A Method for Automatic Spray Painting of Unknown Parts

Andreas Pichler, Markus Vincze
Institute of Flexible Automation
Vienna University of Technology
1040 Vienna, Austria
e-mail: {ap,vm}@infa.tuwien.ac.at

Henrik Andersen, Ole Madsen
Department of Production
University of Aalborg
9220 Aalborg, Denmark
e-mail: {i9hja,i9om}@iprod.auc.dk

Kurt Häusler
Profactor, Profactor Productionsforschungs GmbH
4400 Steyr, Austria
e-mail: Kurt.Haeusler@profactor.at

Abstract

Today industrial automation of spray painting is limited to high part volumes and robot trajectories that are programmed by off-line programming and manual teach-in. This paper presents an approach that uses range image data to obtain the geometry of an unknown part and to automatically generate the robot spray painting trajectories. Laser strip range sensors are installed in front of the paint booth to acquire a range image of the part. Utilizing process knowledge (a geometric library containing constraints specific for the painting application) geometric primitives are detected in the range data. From the geometric primitives a normal vector field is generated that enables to extract main faces. The main faces are located in 3D space and the process knowledge related to each geometric primitive is utilized to obtain the trajectory for the paint gun. Results of painting a car mirror and a stearing column are given.

1 Introduction

The objective of the European RTD project *Flex-Paint* (www.flexpaint.org)is to automate robot programming for painting applications of small lot sizes with a very high number of part variants. The project goal is to provide economic possibilities for usage of robots for industrial painting tasks. As a matter of fact, economic realisation hinder the application of the currently used conventional automation technology (off-line programming and/or manual teach-in) used for high volume production.

In this paper an inverse approach is presented to automatically obtain robotic paint paths from range sensor data and to automatically generate a feasible, complete and executable robot program. The approach must cope with a large spectrum of parts as depicted in Fig. 1. For each industrial customer the part families are known. The goal is to be able to paint any order of parts coming along the conveyor. The technical challenge is to detect the geometry of the part on the conveyor, to automatically infer from the geometry the robotic painting trajectory and to automatically generate a collision free robot program.

Figure 1: Examples of parts to be painted automatically: gearbox with motor, compressor tank, steering column of a truck, small parts on a frame, and rear view mirrors. The pictures show the parts in correct relative size.

The automatic generation of the tool path is known from lathing and milling. Rotary parts are treated as 2D parts and automatic generation of a lathe path from a CAD model is well known. The extension to generic 3D parts is difficult. A possible approach is used in rapid prototyping methods, which produce the part in layers (stereolithography, laser sintering and similar processes), an approach not feasible for painting and similar manufacturing processes. Robots can be used to enable more than 2D paths, however use is presently limited to flat structures such as blades or paddles [7]. 3D turbine blades can be milled when considering planar cross sections [6, 9]. The grid cell approach in [10] handles 3D surfaces via approximations in rectangular boxes, which are iteratively refined. The robot handles the milling tool and cuts the work piece. Due to the rectangular approximation the robot executes the path in layers and aliasing patterns remain on the surfaces.

The automatic generation of a 3D paint path has been attempted in the SmartPainter project. The painting motion was generated by virtually folding out the surfaces to be painted, putting on the painting motion and folding back the surfaces and letting the painting motions following this folding of surfaces [2, 8]. However, this strategy is only applicable when 3D models of the objects are available and the curvature of the objects is relatively small. The patented technology from Advanced Robotics Technologies uses a 2D digital photo as input. The user decides on the screen where to apply paint strokes. The path planning for a robot is then done automatically [US patent no. US 5,429,682].

The approach presented here uses range images of the part to detect geometric features and select the appropriate painting strategy. No CAD models of the parts are required. The paper progresses by giving an overview of the "inverse approach" to automatically generate the paint path from the sensor data (Section 2). The next sections describe the extraction of the basic geometries (Section 3 and 5) and the generation of the paint path (Section 4). Section 6 gives the results of the experiments.

2 System Description

The FlexPaint approach is based on the observation that the parts in Fig. 1 comprise a large number of *elementary geometries* with typical characteristics for an entire product family. Examples are rib-sections (cooling ribs), cylindrical surfaces (both shown on the motor, left in Fig. 1), and cavities (shown at the top of the gearbox and at the steering column in Fig. 1). Another type of surface are the surfaces of the rear view mirror. These surfaces are smooth free-form surfaces, which are very difficult to represent by use of simple geometric attributes such as cylinders, spheres and boxes. Hence, the goal is to specify these elementary geometries in such a way that generic methods for detecting and for path planning can be developed and that the variety of geometries seen in the applications is encompassed.

The specification of elementary geometry types is based on the constraints of the painting process. The idea is to detect elementary geometries that can be linked to a process model. For example, the geometry "flat surface" can be painted with a simple pattern of straight paint strokes. More complex geometric shapes, such as cavities or ribs, need specific painting strategies: spraying into the cavity and painting parallel to the rib orientation, respectively.

The elementary geometry types are defined in the *Geometry Library* and related to the process knowledge, which is specified in the *Procedure Library*. Fig. 2shows how the FlexPaint system operates.

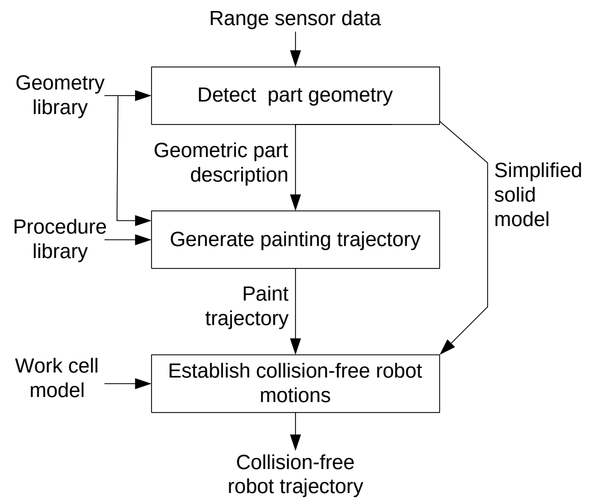

Figure 2: Block diagram of the FlexPaint system.

The module "detect part geometry" uses the geometric definitions of the geometry library to describe the part given by the range sensor data (for details see Section 3). Additionally a simplified solid model is calculated, which represents a convex hull approximation of the part. It is utilized to model the part when generating collision-free motions (see Section 5).

The geometric part description is used to generate the painting trajectory of the spray gun (see Section 4). The module "establish collision-free robot motions" takes the tool trajectory and calculates the actual arm trajectories for a given robot manipulator and finally generates the program in a specific robot language (see Section 5).

3 Geometry Detection

The objective is to detect the elementary geometries in a range image. This section presents the approach of PFeatureDetector (Process-oriented Feature Detector) to solve this task. Using a triangulation laser scanner and camera from IVP, Schweden, a range image of the part is taken when traversing on the conveyor. Triggering the camera with a signal of an impulse generator measuring the actual conveyor motion assures that scans are equidistant. Up to 600 scans per second can be taken. Fig. 3 shows the 3D points of an image of the steering column after filtering. The resolution for a scanning width of $2m$ is about $1.2mm$, which is sufficient for most painting applications.

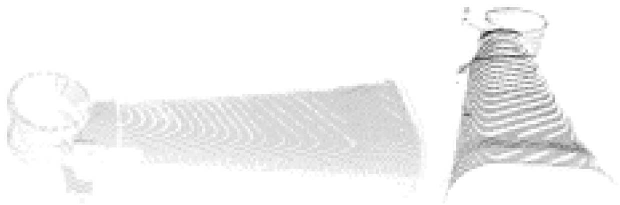

Figure 3: Range image point cloud of a full part scan showing the stearing column. Left: front view. Right: side view.

The technical challenge is to segment the range data into the areas of elementary geometries that have been defined in the *Geometry Library*. Fig. 4 summarises the procedure of PFeatureDetector. The calibrated images are segmented into the parts after subtracting the skid (for example the mirrors in Fig.1). For each segmented part a mesh and an edge map are generated using the procedures of the VTK Visualisation Tool Kit. Additionally, for each part a region segmentation is executed. Using the result of [3], the method developed in [4, 5] is used, since it is more robust and faster in data processing than the classical approach of [1].

Presently two specific elementary geometries can be detected: ribs and cavities. *Rib detection* is based on finding a minimal number of parallel lines at equal distances in the edge map.

Cavity detection is based on finding a region that is lower (using the direction of the surface normal) than the neighbouring regions. Several regions that

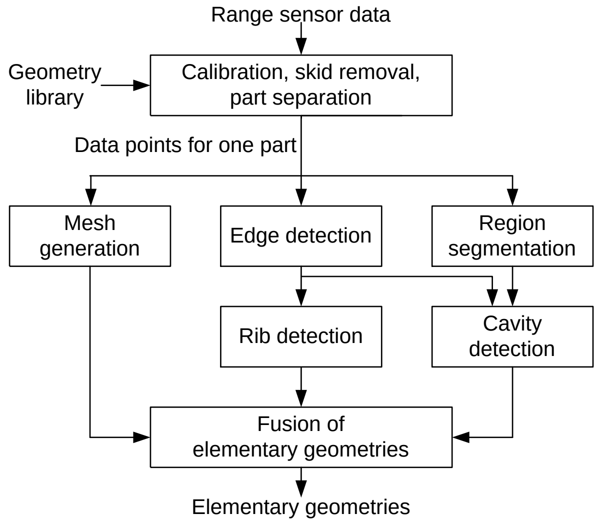

Figure 4: Block diagram of detecting elementary geometries in PFeatureDetector.

are lower than the surrounding region are clustered into a larger region. Using the edge map the rim of the cavity is detected with an accuracy in the range of resolution $1.2mm$. This is important to obtain the full cavity region including the areas that are occluded due to the typical shadows when taking the range image. Fig. 5 shows the regions which have been automatically classified as cavities.

All remaining surfaces are represented with a mesh. The original point data is approximated with a mesh (using VTK) to reduce the amount of data for the path generation step. The reduction is needed to assure that path planning can be done in short time.

In the last step, the fusion of features assigns an elementary geometry (free-form surface, rib, cavity) to each region and determines the parameters (size) and the location relative to the skid mounted on the conveyor. For a part as shown in Fig. 5 the entire process as described in Fig. 4 requires a calculation time of 5 seconds and a 800 MHz PC with 256 Mb RAM. Visualisation of the results takes longer than this, but is not needed for the industrial in-line usage.

4 Paint Path Generation

The generation of paint paths is divided into the following steps: planning of the painting process, planning of collision free spray gun motions, and simulation of the robot trajectory to generate a robot program.

The module 'Generate Painting Trajectory', shown

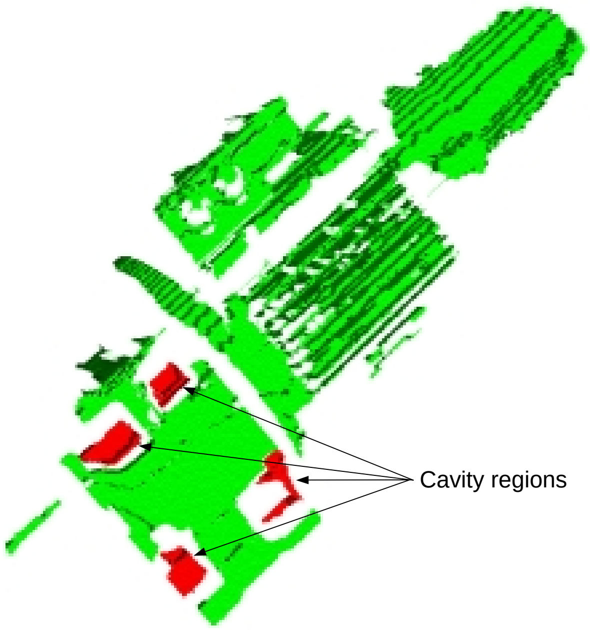

Figure 5: The cavities detected on the gearbox. Cavity detection is executed in about one second.

in Fig. 2, specifies a trajectory of the spray gun, which satisfies the desired paint quality. In this module only spray gun motions are considered in relation to process quality. No restrictions of robots are made and collisions between the spray gun and its surroundings are not considered. The system uses the "Geometry Library" and the "Procedure Library" in order to plan this trajectory. The Geometry Library specifies for each geometric primitive one or more painting procedures, which may be applied for painting that particular type of geometric primitive. The painting procedure specifies how to apply spray gun motions to the surfaces in order to achieve a satisfactory process quality. The procedure library is established through experimental work.

The basic idea is to enable planning of paint strokes that continue throughout the parts even though different geometric primitives must be covered along the surface and even though continuous robot motions cannot follow the surface. The system will attempt to approximate the triangular patches of the surface model by larger plane regions (virtual surfaces), which are oriented in a few main directions. Fig.6 shows a geometry model consisting of triangular patches.

In Fig.7 the patches are approximated by four virtual surfaces, of which 2 are parallel. The painting procedures are executed relative to these virtual surfaces. Each virtual surface represents only one type of geometric primitive and the same painting procedures can be therefore used for the entire part surface. In case different geometric primitives are present along the surface, the system will attempt to establish continuous spray gun motions covering virtual surfaces, which are in continuation of each other. The spray gun motions are specified by paint lines, such as illustrated in Fig.8. From this figure and Fig.7 it can be seen that the paint lines follow the directions and the plane positions of the virtual surfaces. The painting procedure specifies how many strokes the spray gun must make along the paint line and which painting parameters are applied in each of these strokes.

Figure 6: Filtered point data and reduced mesh representing the stearing column. Range image was taken with one scan only.

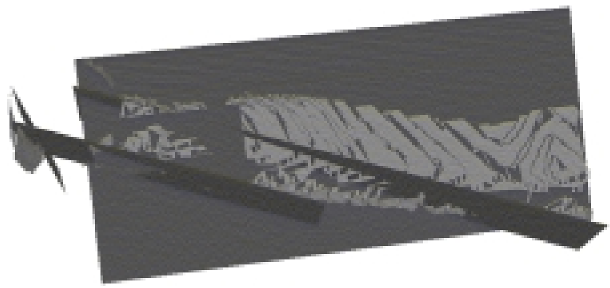

Figure 7: Virtual surfaces approximating major paint areas.

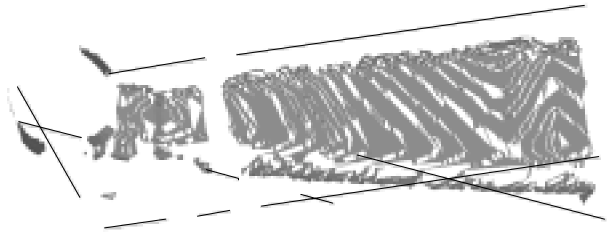

Figure 8: The painting strokes represented by lines.

5 Collision Free Trajectories

The module "Establish Collision Free Robot Motions", shown in Fig. 2, checks the specified spray gun trajectory in order to detect possible collisions between the spray gun and the part or between the spray gun and any other obstacles in the robot cell. In case any collisions are detected this module will change the position and/or orientation of the spray gun within some specified tolerances in order to avoid the collision. Hence, collision is automatically prevented before execution of the robot program. This particular module is already a commercially available product from AMROSE, Ltd., but had to be customized for the FlexPaint project. The collision avoidance is performed using a simplified geometry model of the part surface and the obstacles in the robot cell, such as conveyors etc. The simplified geometry model, conveyor and paint lines are shown in Fig. 9.

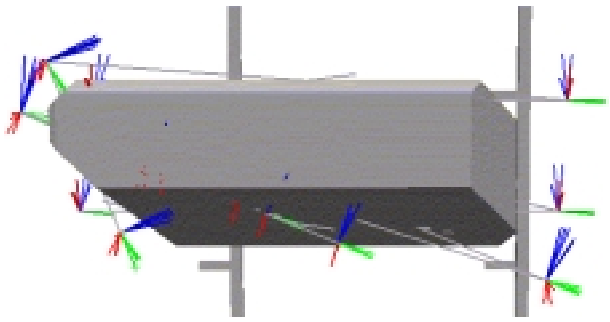

Figure 9: Simplified geometry model, the skid (vertical bars) and paint lines with spray nozzle orientation.

The off-line programming system, RobotStudio from ABB imports the painting trajectories for simulating the spray gun motions including robot kinematics (see Fig. 10). The ToolPlanner software from AMROSE runs in the background and checks if any collisions occur between the robot and its surroundings (up to now collision avoidance has only been made for the spray gun, but it is planned to extend this to global collision avoidance). The off-line programming system finally converts the collision free spray gun trajectory to a robot program. RobotStudio utilises a virtual controller of any ABB robot, hence, the physical robot will execute the program just as it is simulated on the computer.

6 Experiments

The system is already implemented as a prototype and has been tested in ABB's technical center

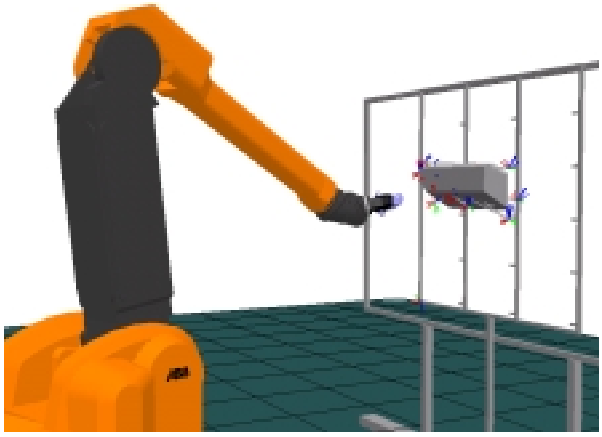

Figure 10: Simulation of the collision free robot motion.

in Eichen, Germany (see Fig. 11). The purpose of these experiments was to prove the basic system concept. It was realised that process quality has to be optimized by establishing validated painting procedures for the individual geometric primitives. The painting procedures used were not established by preceding experiments. The prototype system used only one laser scanner and one robot. Since the surface was only scanned from one direction it was only possible to perform automatic spray painting of the scanned surface. However, it was observed that a relatively good painting quality was achieved on these parts of the surfaces, which were scanned. Some painting errors were caused by switching on and off the spray gun in wrong positions. Fig. 12 shows the robot executing the automatically generated program.

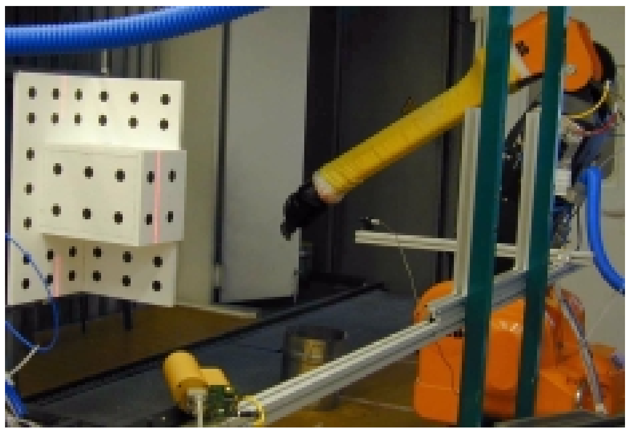

Figure 11: Painting cell with the calibration tool (left) and the camera (bottom front).

The prototype installation demonstrated to be ca-

Figure 12: The robot painting the stearing column with the automatically generated program. The black paint can be seen on the part.

pable of realising production constraints: (1) any series of parts of the industrial parts shown in Fig. 1 can be scanned. And, (2) the motion of the conveyor requires a processing time of about 30 seconds. Range image processing requires only about five seconds on a PC and path planning can be executed in 30 seconds on a high end PC.

7 Conclusion

A method has been developed for automatic spray painting of unknown parts. Experiments at ABB flexible automation, Friedberg have shown that the system concept is feasible. The input for the system does not include any pre-established CAD models or other information about the parts. The parts are entirely specified by geometry models, provided by a laser scanning system and customized feature extraction methods. Up to now the parts were scanned by one laser scanner only. The parts of the surfaces that were visible to the scanner were modelled in a 3D geometry model, which was used to automatically generate the motions of the spray gun. Collisions between the spray gun and the part and conveyor system were automatically avoided by a collision avoidance module from AMROSE, Ltd. Robot programs were automatically generated using the RobotStudio off-line programming system from ABB. The robot programs were executed and the robot painted the scanned part of the surfaces. It was observed that the quality of the painted surfaces was good, although the Procedure Library is not yet sufficiently developed.

Even though the project is primarily aimed towards robotic spray painting, the "inverse approach" proposed can be applied for obtaining process motions for a large range of processes in the field of surface treatment. Examples of processes in which the approach can be applied are: powder painting, washing and cleaning with liquid (including high-pressure cleaning), washing and cleaning with physical contact between tool and part, de-greasing, sandblasting, polishing, sealing (e.g. for corrosion protection), inspection systems, polishing, grinding, deburring and gluing.

Acknowledgements

This work is mainly supported by the RTD project FlexPaint (GRD1-1999-10693). Patents are pending for the contents shown and therefore are confidential until the paper is published.

References

[1] Besl J.P., Jain C.R.: Segmentation through variable-order surface fitting; IEEE PAMI 10, 167-192, 1988.

[2] Hertling, P., Hog, L., Larsen, L., Perram, J.W., Petersen, H.G.: Task Curve Planning for Painting Robots - Part I: Process Modeling and Calibration; IEEE Transactions on Robotics and Automation 12(2), April 1996, 324-330.

[3] Hoover, A., et.al.: An experimental comparison of range image segmentation algorithms; IEEE PAMI 18(7), 1-17, 1996.

[4] Jiang X. Y., Bunke H.: Fast segmentation of range images into planar regions by scan line grouping; Machine Vision Applications 7, 115-122, 1994.

[5] Jiang X. Y., Bunke H.: Edge detection in range image based on scan line approximation; Computer Vision and Image Understanding 73(2), 183-199, 1999.

[6] Kwok, K.S., Louks, C.S., Driessen, B.J.: Rapid 3-D Digitizing and Tool Path Generation for Complex Shapes; IEEE ICRA, pp.2789-2794, 1998.

[7] Mule, B., Schmid, D., Berger, U.: Robots for rapid prototyping and for tooling of high volume workpieces; Industrial Electronics Society, Proceedings of the 24th Annual Conference of the IEEE IECON '98, Vol.4, pp.1941 -1945, 1998.

[8] Olsen, M.M., Petersen, H.G.A: A new method for estimating parameters of a dynamic robot model; IEEE Transactions on Robotics and Automation 17(1), pp.95 -100, 2001.

[9] Sheng, X., Krmker, M.: Surface Reconstruction and Extrapolation from Multiple Range Images for Automatic Turbine Blades Repair; IEEE IECON Conference, Vol.3, pp.1315-1320, 1998.

[10] Tse, W.C., Chen, Y.H.: A robotic system for rapid prototyping, IEEE International Conference on Robotics and Automation Vol.3, pp.1815 -1820, 1997.