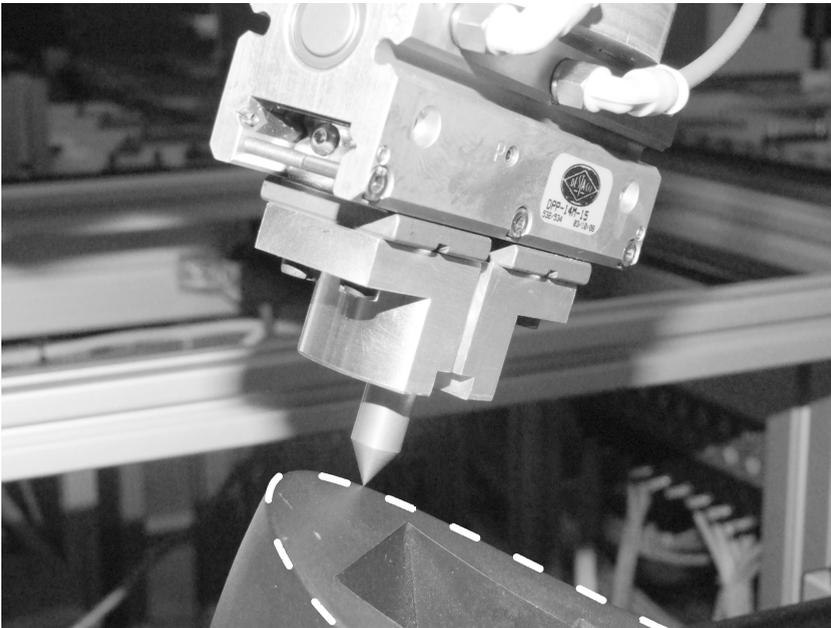


Fig. 9. 3D contour following using a sharp tool tip

this transform, any homogeneous 3D point $P_{sensor} = (X, Y, Z, 1)$ detected by the laser sensor can be expressed in the robot reference frame (World) using:

$$P_{world} = T_{robot}^{DK} T_L P_{sensor} \quad (2)$$

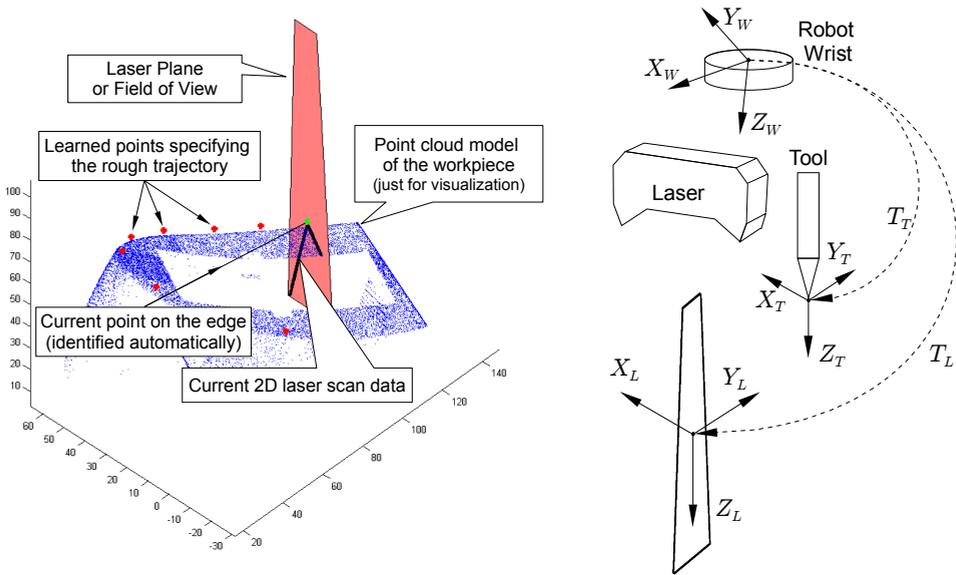


Fig. 10. (a) Trajectory learning assisted by automatic edge recognition; (b) The two tool transformations: one for trajectory teaching, other for following it using the physical tool

where T_{robot}^{DK} represents the position of the robot arm at the moment of data acquisition from the sensor. The robot position is computed using direct kinematics.

The second transformation, T_T , moves the tool center point on the tip of the physical tool. These two transformations, combined, allow the system to learn a trajectory using the 3D vision sensor, having T_L active, and then following the same trajectory with the physical instrument by switching the tool transformation to T_T .

The learning procedure has two stages:

- Learning the coarse, low resolution trajectory (manually or automatically)
- Refining the accuracy by computing a fine, high resolution trajectory (automatically)

The coarse learning step can be either interactive or automatic. In the interactive mode, the user positions the sensor by manually jogging the robot until the edge to be tracked arrives in the field of view of the sensor (Fig. 10(a)). The edge is located automatically in the laser plane by a 2D vision component. In the automatic mode, the user only teaches the edge model, the starting point and the scanning direction, and the system will advance automatically the sensor in fixed increments, acquiring new points. For non-straight contours, the curvature is automatically detected by estimating the tangent (first derivative) at each point on the edge. The main advantage of the automatic mode is that it can run with very little user interaction, while the manual mode provides more flexibility and is advantageous when the task is more difficult and the user wants to have full control over the learning procedure.

A related contour following method, which also uses a laser-based optical sensor, is described in (Pashkevich, 2009). Here, the sensor is mounted on the welding torch, ahead of the welding direction, and it is used in order to accurately track the position of the seam.

4. Conclusions

This chapter presented two applications of 3D vision in industrial robotics. The first one allows 3D reconstruction of decorative objects using a laser-based profile scanner mounted on a 6-DOF industrial robot arm, while the scanned part is placed on a rotary table. The second application uses the same profile scanner for 3D robot guidance along a complex path, which is learned automatically using the laser sensor and then followed using a physical tool. While the laser sensor is an expensive device, it can obtain very good accuracies and is suitable for precise robot guidance.

5. References

- Borangiu, Th., Dogar, Anamaria and A. Dumitrache (2008a), Modelling and Simulation of Short Range 3D Triangulation-Based Laser Scanning System, *Proceedings of ICCCC'08*, Oradea, Romania
- Borangiu, Th., Dogar, Anamaria and A. Dumitrache (2008b), Integrating a Short Range Laser Probe with a 6-DOF Vertical Robot Arm and a Rotary Table, *Proceedings of RAAD 2008*, Ancona, Italy
- Borangiu, Th., Dogar, Anamaria and A. Dumitrache (2009a), Calibration of Wrist-Mounted Profile Laser Scanning Probe using a Tool Transformation Approach, *Proceedings of RAAD 2009*, Brasov, Romania
- Borangiu, Th., Dogar, Anamaria and A. Dumitrache, (2009b) Flexible 3D Trajectory Teaching and Following for Various Robotic Applications, *Proceedings of SYROCO 2009*, Gifu, Japan
- Calin, G. & Roda, V.O. (2007) Real-time disparity map extraction in a dual head stereo vision system, *Latin American Applied Research*, v.37 n.1, Jan-Mar 2007, ISSN 0327-0793
- Cheng, F. & Chen, X. (2008). Integration of 3D Stereo Vision Measurements in Industrial Robot Applications, *International Conference on Engineering & Technology*, November 17-19, 2008 – Music City Sheraton, Nashville, TN, USA, ISBN 978-1-60643-379-9, Paper 34
- Cignoni, P. et al., MeshLab: an Open-Source Mesh Processing Tool *Sixth Eurographics Italian Chapter Conference*, pp. 129-136, 2008.
- Hardin, W. (2008). 3D Vision Guided Robotics: When Scanning Just Won't Do, *Machine Vision Online*. Retrieved from <https://www.machinevisiononline.org/public/articles/archivedetails.cfm?id=3507>
- Inaba, Y. & Sakakibara, S. (2009). Industrial Intelligent Robots, In: *Springer Handbook of Automation*, Shimon I. Nof (Ed.), pp. 349-363, ISBN: 978-3-540-78830-0, StÄijrz GmbH, WÄijrzburg
- Iversen, W. (2006). Vision-guided Robotics: In Search of the Holy Grail, *Automation World*. Retrieved from <http://www.automationworld.com/feature-1878>
- Palmisano, J. (2007). How to Build a Robot Tutorial, *Society of Robots*. Retrieved from http://www.societyofrobots.com/sensors_sharpirrange.shtml
- Pashkevich, A. (2009). Welding Automation, In: *Springer Handbook of Automation*, Shimon I. Nof (Ed.), pp. 1034, ISBN: 978-3-540-78830-0, StÄijrz GmbH, WÄijrzburg
- Peng, T. & Gupta, S.K. (2007) Model and algorithms for point cloud construction using digital projection patterns. *ASME Journal of Computing and Information Science in Engineering*, 7(4): 372-381, 2007.
- Persistence of Vision Raytracer Pty. Ltd., *POV-Ray Online Documentation*

- Scharstein, D. & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7-42, April-June 2002.
- Spong, M. W., Hutchinson, S., Vidyasagar, M. (2005). Robot Modeling and Control, *John Wiley and Sons, Inc.*, pp. 71-83, 2005