Algorithm for Robot Writing using Character Segmentation

Salman Yussof, Adzly Anuar, Karina Fernandez
Universiti Tenaga Nasional
Jalan Kajang-Puchong
43009 Kajang, Selangor, Malaysia
{salman, adzly}@uniten.edu.my
karina.fernandez@shell.com

Abstract

Currently, there are many ongoing researches that are targeted at making robots more human-like. One of the tasks that can be done by humans easily but is difficult to be done by robots is writing. In this paper, we are presenting a flexible algorithm that can allow a robot to write. This algorithm is based on character segmentation, where the main idea is to store character information as segments and the segment information can then be used by the robot to write. We have also developed a sample application using the proposed algorithm to allow a Mitsubishi RV-2AJ robotic arm to write English characters and numbers. Through our experiment, it has been proven that the algorithm developed is able to allow the robotic arm to write.

1. Introduction

Robots were initially designed to assist humans in doing work. Robots are really good at doing work that requires repetitive actions [2]. Since robots can work continuously and tirelessly, the use of robots is usually desirable in places where continuous operation is required, such as at the production lines. However, robots cannot do every single work that can be done by humans. There are tasks that can be easily done by humans but cannot be easily done by robots. One example of such a task is writing.

Writing is a task that almost every human being can do easily. From a person's point of view, writing requires him or her to recognize a pattern, such as the shape of a character, and then reproduce the same pattern on a medium, such as a piece of paper. The process seems awfully simple to a human. This is because our brain is very good at recognizing and producing simple patters. However, to enable a robot to do the same thing is not an easy task. All the processes that are automatically done by our brain need now be carefully programmed into the robot.

To enable a robot to write, the robot should have an arm-like component together with a hand or gripper that can hold a pen or pencil. Various researches have been done in studying the mechanism and control of robotic hand that can perform the task of writing [5, 6, 7]. Once the correct structure is available, writing can then be achieved by attaching a pen or a pencil to the robotic hand or gripper and moving the arm in such a way that the correct writing can be produced.

In this paper, we are proposing a character-segmentation-based algorithm to enable a robot to write characters. The core of this algorithm is to store the character information as segments and the segment information can then be used by the robots to write. For the scope of this paper, we are concentrating on segmenting Latin character set (the alphabets that we use in English and Malay language). This segmentation-based algorithm requires each character to be divided into several segments. Each of the segments will then be stored as a single movement. This paper will try to answer the following three main issues:

a) How to segment the characters?
b) How to store the segments?
c) How to use the stored segments?

2. Character Segmentation

Alphabets in Latin character set consist of two main types of segments, mainly a straight line and a curve. Although different characters are made up of segments of different sizes, all alphabets in Latin character set are made of these two types of segments. This fact has a significant impact on our goal to enable robots to write. Since Latin characters are only made of straight lines and curves, any robot that can draw straight lines and curves has the ability to write Latin characters.

Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)
0-7695-2316-1/05 $20.00 © 2005 IEEE

What is needed by the robot to properly write the characters is a proper guideline on the type and size of the segments that need to be drawn for a particular character. This is where segmentation comes in.

Character segmentation divides a character into segments that consist of straight lines and curves. By doing so, we are dividing a character into components that can be drawn by a robotic arm. For example, character 'n' is segmented into three segments. The first and third segments are straight lines, while the second segment is a curve. Figure 1 shows the segmentation of three different characters, 'a', 'c' and 'n'. As can be seen, different characters can have different number of segments.

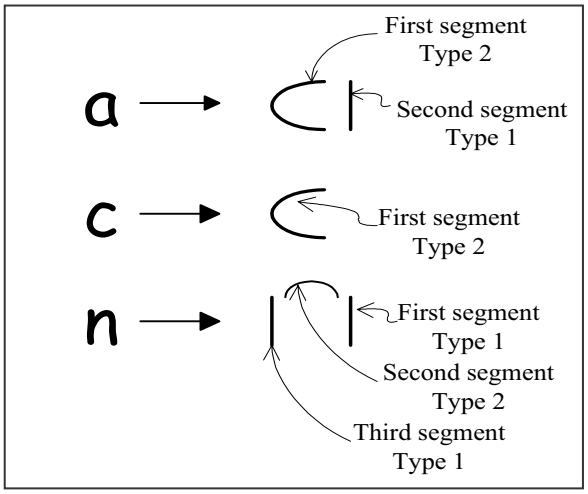

Figure 1. Segmentation on letters 'a', 'c' and 'n'

There are several types of robot movement commands that can be utilized to instruct the robot to produce or write the characters. However, due to the simplicity of the character segmentation design, the use of two types of movement command is enough to produce all the different characters.

a) Linear interpolation movement. In this movement type, the end of the robot hand is moved with linear interpolation to the designated position.
b) Circular interpolation movement. The robot moves along an arc designated with three points using three-dimensional circular interpolation.

The first type (Type 1) is used to draw straight lines while the second type (Type 2) is used to draw curves.

3. Segment Storage

Once the characters have been segmented, the segment information need to be stored in a database. The most straightforward way to store the segment information is to use a flat table. An example of such a table is shown below in Table 1 below.

Table 1. Character segment information

Character	No. Of Segments	Segment 1		Segment 2	
		Type	Points	Type	Points
a	2	1	a1, a2	2	a3, a4, a5
b	2	1	b1, b2	2	b3, b4, b5
c	1	2	c1, c2, c3	-	-
. . .	. . .				

Each of the characters consists of one or more segments. The database should store information of each segment. There can only be two types of segments, either a line or a curve. A line will have two points associated with it, which are the start point and the end point. A curve, on the other hand, have three points associated with it, which are the start point, the end point and the peak-of-the-curve point.

Assuming that the robotic arm is going to write only on a horizontal, flat 2-dimensional surface, the points are represented using the (x, y) coordinate format. Depending on implementation, the actual number stored for the points can have several different meanings. For example, the points can represent any one of the following:

a) The actual coordinates that the robotic arm is going to use.
b) Displacement from a particular (x, y) coordinate.
c) Displacement from the current position of the robotic arm.

Other definitions for the points are also possible. If the need arises, the 3-dimensional coordinate format (x, y, z) can be used instead of the 2-dimensional coordinate format.

Even though using a flat table is the simplest way to store the segment data, it is not an efficient and practical method to store the data. This is because different characters have different number of segments. For example, character ‘b’ has two segments, while character ‘c’ has only one segment. Therefore, a table used to store the segment data must reserve enough space to allow the maximum number of segments possible. For characters that have less than the maximum number of segments, there will be unused columns in the table. This would lead to inefficient use of storage space. A more robust and practical approach would be to use a database.

There can be many ways to design a database to store the segment information. Here, we propose one database design as an example to illustrate the possibility of storing the segment information in a database.

The database that we are proposing consists of three tables. The first table is called the character table. It contains information regarding the number of segments and the number of points that a character has. An example of the table is shown in Table 2 below.

Table 2. Example of a character table

Character	No_of_segments	No_of_points
a	2	5

The second table is called the segment table. It contains information regarding types of segment for each segment in the character. An example is shown in Table 3 below.

Table 3. Example of a segment table

Character	Segment_no	Segment_type
a	1	Line
a	2	Curve

The third segment is called the point table. It contains the information regarding the location of each point in the character. An example is shown Table 4 below.

Table 4. Example of a point table

Character	Segment_no	Point_no	Points
a	1	1	a1
a	1	2	a2
a	2	1	a3
a	2	2	a4
a	2	3	a5

It can be verified that the information stored in the database is similar to the one presented in the flat table above. However, the database format is more practical, efficient and robust. An Entity Relationship Diagram (ERD) of the database is shown in Figure 2.

Of course, other database designs are also possible. The database can also be extended to include other information such as font type, kerning or robot instruction information.

4. Movement Programming

In order to allow the robotic arm to write, a computer program needs to be written. There are four major tasks to be done by the program.

a) Figure out the character(s) to be written.

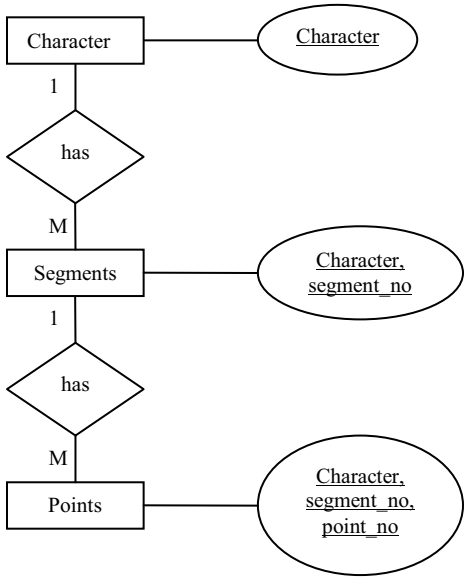

Figure 2. ERD of the database

b) Extract the segment / movement information for that particular character from the database.

c) Generate the exact data / command to be sent to the robotic arm.

d) Send data / command to the robotic arm.

The first task is done by querying from an external input. This can either be an input from user or from an input device. The second task is done by querying the database for all the information related to the particular character to be written. This information can then be stored in an appropriate way and then used to do the third task. In the third task, the information from the database needs to be manipulated in such a way that the actual data needed by the robotic arm can be produced. For example, if the point information in the database represents the displacements from a particular point, then the exact coordinate that is going to be used by the robotic arm needs to be calculated. After the calculation is done, a sequence of commands that is understood by the robot will be generated and then sent to the robot. These commands are robot-specific, and will be different between different robots depending on the types of command that the robot accepts.

5. Implementation of the Algorithm

An implementation of the writing algorithm has been tested using a Mitsubishi RV-2AJ robotic arm [4]. In this implementation, an application was developed using Microsoft Visual Basic and is run on a computer that is directly connected to the robotic arm. Figure 3 shows the overall architecture of the robot writing system.

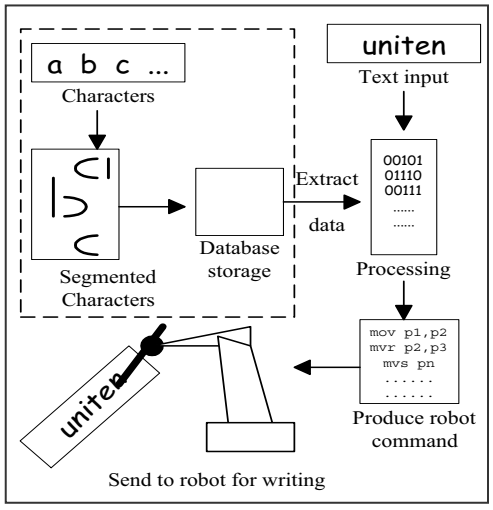

Figure 3. Block diagram of the robot writing system

The application that has been developed can accept any string of characters written by the user, as long as the strings can fit on the drawing board which size has been pre-determined. Using the segment information from the database and a kerning algorithm [1], a preview of the output is then displayed by the application. Figure 4 shows the interface of the robot writing application with a preview of the text to be written. Figure 5 shows a snapshot of the RV-2AJ robotic arm while it is writing down a word.

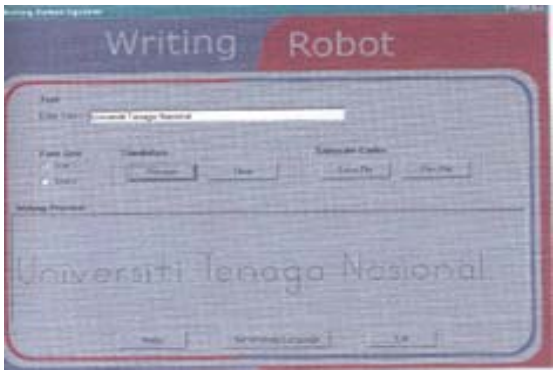

Figure 4. Interface of the robot writing application

Figure 5. Snapshot of the RV-2AJ robot arm while writing

An analysis of the writing produced by the robot shows that it is exactly the same as the one produced by the preview generated by the application. From here, it can be concluded that the segmentation algorithm works and therefore can be used by robotic hands or arms to perform writing. A more detailed description of the implementation of the RV-2AJ robot writing application can be found in [3].

6. Conclusion

In this paper, we have proposed a method of segmenting alphanumeric characters such that it can be written using a robotic hand or arm. We have also proposed a way to store the segmented characters and how to use them. By integrating the three concepts together, it is possible to write a program to enable any robotic arm or robotic hand to write. An application to enable a Mitsubishi RV-2AJ robotic arm to write was developed to test the algorithm. The application had performed successfully, and thus verified the algorithm.

7. References

[1] Blatner, D., "Spacing out with your type; leading and kerning", MacWorld, vol. 14, pp. 158 (2), 1997.

[2] D. McCloy, M.Harris, *Robotics An Introduction*, Open University Press, 1986.

[3] Karina Fernandez, "Writing Robot", Final Year Project Thesis, Universiti Tenaga Nasional, April 2002.

[4] Mitsubishi Electric Shell Sheet, "RV-1A, RV-2AJ Industrial Robots New Concepts for Better Solutions", USA, February 2000.

[5] P. Michelman, P. Allen, "Compliant manipulation with a dextrous robot hand", Proc. IEEE International Conference on Robotics and Automation, vol. 3, pp. 711-716, 1993.

[6] S. Masui, T. Terano, "Calligraphic robot with fuzzy logic", on Proc. The third IEEE Conference on Fuzzy Systems, vol. 3, pp. 1598-1603, 1994.

[7] V. Potkonjak, M. Popovic, M. Lazarevic, J, Sinanovic, "Redundancy problem in writing: from human to anthropomorphic robot arm", IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 28, issue 6, pp. 790-805, Dec. 1998.