¶
¶
¶
¶
¶
¶
¶

GENERATION OF ADAPTIVE ROBOT SCANNING PATHS FOR 3D SURFACE RECONSTRUCTION

¶

TH. BORANGIU[*] **A. DOGAR**[*] **A. DUMITRACHE**[*]

¶

Abstract: *The paper addresses an implementation issue of a robotized reverse engineering platform for not modelled obje cts. One important objective of this work is to develop suitable scanning strategies and robot motion patterns for automatic sensor guiding and acquisition of 3D surface data of objects to be modelled. The proposed strategy implies the synchronization of the robot movements with the rotation of the turntable, in order to avoid collisions or axis out of range and to keep, if possible, the robot in the central area of its workspace. A 6 d.o.f vertical articulated robot simulator was developed to generate the set of necessary robot configurations and turntable angles corresponding to the scanning strategies. The simulator is used to test whether a scanning path is feasible from the kinematics point of view.*

¶

Keywords: *intelligent control, path planning, shape description, algorithms, simulation*

1. Project Description

¶

The described work is part of the designing and implementing of reverse engineering platform. The platform consists in a 6. d.o.f robot arm, dual camera laser scanner device, a turntable and a 5 axis CNC milling machine. The laser scanning device mounted on the robot arm probe is able to measure distances from 70 to 250 millimetres, with an accuracy of achieving 30 µm. The robotic arm moves around the work piece being scanned by computer-generated adaptive scanning paths. The scanning device is a class-2, short distance, triangulation one, and has two CMOS sensors allowing the scanning of complex object surfaces. The optimal scanning distances range from 71 mm to 242 mm. The width of the scanning line varies between 31 mm and 83 mm, and the average measuring precision at point level is 31 µm. The acquisition rate is between 50 and 150 frames per second, the number of points which are read on a scanning line being 480. The laser range finder system is

---

[*] Dept. of Automation and Applied Informatics,Computer Science and Automation Faculty, University Politehnica of Bucharest

This work is funded by the National Council for Scientific University Research, in the framework of the National Plan for Research, Development and Innovation, grant 69/2007.

interfaced to a 3.2 GHz IBM PC-type station by means of a standard USB input port, and uses additionally a digital RS485 line for synchronization with the robot controller

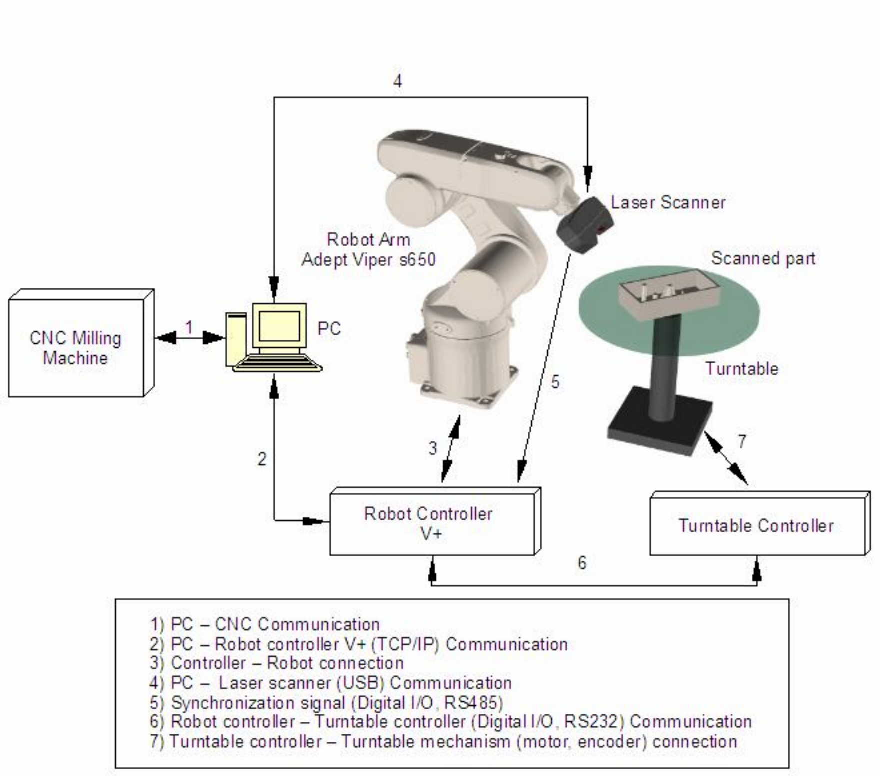


Fig 1. *Hardware architecture of the laser scanner-robot-CNC machine platform for reverse engineering tasks*

The usual ‘teach and play’ method used in industry to generate work data for robots is not suitable in this case. The proposed scanning strategies consist of two stages. In the first stage a predefined motion pattern is followed based on the chosen scanning strategy. There are considered three types of object classes that are to be modelled and for each scanning strategies are developed. Thus, in the first stage of the scanning process, an off-line programming method is used to generate motion data for the robot. One of the difficulties to overcome in order to generate functional OLP data, is Collision-Free Path Planning, which creates safe robot paths so that there are no collisions between robot and obstacles**.**

2. Reverse Engineering Platform Simulator

¶

The reverse engineering platform simulator is designed to be a development tool and test bench for the developed adaptive scanning algorithms. The three main components of the platform are simulated. For the robotic arm, the simulator allows displaying the robot in any user-defined position. The robot simulator is comprised of two modules. The static simulation module allows specifying the robot pose, using one of the three input methods:

- specify the angular values for each joint and for the rotary table (direct kinematics)
- specify the position in Cartesian coordinates and the orientation in ZYZ’ Euler angles, in the robot’s reference frame (inverse kinematics with respect to robot)
- specify the angle of the rotary table, and the position and the orientation of the robot end point in the rotary table’s reference frame (inverse kinematics with respect to rotary table)

If the specified pose with respect to rotary table cannot be reached, the table is rotated automatically until the requested pose can be reached.

The motion simulation module lets the user simulate and analyze the behaviour of the robot using a sequence of user-defined trajectories. The user interface has an editor for the motion sequence, and controls for generating the animation.

The simulator can use two modes for computing inverse kinematics:

- Internal inverse kinematics routine, using Peter Corke's Robotics Toolbox for Matlab, Release 7.1
- The inverse kinematics routine from an Adept robot controller, which is invoked using a TCP/IP connection.

The first method, which uses a generic inverse kinematics based on the pseudo-inverse of the manipulator Jacobian [1], have the advantage that it can run on a standalone computer, and it executes fast for most situations where a solution for inverse kinematics does exist. However, if a solution cannot be found, the routine hangs for a few seconds before reporting the lack of convergence. Another disadvantage is that this routine is more likely to find a solution outside the joint limits of the robot, when the initial robot pose is far from the destination.

The second method is more robust with respect to initial pose of the robot, can be configured to compute solutions with desired configurations of the robot (i.e. LEFTY, RIGHTY, ELBOW UP, ELBOW DOWN, FLIP, NOFLIP), and runs fast even if there is no solution for inverse kinematics, although it has an overhead from TCP communications. When it is called directly from V^+, without the network overhead, this method is very fast, being able to perform about 500 inverse kinematics computations per second.

The motions can be defined as trajectories that interpolate between some positions, specified as transformations. A transformation contains six values: the Cartesian coordinates for the position, and the yaw, pitch and roll angles for the orientation. The points may be defined with respect to two predefined reference frames: the robot reference frame, which is fixed, and the rotary table reference frame, which changes every time the rotary table moves.

The user has two choices for interpolation between positions: either straight line, or joint interpolated motion. These modes correspond to the MOVES and MOVE instructions from the V^+ language.

When the motions are executed in the rotary table reference frame, the interpolation is also performed in the rotary table reference frame. This means that, for a straight line motion, for any possible trajectory for the rotary table, the motion of the end effector of the robot will be always be a line with the respect to the table, but with respect to the robot base, the motion may have a complex shape. This situation is handled by the motion planner for the rotary table

As for the rotary table, its rotation is simulated, and the workpiece that sits on the table should rotate synchronously with the table. Behaviours such as inertial movement due to fast table movements do not have to be simulated; the workpiece should be considered attached to the table.

For the laser probe, the software simulates the interaction of the laser beam with any user-defined workpiece, having various surface properties. The two cameras which are integrated into the laser probe are also simulated, and the image which captured by them is displayed to the user. Furthermore, the laser beam is detected in the images from the two cameras, and the triangulation equations are applied to them in order to compute a point cloud. The point clouds obtained from simulating the scanning process from different viewing angles will be transformed into a fixed coordinate system that will be attached to the workpiece being scanned.

In the second stage of the scanning process the acquired data is used to generate new scanning paths, whether to obtain a high resolution detail or to manage occlusions.

¶

3. Adaptive Scanning Strategies

¶

Using a predefined strategy like raster or spiral gives the advantage of an easy implementation and can lead to very good results in many cases. There are cases when a predefined strategy will not be able to detect all the features of the analyzed object. In this paper we consider only one type of objects that are to be scanned. So in the case of molds, the scanning surface can be described as a $z = f(x, y)$ function, so it is a 2.5D surface. In this case it is possible the scanning using only trajectories in the *XY* plane, varying the distance of the scanner.

It is proposed a two stage scanning strategy. In the first stage of the scanning process, there is no information regarding the object; this laser scanning will be done at high speed rates, which implies a low resolution data set. In the second stage of the scanning process, there is an approximate model of the object and it can be computed an efficient scanning strategy. The scanning speed rate will be much lower; this is why the scanning paths used must be efficiently optimized. On the other side, the optimization in the first stage is not critical.

Let consider the mould profile in the Figure 2. It can be observed the trapezoidal region of the laser plane which is analyzed by the optical sensor. It can be noticed that the prolongation of the trapeze sides intersect in the point *O'*, which is not identical with the origin of the laser beam *O*. Also, there are parts of the laser beam which are not in the camera field of view.

There appears an occluded region due to one vertical wall, for which a supplementary trajectory must be generated.

If in the first stage of the scanning process, the obtained cloud point is not analyzed, the next data acquiring pose is *X* (Fig. 2).

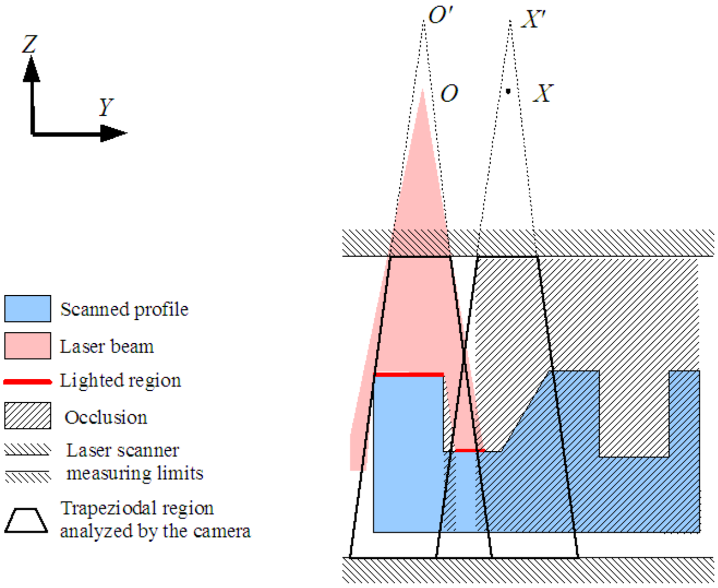


Fig. 2. *First scanning of a profile and the next trajectory in the scanning process*

It can be observed that there is redundant data in the captured point cloud, so there must be eliminated the excess points.

In order to generate an optimal strategy for the second stage of the scanning process, the occlusions must be analyzed in the first stage. So a possible solution for the proposed profile is shown in the Figure 3, based on the following rules:

- if there is an occluded area, generate trajectory for scanning this area;

otherwise, generate the next path so that the up-left corner of the trapeze have the same horizontal coordinate as the last point detected in the previous scan.

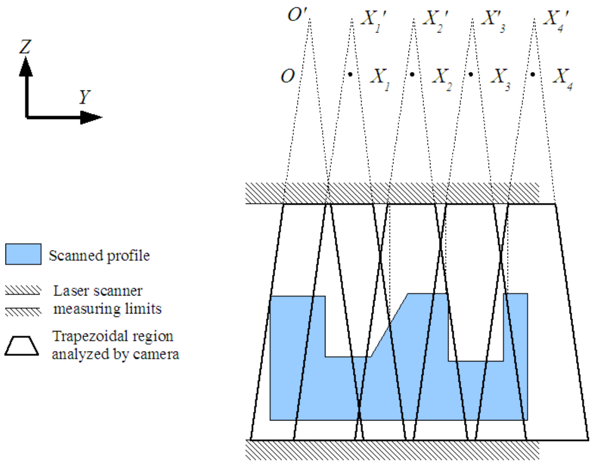
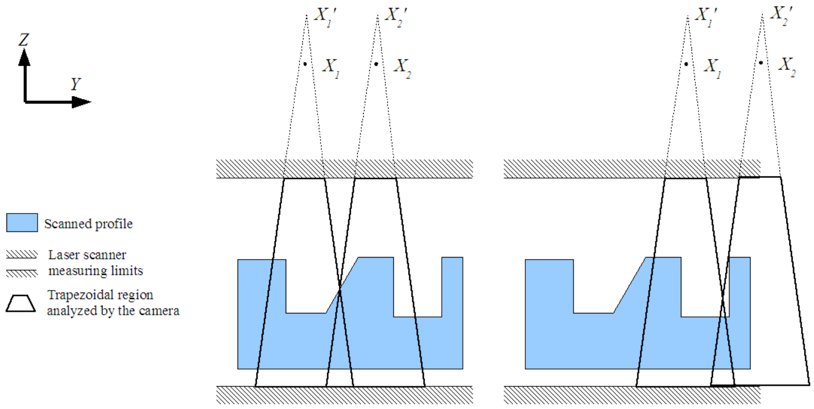


Fig. 3. *Possible solution for adaptive scanning in the first stage*

In this second stage there is available an approximate model of the scanned object. This model is used for modeling the optimal strategy that will lead to the complete model of the object.

In the Figure 4, are presented two consecutive trajectories. In the *a)* case the intersection of the two trapezes sides on the profile surface does not create any occlusions. In the *b)* case the intersection of the trapezes sides on the mould surface will lead to an occlusion, in this case the trajectory must be placed so to be tangent to the vertical wall.



a) *b)*

Fig.4. *Two consecutive trajectories for adaptive scanning in the second stage*

For implementing this situation, it is useful a nonlinear transformation of the mould profile. This transformation is defined for circle of given center and radius (Fig. 5) and will be noted as **T**.

Let be the circle $O = (x_0 ; y_0)$ and radius r, in the Cartesian plane XY, and a point $P = (p_x,\ p_y)$ that will be transformed in $P' = (\ p_x' ; p_y'\ )$. Let be α the angle that is made by the segment OP with the axis Oy, and ρ the distance from P to O.

The coordinates of the point P' will be for a circle with an arbitrary center:

$$p_x' = x_0 + \alpha \qquad (1)$$

$$p_x' = y_0 - \rho \qquad (2)$$

where

$$\alpha = \operatorname{arctg}\frac{p_x - x_0}{y_0 - p_y} \qquad (3)$$

$$\rho = \sqrt{(p_x - x_0)^2 + (p_y - y_0)^2} \qquad (4)$$

The transformation **T** is not defined by the point *O*.

Using this transformation, a circle with an arbitrary radius *R* and center *O* becomes a horizontal line at the level $y' = -R$, and a circle radius, becomes a vertical line

$x' = \alpha$, where α is the angle between the radius and the *Oy* axis. So, the points *P'* and *Q'* from Figure 5 will have the same abscise, since *P*, *Q* and O are collinear.

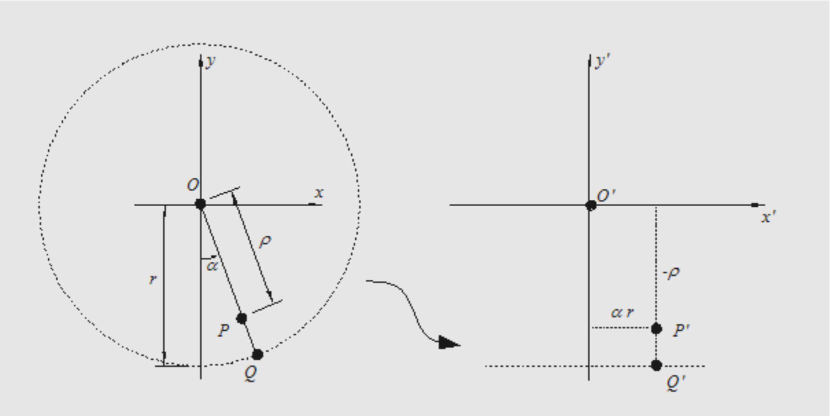


Fig. 5. *Nonlinear transformation from circle to segment*

This transform facilitates the following calculus:

a) applied with the center in the point *X_1'* from Figure 5, permits the computing of the intersection between the scanned profile and the side of the trapeze, by inspecting the *y'* coordinates of the scanned profiled points;

b) applied in the *X_i* points, the origin of the laser beam, facilitates the detection of the vertical wall which lead to occlusions. This vertical wall, appear under this transformation as concavities.

¶

4. Conclusions and future work

¶

The simulator of the laser scanner – robot arm system is designed to be a development tool and test bench for the adaptive scanning algorithms developed. Using a simulation environment for designing the scanning algorithms has several advantages:

- The possibility of collisions between the robotic arm, laser probe and workpiece is eliminated. As the laser probe is an expensive device, collision avoidance is a very important point to consider;
- The system can be analyzed in ideal conditions, with no surface reflections, external light sources or perturbations in the measurements;
- The parameters of the scanning system components, like camera location, focal length, optical
- Sensor resolution, laser beam width, rotary table size and location, can be freely changed, and the influence of these changes can be analyzed thoroughly;

This work has been done in the Robotics and AI Laboratory of the Faculty of Automatic Control and Computers.

The close-future work includes developing the algorithms for complex 3D path following, optimizing and speed – up methods of the implemented algorithms.

References

¶

1. Borangiu, Th. (2004). *Intelligent image processing in Robotics & Manufacturing,* Romanian Academy Publish. House, Bucharest.
2. O'Gorman, L. (1996). Subpixel Precision of Straight-Edge Shape for Registration and Measurements, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **18**, *11*, 57-68
3. Persistence of Vision Raystracer Pty. Ltd., POV-Ray Online Documentation
4. Adept Technology, Inc. Web Site, www.adept.com