

Using Targets in Roboguide to Visualize Path Planning of a FANUC Robot

Ming Li

Department of Computer Science, Engineering and Physics
University of Michigan-Flint

Abstract

Traditionally, a tool path of a robot is taught by an operator manually using a teach pendant to record all points along the path. Though this method is still most commonly used, offline programming software becomes more competitive since they are cost efficient. Roboguide, developed by FANUC, is a family of offline simulation software products which can simulate operations in work cells and construct tool paths automatically. The Targets feature of Roboguide can be utilized to visualize and verify path planning for robot arms. In this paper, a FANUC LR Mate 200id/4s was used to simulate how a human being would draw through the Targets feature. The robot was designed to draw a portrait constructed through connections of a series of points. These points were introduced to the work cell as targets, and the robot followed these targets to complete the drawing. In this case, using Targets feature to generate a tool path is much faster than both the tedious teach pendant method and the CAD to Path programming.

Introduction

Path planning is very important for painting robots and welding robots which are widely used in automotive manufacturing. The features of the planned path can greatly affect the quality of products (Sheng, 2000). However, to generate a tool path can be tedious and time-consuming. Traditionally, a tool path is taught by an operator manually, i.e., the operator jogs the robot through the trajectory and records points one at a time using a teach pendant. After all points along the path have been learned, the robot can play back to accomplish a given task. Though this method is still most commonly used (Schraft 2006), offline programming software has been developed to construct a tool path automatically.

Offline programming methods create a virtual 3D work cell in which the position and orientation of a robot can be taught by a virtual teach pendant, by dragging and dropping with a mouse, or by inputting position data directly (Pan, 2012). Teaching programs can be developed, simulated and validated in the virtual work cell, and then be uploaded to the real robots to execute the operations. Offline programming method is fast, cost effective and can significantly reduce down time (Kumičáková, 2015). Its usage has seen growth year by year.

Roboguide is a robot simulation software developed by FANUC, which includes a family of products such as HandlingPRO, PaintPRO, PalletPRO, and so on (FANUC, 2016). With Roboguide,

the virtual 3D work cell can be set up easily through selecting and modifying various parts from its extensive library (Sakakibara, 2003). It also can perform collision detection, reach verification, cycle time calculation, and other operations. What is notable about Roboguide is its CAD to Path programming, which can import CAD data and generate a path based on it (Li, 2007). CAD to Path programming takes advantage of the information stored in the IGES file of a CAD model, and greatly improve the process to construct a trajectory of an end of arm tooling (EOAT) (Veiga, 2013). However, CAD to Path is based on a CAD model which needs to be created or modified if it is not in the current library. Then the tool path has to be generated through Edge, Surface, Curve and so on, as shown in Figure 1. Thus, it may be suitable for complicated 3D tool paths which cannot be conveniently constructed through other ways.

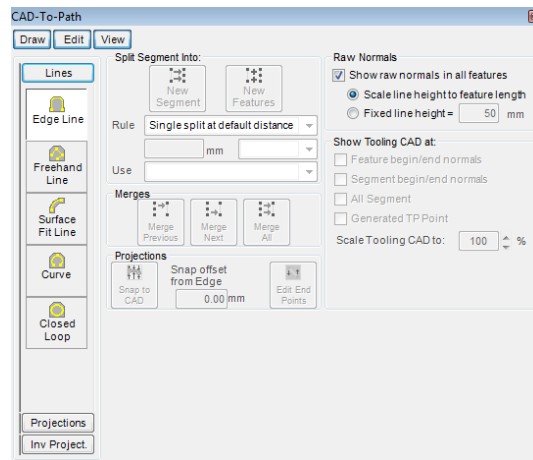


Figure 1. Roboguide CAD-To-Path programming

For a 3D path which has less change of orientations or the points along it are already known, Targets feature of Roboguide might be a better choice. Targets are the position data representing points in a work cell (FANUC, 2015). Targets include the position and orientation information (x, y, z, w, p, r), which indicates where the EOAT will be moved to and how it will be moved. Targets can be attached to an object, and move when the object moves (FANUC, 2015). They also can be shared among multiple robots. Roboguide can verify if a target is reachable for a certain robot or not, and indicate the target with green if it is reachable or red if it is not reachable. Roboguide also provides features that can search close by reachable points. After all the targets have been validated, TP programs can be automatically created based on the choices of motion, speed, termination, etc. Then, simulation of the motion of EOAT can be executed to visualize and validate the tool path as shown in Figure 8.

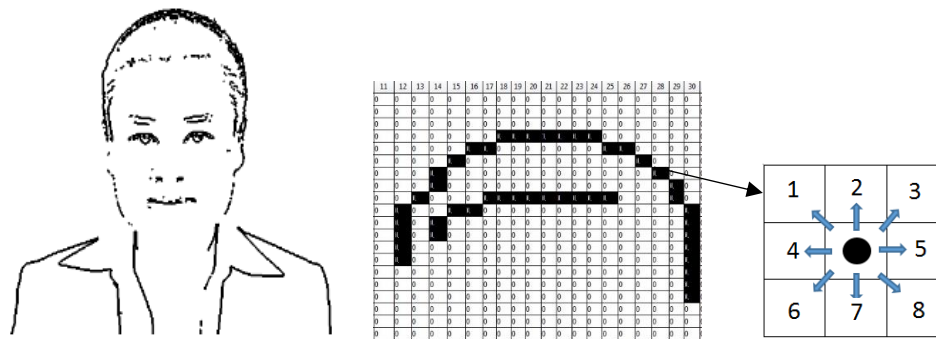
In this paper, Targets feature of Roboguide was utilized to develop a robot which can draw a portrait. Instead of printing out pixels of a raster image like a printer, this robot was designed to simulate how a human being would draw. The robot recognized the drawing as a big picture, and identified the main composition with longer lines and then filled in details with shorter lines. The drawing was generated through connecting a series of points which were input as targets in the work cell. When the robot moved along these targets in order, a portrait drawing could be completed.

Methods

Image Processing

A Matlab program was developed to create a portrait for the aforementioned robot to draw using according functions provided by Matlab Image Processing Toolbox. A picture of a person was first converted to a gray scale picture. After comparing several edge detecting methods, the Prewitt method was selected to detect the edge of the portrait. Then a binary picture was generated as shown in Figure 2 a. In this picture, the value of each pixel is either 1 or 0, with 1 representing a point along the edge and 0 representing a blank point. The robot was designed to draw the edge like a human being.

The robot scanned the whole binary picture to identify the main composition. After the robot identified the first point of an edge segment, a recursion subroutine would figure out all the pixels connected with it and generate a drawing line starting from the aforementioned point, as shown in Figure 2 b and c (Erik, 2007). Thus, longer lines representing main composition were constructed first and shorter lines representing details were constructed afterwards. When the robot followed the sequence to draw these longer and shorter lines, it imitated how a human being would draw.



a. Binary picture

b. Pixels representing edge

c. Recursion around edge

Figure 2. Image processing

The drawing robot was developed using a 6 degree of freedom robot, the FANUC LR Mate 200id/4s shown in Figure 8. The position and orientation of a EOAT are determined by both position and orientation information, i.e, x, y, z, w, p, r coordinates of a Cartesian coordinate system attached to the base of the robot. x, y, z are the position values along the according axes; and w (yaw), p (pitch), r (roll) are the rotation angles of x, y and z axes.

The position information was derived from the binary picture. The pixels of the portrait were represented by a matrix, after the previous procedure, the pixels of the drawing lines had been recorded according to the sequence. The value of the row and column of the pixels were then mapped to a piece of paper to construct the x and y value as indicated in Figure 3. The directions of x, y axes are from the robot's point of view. Based on the end of the arm tooling (in this case, a gripper) and the pen utilized, the z value could be defined. Since it's a 2D drawing, and the pen could be perpendicular to the paper, the w, p and r could be easily determined. In this case, w, p, r were selected as 0, 180, 0. Thus, an array of x, y, z, w, p, r coordinates constructed the path which the EOAT would move along. The array needed to be saved as a .CSV file to be imported into the Roboguide virtual environment, which is shown in Figure 4.

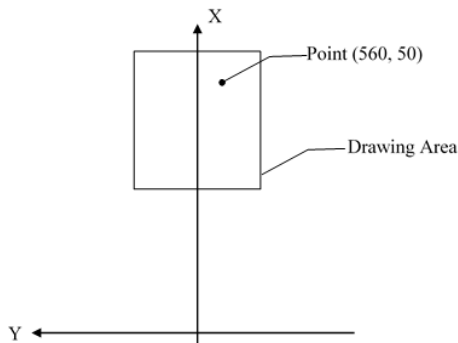


Figure 3. Map the portrait to the drawing area

	A	B	C	D	E	F
1	613	8	-230	0	90	0
2	613	7	-230	0	90	0
3	613	6	-230	0	90	0
4	613	4	-230	0	90	0
5	613	3	-230	0	90	0
6	613	2	-230	0	90	0
7	613	1	-230	0	90	0
8	613	0	-230	0	90	0
9	612	-1	-230	0	90	0
10	612	-2	-230	0	90	0
11	613	-3	-230	0	90	0
12	613	-4	-230	0	90	0
13	613	-6	-230	0	90	0
14	613	-7	-230	0	90	0

Figure 4. CSV File

Targets Feature

The teaching points generated and saved in the previous step needed to be input as Targets of the work cell as illustrated in Figure 5. They were then introduced into a target group as members to generate TP programs as shown in Figure 6. However, not all of the targets could be reachable by the robot because of the limitation of the robot or the previous position and orientation.

As shown in Figure 7, the according position and orientation information was highlighted as Targets in the 3D work cell. Green and Red color indicate if the according point was reachable or not. If the point was green it meant the point could be reached, otherwise the red point could not be reached. If there was any red point, the generation of the path failed. The constructed path needed

to be revised. Red points could be removed or replaced with reachable close by points. Thus, the tool path generated could be verified. After all adjustments had been made, the TP program could be generated through the Generate Feature TP Program button as shown in Figure 6. Then the TP program was loaded to the robot. When the robot followed all the teaching points, it could draw a portrait like a human being.

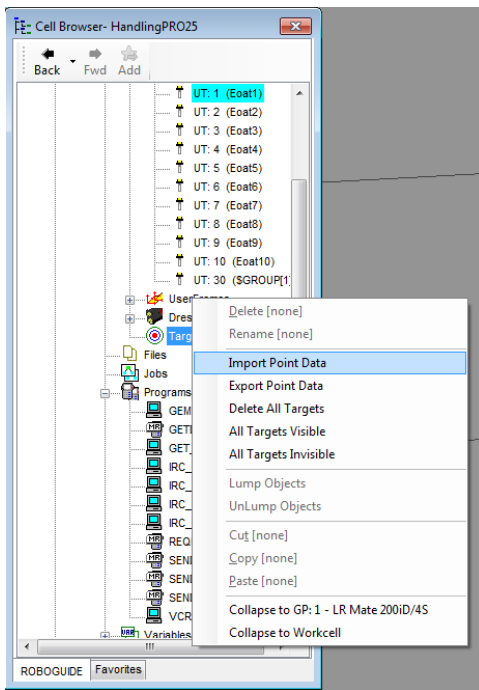


Figure 5. Import point data as Targets

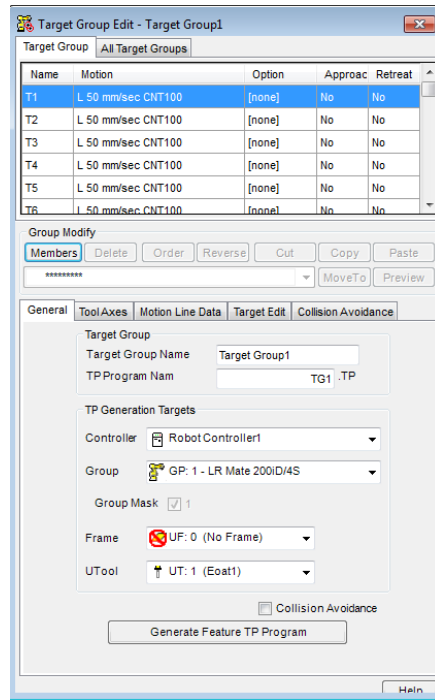


Figure 6. Target Group Edit dialogue

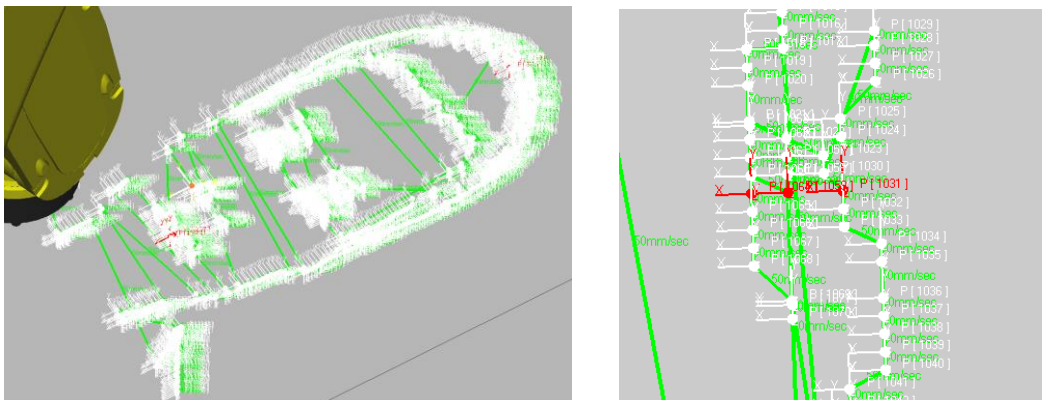


Figure 7. Teaching points highlighted as Targets

Results and Discussion

Figure 8 shows the virtual work cell and the simulation of the execution of the TP program. Figure 9 shows the portrait drawing by the robot.

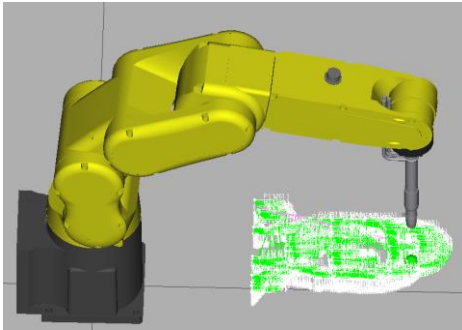


Figure 8. Virtual work cell



Figure 9. Portrait drawn by robot

From the project, the Targets feature can be utilized to dramatically speed up the generation of a 3D tool path which has little change of orientations or the points along the path are already known. In this case, using Targets feature to generate a tool path is much faster than both the tedious teach pendant method and the CAD to Path programming. If teach pendant method was used, the operator would need to jog through the points along the drawing one by one, and record the path for the robot to play back. And if another person's portrait needed to be drawn, the tedious procedure would have to be repeated again. If the CAD to Path programming was used, the CAD model of the portrait would need to be built up and the programmer would need to move the mouse along the edge segments to generate the drawing path. However, if the orientation of a EOAT is unknown or continually changing, targets cannot be created quickly and effectively.

Reference

- Erik. (2007). *Erik's XY-Plotter*. Retrieved from http://www.convict.lu/Jeunes/ultimate_stuff/Erik_s_xy_plotter/E_xy_plotter.htm
- FANUC. (2015). *SimPro_Master_Help*. ROBOGUIDE [computer software] help file.
- FANUC. (2016). *ROBOGUIDE - FANUC Simulation Software*. Retrieved from <http://robot.fanucamerica.com/products/vision-software/ROBOGUIDE-simulation-software.aspx>
- Kumičáková, D., & Rengevič, A. (2015). *Automation of manufacturing technologies with utilisation of industrial robots*. Applied Computer Science,11(3).
- Li, L., Li, X., Zhou, X., & Yue, J. (2007). *Study of Off-Line Programming System of Arc Robot Based on the Software of ROBOGUIDE*. In Robotic Welding, Intelligence and Automation (pp. 401-408). Springer Berlin Heidelberg.

- Pan, Z., Polden, J., Larkin, N., Van Duin, S., & Norrish, J. (2012). *Recent progress on programming methods for industrial robots*. *Robotics and Computer-Integrated Manufacturing*, 28(2), 87-94.
- Sakakibara, S. (2003, September). *The latest robot systems which reinforce manufacturing sector*. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on* (Vol. 2, pp. 2878-2883). IEEE.
- Schraft, R. D., & Meyer, C. (2006). *The need for an intuitive teaching method for small and medium enterprises*. *VDI BERICHTE*, 1956, 95.
- Sheng, W., Xi, N., Song, M., Chen, Y., & MacNeille, P. (2000). *Automated CAD-guided robot path planning for spray painting of compound surfaces*. *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on* (Vol. 3, pp. 1918-1923). IEEE.
- Veiga, G., Malaca, P., & Cancela, R. (2013). *Interactive Industrial Robot Programming for the Ceramic Industry*. *International Journal of Advanced Robotic Systems*, 10.
- Yang, B., & Chen, C. Y. J. (2011). *Development of a Robotic Platform for Teaching Model-Based Design Techniques in Dynamics and Control Program*. In *American Society for Engineering Education*. American Society for Engineering Education.