Automatic Path and Trajectory Planning for Robotic Spray Painting.

Topic: Robotics in Production

Alessandro Gasparetto,
Dep. of Electrical, Mechanical, and Management Eng. University of Udine, Universitá di Udine, Italy

Renato Vidoni
Faculty of Science and Technology, Free University of Bozen-Bolzano, Italy

Daniele Pillan
CMA-Robotics, Pradamano (Udine), Italy

Ennio Saccavini
CMA-Robotics, Pradamano (Udine), Italy

Abstract

In this paper, a new automated optimum path and trajectory generation system for robotic painting process is presented. Usually, a direct self-learning manual method is usually adopted, i.e. the operator drives the robot manually through a complete spraying cycle. Recently, in order to-avoid this time-consuming procedure, CAD-based or "acquire-compare-recognize" methods have been developed. Here, a general technique that avoids the need for manual programming or CAD drawings and allows to obtain an optimal path and trajectory by using the graph theory and operative research methods is developed. After an image acquisition phase, a partitioner splits the object into a set of primitives, then the proposed algorithm is run on the graph in order to generate the optimal path; then, a optimum trajectory planning phase is performed. Kinematic and dynamic simulators of the CMA Robotics robots have been developed and validated in order to allow a correct planning and evaluation of the results. The new path planning algorithm has been implemented in Matlab and Visual-Studio.NET environments. Practical applications are presented.

1 Introduction

The spray painting production is widely used in industry and becomes everyday more competitive. Current research deals mainly with automation of the process, optimization and quality of the paint. Indeed, the uniformity of the paint thickness on a product strongly influences its quality and such a feature is typically determined by the robot tool path. As a consequence, in spray painting processes, tool path and trajectory planning with uniform material distribution and minimum wasted paint remain challenging research topics.

In order to specify a path for a robotic manipulator, a manual tool planning, i.e. a teaching method, is usually performed. An operator teaches the robot painting path by moving the end-effector around the parts to be painted and, in the meantime, positions and orientations are recorded by the robot controller. The overall procedure requires a lot of time. In order to obtain a good painting result, the operator has to carry out extensive tests on a work cell and the final path is often the result of a trial and error approach.

In the last years, automatic and semi-automatic methods that realize optimal spray gun paths or trajectories, thus avoiding the manual self-learning programming procedures, have been developed. Computer Aided Design (CAD) tool path planning methods are extensively used in automotive industry. A model and pieces CAD-drawing [1, 2, 3] are directly exploited to create a path for the gun and product manufacturing process. This solution, that allows to reduce dramatically the operation time and the human labor, needs a CAD-drawing of the pieces that is not always available in many industrial applications (e.g. painting of windows, doors, chairs, home appliances). Thus, it is rarely adopted in small industries also because such CAD software packages can be expensive and no trained and qualified personnel is available.

A different approach for the automation of the painting process deals with the exploitation of a vision system: the image of the piece is acquired and matched with one of the patterns stored in a dynamically updated library [4, 5, 6]. The technical challenge is the detection of the geometry of the part and the automatic generation of the robotic painting path and trajectory. When small different lots or particular and new shapes are to be painted (e.g. customized doors, windows and gates), such a method can fail since no pre-loaded forms are present in the library and errors occur.

In this work, the improvement of the efficiency of the CMA-robotics painting process has been evaluated in order to generate continuous and smooth motions of the spray paint gun by taking into account the optimization of the overall painting path and trajectory definition.

The robotized painting process and system developed in this work exploits robot vision to define the structure of the object to be painted, in motion on a conveyor, by detecting its front edge and primitives in an automatic way. Thus, thanks to a partitioning phase, the issue of how to find an optimal painting path is addressed studying a collection of basic geometries. In such a way, any kind of shape can be treated and an optimal aesthetic result and a minimization of the wasted paint can be reached by planning a continuous and smooth robot motion. This search for the optimal path is done by means of the set of basic geometric coordinates result of the partitioning phase and of the representation of the object as a graph.

A graphical interface allows the user to choose the optimal path between different available solutions in an automatic or semi-automatic way. These solutions are related to different criteria such as painting priority (e.g. internal or external elements first, plans or lines), path continuity and optimal fluidity. The obtained path is tested using a simulator, integrated into the developed system, that implements a model of the anthropomorphic robot to which the painting task is assigned.

The availability of the path allows to plan the trajectory by considering or setting velocity, spray gun opening/closing phases, spray distance, flux and color of the paint. Since constant speed is advisable to obtain a better quality painting, trapezoidal speed profile or polynomial primitives that allow at least the continuity of the velocity are chosen.

Thanks to a dynamic simulator of the robot, the overall planned trajectory can be evaluated and the required energy consumption can be forecast and eventually minimized. In Fig.1 the structure of the automated process is presented and in the following the main phases of the process are described.

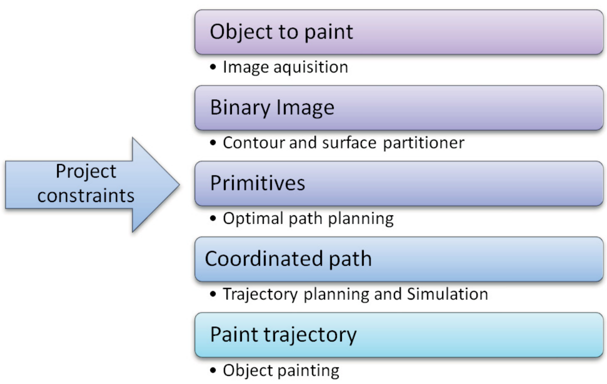

Figure 1: Layout of the (semi)-automated process

1.1 Image acquisition

To define the structure of the object to be painted automatically, i.e. detect the edges and extract primitives, pattern recognition techniques and/or artificial vision can be used. In this work, objects are detected using a barrier sensor which working principle is based on the interruption of the light beam (see Fig.2). These sensors are independent from the reflection and from the color of the object and, thus, are the ideal solution for this kind of task also taking into account the fact that objects are in motion on a conveyor.

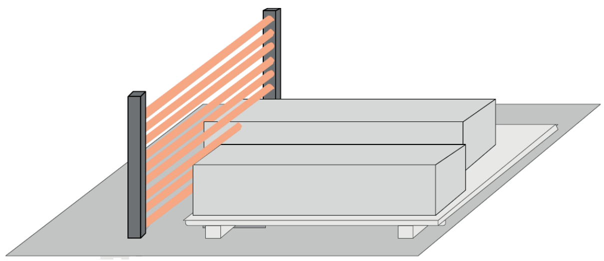

Figure 2: Barrier sensor.

At the end of the acquisition phase a binary image is available. This is then elaborated by a software module able to recognize objects and their contours and surfaces. Indeed, this partitioner extracts from each distinct object a set of primitives (Fig. 3), defined as the different simple entities that constitute the object.

Two main kinds of primitives are recognized: segments or curves that form the edge and segments, curves and planes called "frontal" primitives.

The width of the painting stroke defines the limit between segments and planes. Indeed, a plane consists of a rectangle with both sides longer than the painting stroke width and is composed of segments of maximum width parallel to the plane longer edge. The direction of the strokes is carried along the longer side of the plan or alternatively can be chosen by the user (vertical or horizontal). The distance between the strokes is determined by interface graphics.

A file containing the characteristics of the object is then stored and sent to the Mapping phase.

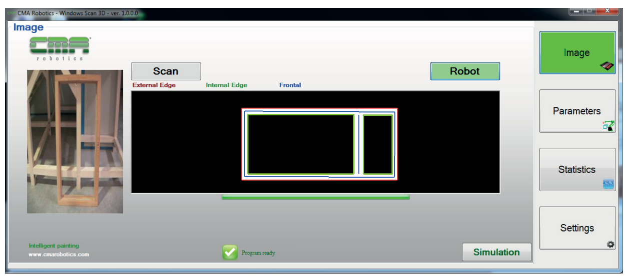

Figure 3: Image elaboration

1.2 Mapping

Mapping is constituted by the primitives of the object. The object to be paint is considered as a graph [7], a useful representation because it allows to easily determine the continuity of the primitives. Indeed, segments or curves are considered as arcs and the coordinates that constitute them are nodes. Graphs can be represented extensively as a list of all the elements, i.e. nodes and vertexes. Different representations are possible, e.g. adjacency matrix, adjacency list, incidence matrix, path matrix, circuit matrix, cut matrix and distance matrix. A constraint of the project consists of avoiding multiple passes along the same primitive, it is therefore important to use a representation that underlines the arcs. A B-type matrix of incidence representation has been chosen in order to easily evaluate if multiple passes along the same primitive are planned. The incidence matrix has dimensions $n \times m$ where the number of rows n is equal to the number of nodes and the number of columns m is equal to the quantity of arcs. Each arc is represented in the columns of the incidence matrix with two elements that assume a value equal to one in correspondence to the nodes that constitute. Clearly, the sum of the columns of an arc is equal to two.

1.3 Path and Trajectory Planning

An optimal path for the painting of an object of any form and dimension that satisfies the requisites imposed by the practical applications is the result of this phase.

The project requisites depend on the application. The most important and common are: cover the whole figure, avoid manifold passages along the same primitive, allow constant speed during the painting, allow the choice of the starting point and search continuity of painting provided that the angle formed by two primitives with an extreme in common is greater than a critical fixed angle.

Among the algorithms proposed by the operational research (e.g. shortest path, minimum length algorithms [8, 9]) it is hard to find out a solution suitable for all the previously mentioned constraints.

Eulerian cycles (*Chinese Postman Problem*) could be the optimal solution, indeed an Eulerian path is a path in a graph which visits each edge exactly once but for having an Eulerian path the graph must be connected and has to show particular features. Thus, in order to have a planner independent from the form of the considered object, from the number of incidental arcs in every node and from the choice of the type of optimal path (maximum path or search of the fluidity), an ad-hoc iterative algorithm based on the graph theory has been realized using the characteristics and the requisites of the project (see Fig. 4 as an example). A search process that specifies which rules apply and when to stop the calculation if a termination condition is met has been implemented in Matlab and Visual Studio.Net environments. Firstly the algorithm find the nodes of the segments and arcs that constitute the object and searches the primitive belonging to the edge. After that, the graph is subdivided in subgraphs constituted from primitive that guarantee the continuity of the path and the imposed requisites. Then, optimization is searched. Different requests con be chosen:

- Contour and minimum angle priority (see Fig.4). This function imposes a priority in the execution of the contour primitives and, if more than one contour arc insist on a node, the one that forms an angle nearest to 180° with the previous primitive is selected. This function is cyclically repeated until all the edge primitives are considered.

- Contour and maximum length priority. This function imposes a priority in the execution of the contour primitives privileging the maximum path length.

- Path continuity and minimum angle priority. This function privileges the path continuity starting from the contour primitives.

- Path continuity and maximum length priority. This function recursively searches the primitives that allow the maximum length path.

- Maximum length sub-graph. This function searches the primitive sequence that forms the maximum length path. If the same maximum length is reached by more than one sub-graphs, the one with the first node nearest to the chosen starting point of evaluation is selected.

Specific settings can be chosen by the operator and automatically checked when optimal path is searched. Among these critical angle, end-effector distance from the edge of the object at the beginning and end of the path, z axis and Euler angles of the spray gun, search mode for internal and edge primitives are the most important.

Finally, the optimal path found is stored in a file and sent to the trajectory planner module. The trajectory is planned at constant speed in order to obtain a better quality painting. For this reason a trapezoidal speed profile which requires constant acceleration in the starting phase, a cruising constant speed and a constant deceleration in the final phase is usually adopted.

The spray is modeled as a cone with an ellipse base. The parameters of the ellipse are calculated from the fan parameters given by the gun constructor. The value of the cone height corresponds to the spray distance from the gun to the object to be painted. To optimize the time of painting off all the objects the algorithm finds the minimum length in transition from one component to another and from different objects.

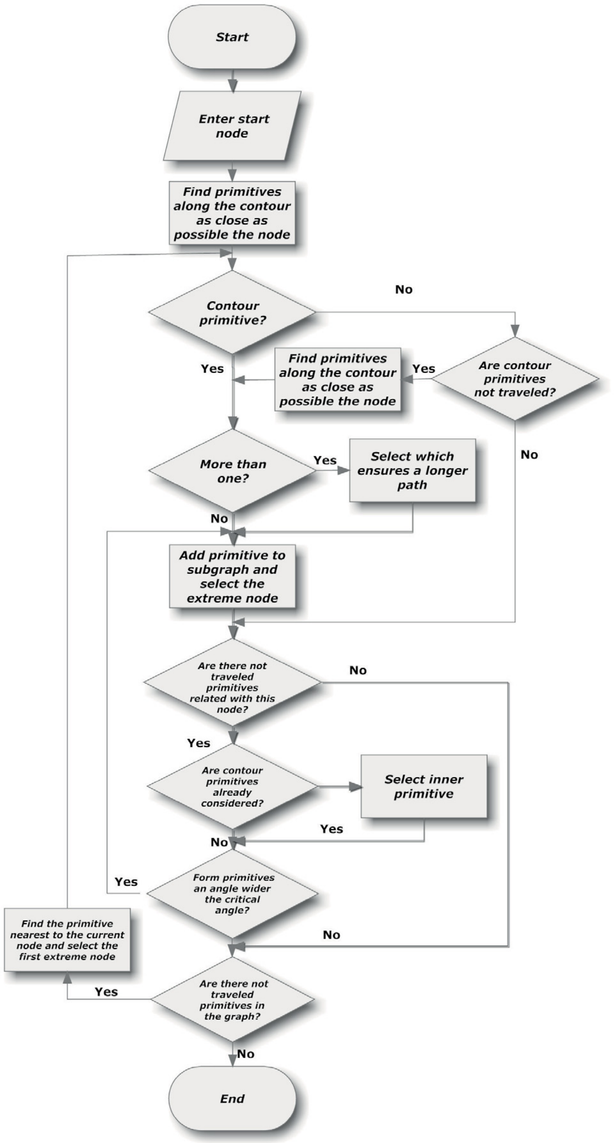

Figure 4: Contour and minimum angle priority algorithm.

1.4 Kinematic and dynamic models and simulators

The optimal path and trajectory are sent to the ad-hoc developed simulator module that reproduces the kinematics and dynamics of the anthropomorphic robot to which the painting task is assigned.

Key concepts of robotics [11] such as direct kinematics, inverse kinematics, differential (first and second order) kinematics, statics and dynamics have been evaluated and implemented. The dynamic simulator has been developed taking into account the geometrical and inertial parameters of the robots in use; in particular, a Newton-Euler approach has been implemented. Thanks to this simulator, a direct evaluation of the required effort for the chosen trajectory can be performed.

In order to show the effectiveness of the dynamic simulator, a comparison between simulated and measured torques for the second motor of the anthropomorphic robot under a painting trajectory is presented in Fig. 5. As can be seen, the simulator keeps well the dynamics of the real system and thus, the simulated results can be considered as realistic.

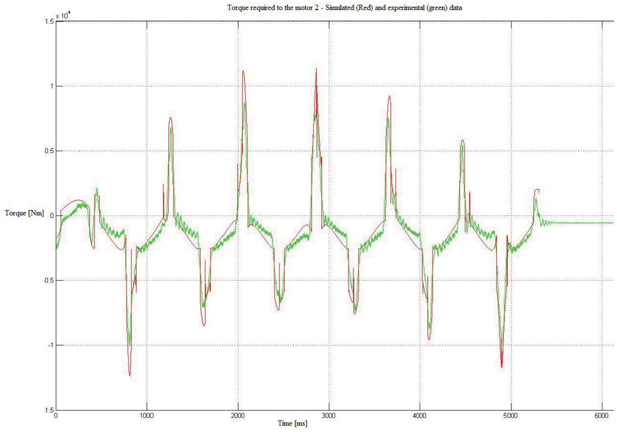

Figure 5: Simulated and experimental torque comparison: red (simulated); experimental (green)

Moreover, the simulator module uses the same interface present on the programming devices of the robots. A mesh creator converts the digital image, that contains the objects to be painted, in a polygon meshes. Meshes are a collection of vertexes and faces that define the shape of an object and simplify the graphic rendering.

In the simulation, the quantity of paint applied on the whole surface of the object is highlighted by means of the variation of different colors. In Fig. 6 the graphical interface of the simulation module is presented.

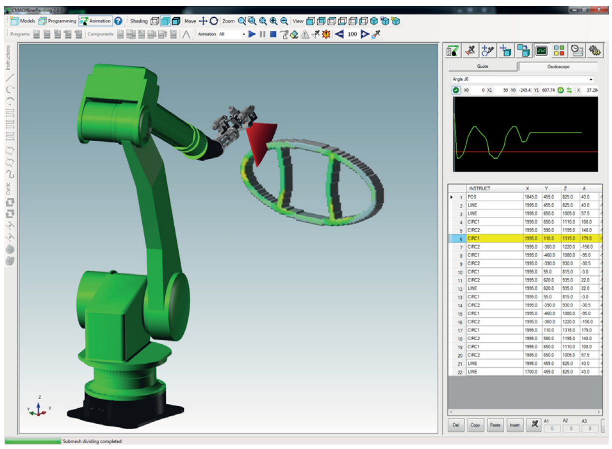

Figure 6: Graphical interface of the simulator module

The path of the end-effector can be simulated highlighting the strokes. Alternatively, the painting under examination can be reproduced by highlighting the on/off spray gun and the object lines painted.

2 Implementation

The overall idea has been extensively tested and validated. First of all, the algorithm has been experimented on a CMA Robotics' robot, the GR-6100: an anthropomorphous manipulator suitable for painting in line with the spray gun on the end-effector. The basic steps of the process have been performed: artificial vision acquisition, partitioning, mapping, path searching, trajectory planning and dynamics and painting simulation, and, finally, physical painting. The object is moved by means of an aerial conveyor at 5 m/s; the optical barrier resolution is less than 1 cm. The developed algorithm requires less than one second to obtain a solution (about 0.62 seconds with a PC Intel Core 2 Duo 2.40GHz, ram 4GB).

Usually, the manual self learning method takes minutes for moving the robots end-effector around the piece to be painted, to store points and create the program, while the developed system takes few seconds. Moreover, the path planning step allows the user to select different optimal paths without a process interruption. Thus, the overall process is much more efficient with respect to the traditional manual one.

After the experimental validation, the system has been implemented and integrated into the CMA-Robotics painting systems. Fig.s 7 and 8 are taken from a developed application.

Figure 7: Image acquisition phase

Figure 8: Painting phase

3 Conclusions

An automated optimum path and trajectory generation system for robotic painting process is presented.

The process structure, the automation and the implementation steps are described. Graph theory and operative research techniques are applied to provide a general and optimal solution for the path planning problem. The path optimization has been done considering the coordinates of the primitives such as nodes, segments and curves like arcs and creating an algorithm able to take into account several constraints posed by robotized painting.

The system is easy-to-use and flexible allowing to treat different and special shapes, and to fill the gap between the image acquisition and trajectory planning process leading a new, fast, general and automatic mechatronic system for robotic painting purposes.

References

[1] Li, X., Landsnes, Ø.A., Chen, H., Sudarshan, M-V, Fuhlbrigge, T.H., Rege, M-A., *Automatic Trajectory Generation for Robotic Painting Application* In Proc. of Robotik 2010.

[2] Chen, H., Fuhlbrigge, T., and Li, X., *Automated industrial robot path planning for spray painting process: A review* In Proc. of the 4th IEEE Conference on Automation Science and Engineering, 2008.

[3] Chen H, Fuhlbrigge T, Li X, *A review of CAD-based robot path planning for spray painting*. Industrial Robot: An International Journal, Vol. 36 Iss: 1, pp.45 - 50, 2009.

[4] Vincze, M., Pichler, A., Biegelbauer, G., Husler, K., Andersen, H., Madsen, O., and Kristiansen, M., *Automatic robotic spray painting of low volume high variant parts*. In Proc. of the 33rd ISR: International Symposium on Robotics, 2002.

[5] Vincze, M., Biegelbauer, G., and Pichler, A., *Painting parts automatically at lot size one*. In Proc. of the International Workshop on Robot Sensing, 2004 (ROSE 2004), 2004.

[6] Pichler, A., Vincze, M., Andersen, H., Madsen, O., and Hausler, K., *A method for automatic spray painting of unknown parts*. In Proc. of the 2002 IEEE International Conference on Robotics and Automation (ICRA ' 02), 2002.

[7] University of Liverpool, *Graph theory in operations research*, T.B. Boffey, Hong Kong 1982,

[8] Sebastian Thrun, Wolfram Burgard, Dieter Fox *Probabilistic Robotics*, Mitt Press, Massachusetts 2005,

[9] Mauro Cerasoli, Franco Eugeni, Marco Protasi *Elements of discrete mathematics*, Zanichelli, Bologna 1992,

[10] Edward Minieka *Optimization Algorithms for Networks and Graphs*, Wilbur Meyer, New York 1978,

[11] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo, *Robotics: Modeling, Planning and Control*, Milan, McGraw-Hill 2008