Final Report: 3D Printer with 4DOF Dobot Arm

A project of the 2017 Robotics Course of the School of Information Science and Technology (SIST) of ShanghaiTech University
`https://robotics.shanghaitech.edu.cn/teaching/robotics2017`

Yang Dan
University of Shanghaitech
yangdan@shanghaitech.edu.cn

Hu Hang
University of Shanghaitech
huhang@shanghaitech.edu.cn

January 17, 2018

Abstract

In this project, we learned Gcode who plays an important role in 3D printing and we write a slicer for simple geometric figures. For the slicer, we only need the basic information of geometric figures as an input to generate the Gcode for 3D printing. We refit the hardware of the Dobot Arm, which enables us to control the gripper and 3D printing under 3D printing framework at the same time. Therefore, we achieve three different stages of demo in our project, using all the techniques mentioned above.

I. Introduction

There are three stages we implemented in this project.
At the first stage, we make the Dobot Arm automatically print, grab and seal. An ordinary 3D printed product has all its parts connected to each other, so our goal is to print some small objects inside a container with every part separated to each other. To do this, we integrate a gripper into the 3D printing system of Dobot Arm.

As for the second stage, we print a Matryoshka doll and put them together using the gripper. The main difference between this stage and the previous stage is that grabbing a lid onto the other object will need a much more accurate control of the gripper than simply placing one object into the other.

At the third stage, we try to achieve real 3D printing and are able to print objects like pyramids and cubes. Existing 3D printers are actually "2.5D" , since they print objects layer by layer. What we want do is to print objects in the real 3D way.

The major task is to figure out how to control the gripper with 3D Printing framework first. Then do experiments with the material under different conditions to find the best setting for printing so that we can achieve real 3D printing.

The hardware we used is a 4-Degree-of-Freedom (DOF) Dobot Magician arm, with high accuracy and multi function including grabbing, 3D printing and laser printing, etc. However, only one function can be applied at one time due to the limitation of the framework.

In this report, we first present some related papers and softwares in the State-Of-Art part, and then describe our system in the Project Description part. The results will be shown in the System Evaluation part and the conclusion comes at last.

II. State of the Art

i. Member: Hu Hang

i.1 Papers

- **3D Printing and Camera Mapping - Artwork: Digital Buddha**[Luo H L, 2016]

  This paper presented a way to recognize object in real world and read the data of the object using a camera and then send the data to a 3D printing system so that we can make a copy of the object without knowing the detail parameters of the object.
- **Research on Motion Control Algorithm of 4-DOF Multi-Function Robots**[Hao X Q, 2014]

  This paper shows how to build a motion controling algorithm for 4 degree-of-freedom arm, the paper introduce inverse-Kinetic as well as Direct-Kinetic motion controling algorithm. This paper focus much on mathematical realization instead of a real world operation and evaluation.
- **Kinematic Analysis of a 3D Printable 4-DOF Desktop Robot Actuated Exclusively by Revolute Pairs** [Dallalibera F, 2016]

  This paper give us a kinematic analysis of a 3D printable 4 degree-of-freedom desktop robot, where the hardware the paper used, is much the same as our hardware. This paper is aiming at give a kinematic approach which can serves all arms including low accuracy and high accuracy to achieve a better performance in 3D printing.

i.2 Recognition of Off-Line Handwritten Devanagari Characters using Combinational Feature Extraction

[R. Birajdar D, 2015]

This paper is aiming at recognizing off-line handwritten characters using combinational feature extraction. The major task is that the peculiarities involved in the writing styles of different person are very much different. So matching them to the same target is not an easy task.

In this paper, they first generate a binary image for each target, and then extract features from handwriting characters using shadow features of the character, they scaled the image and generate a 64 Dimensional feature for each character which is called the bonding box, then they collect the data in the 64 Dimensional feature to build a histogram of each direction. All these data are to serves the training method of the classifier, which is a standard back propagation algorithm. They achieve a higher classification performance by designing a pattern recognition system. They also applied a combination of individual classifier outputs so that it can overcome deficiencies of features and of single classifier.

The result shows that when using top 5 choices result as a proposed method result, this approach can reach an accuracy up to 99% , which exceed the previous approach presented by others, which can only reach an accuracy of 90.65%.

Therefore the paper achieve a really high accuracy in handwriting character recognition. And we can have a better performance in recognizing demand if apply their method.

i.3 DobotStudio & open source: open-dobot-gui

This project is implemented based on the DobotStudio, which is a software that given by the developer of Dobot Magician, it was easy to operate and serves many needs.

The system of the Dobot Magician mainly consist of two parts: all ports related to SW is used for sending messages to DobotStudio while 3D printing framework is working, and all ports related to GP is used for sending messages to DobotStudio while any operation other than 3D printing. DobotStudio was used to burn the appropriate framework into Dobot Magician arm to implement different tasks.

The structure of DobotStudio is clear, and I am only going to refer to the script part, the script that Dobot offered was written in python, and it seems there is a complete language system in python that serves Dobot Magician, we can use those command to manipulate the arm to achieve several tasks such as rotate, translate, and even grip, so using python to achieve gripping is a promising way.

There is also another Dobot operating software which is: open-dobot-gui, the system was build to control a Dobot robotic arm via an Arduino Mega 2560 and a RAMPS 1.4 board. Since we would like to manipulate the arm to implement grip and 3D printin at the same time, it might be neccesary for us to operate with Arduino, if there is no appropriate port left for the control of air pump.

The gui app is written in python and uses pyqt. It is not yet completely functional, but works enough for testing purposes.

However, the 3D printing part has not yet covered by the open-dobot-gui, so we still have to find the best way to manipulate the Dobot Magician arm. A combination of python code and the 3D printing part of the DobotStudio seems to be a nice choice.

ii. Member: Yang Dan

ii.1 Papers

- **Geometric Figure Recognition by Using Multiple Algorithms Combination** [Wei Xu et al., 2011]

  This thesis is mainly focused on the geometric figure recognition. Based on different kinds of geometric shapes, a number of commonly used figure classification algorithm were designed to recognize all kinds of figures and a geometric figure painted on palette arbitrarily. The experimental results indicate that figure recognition algorithms can be well performed in an integrated graphical user interface
- **The recognition the geometric figure based on the Hough transformation theory** [Yang and Zhou, 2002]

  The recognition of the geometric figure is the basis of the pattern recognition and the computer optic, moreover the recognition of lines, rectangles and circles are the basis of the figure recognition. In this article, basing on the Hough transformation theory and its vote system, working with the geometric character of each figure, the writer achieved the recognition of kinds of figures and detected there geometric parameter accurately
- **Geometric Shape Recognition of Freeform Curves and Surfaces** [Elber and Kim, 1997]

  The researchers formulate intrinsic conditions that are parameterization independent whenever possible. These conditions can detect whether a curve segment is a line or a circle which is exactly what we need for our recognition of drawing on a paper.
- **A Simple Approach to Recognize Geometric Shapes Interactively** [Jorge and Fonseca, 2000]

  In this paper, the researchers described a simple and fast recognizer for elementary geometric shapes. Their intent was to provide a means to support calligraphic interaction.

  It was temporal adjacency and global geometric properties of figures to recognize a simple vocabulary of geometric shapes including solid and dash line styles, selection and delete gestures.

The geometric features used are invariant with rotation and scale of figures. They have found the method very usable with acceptable recognition rates although the multi-stroke approach poses problems n choosing appropriate values for time-outs.

Their experience using this recognizer in interactive setting shows that the observed recognition rates make it usable for drawing input. The time-outs required to recognize shapes regardless of the number of strokes using temporal adjacency have proved to be the most unnatural constraint so far.

The recognizer successfully identified 91% of the sketches drawn. The high recognition rates and fast response characteristic of this recognizer make it very usable in interactive applications.

ii.2 Tools: Cura and G-code

Cura is free Open Source software released on the AGPLv3 license. Cura 3D is the friendly face of slicing software for 3D printers, it takes a 3D model and slices it into layers to create a file known as G-Code, this is the code that a 3D printer understands.

It is currently maintained by Ultimaker who employed David to work on it. Cura is well suited for Bowden-extruder machines like the Ultimaker and most Delta-bots. also you can use it with most of reprap printers. Cura features it's own Slicing engine, GCode sender and other tools to provide an "all in one" solution for 3D printing. However all components can be used on their own.

G-code, which has many variants, is the common name for the most widely used numerical control programming language. It is used mainly in computer-aided manufacturing to control automated machine tools.

G-code is sometimes called G programming language in which people tell computerized machine tools how to make something. The 'how' is defined by G-code instructions provided to a machine controller that tells the motor where to move, how fast to move, and what path to follow. The same concept also extends to non-cutting tools such as forming or burnishing tools, photo-plotting, additive methods such a 3D printing, and measuring instruments.

iii. Additional Paper

iii.1 Research on Modeling and Simulation and Trajectory Planning Algorithms for Dobot Manipulators

[Sun J, 2016]

This paper aims to study D-H modeling , kinematics analyses , trajectory planning algorithms ,and simulation of Dobot manipulators using Mabtlab environment, although the controling system is very much different from our project, however, this is the only paper that talk about Dobot Magician and we believe read a paper about Dobot can help us with manipulating it. Also, we will try to achieve automatic trajectory finding if we still have time.

They first find the homogenous transform matrix for each joint, and then use all these transformation to construct the transformation matrix of the end node with respect to the base jonint. Then they use Matlab to generate different types of trajectory for the arm, and analysis the corresponding data.

Based on all trajectory that found, they concluded some graph with respect to different part of the arm, the graph shows how each joint move during the whole process, and it also shows that it match the expectation.

This paragraph will be helpful when we are implementing our project in the object gripping part. It give a way to implement object griping with inverse-Kinetic.

III. System Description

i. System structure

i.1 Physical Structure

The physical part of the system will be based on a Dobot Magician 3D Printer along with a gripper. There is also some refit hardware that will be described in detail.

Figure 1: *Dobot Arm*

The Dobot Magician arm is a 4 DOF robotic, this arm is built for the use of everyone, which means it is easy to operate. It is a highly integrated arm with the ability to 3D printing, gripping, laser printing and also writing and painting. All tasks can be implemented with a high accuracy, however, there is one limitation that it can only implement one type of task at a time. For example, you can not control the arm to do 3D printing under grabbing framework. And this makes us impossible to achieve our goal without refitting the hardware.

The 3D Printer part has an extruder and a heater with a fan behind. The heater heats the temperature up so that the material(ABS) used for 3D printing can be melted. As long as the temperature reaches 200 Celsius, the extruder will push the material out. One note is that everytime we want to 3D printing something we should test the extrusion condition first, making sure that it reaches stable extrusion.

Figure 2: *The extruder for 3D printing*

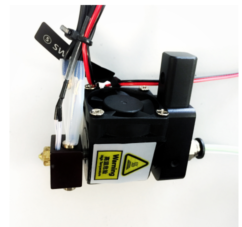

Figure 3: *The heater and fan for 3D printing*

The gripper part consists of a gripper and an air pump. When the air pump turns to the open mode, it blows the air into a tube and the gripper will open. When the air pump turns to the close mode, it sucks the air and the gripper will close.

Figure 4: *The gripper and the air pump*

We build a voltage divider on a breadboard and also refit the hardware.

In the extruder and the gripper part, we use a 3D printed object to connect these two parts, so that we can move the extruder and the gripper at the same time. However, it is different

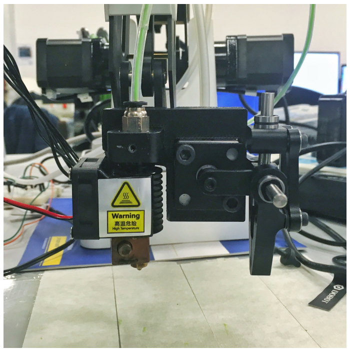

Figure 5: *The refit extruder and gripper*

from using the gripper in a normal way.

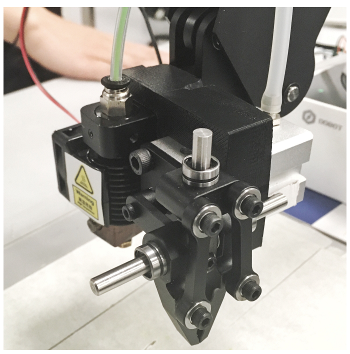

Figure 6: *The refit extruder and gripper*

First of all, the gripper has a different coordinate system with respect to the extruder. We make them lie in a same horizontal line and also have no difference in the x direction, but there is a $53mm$ difference in the y direction. Therefore, we have to notice the difference of the coordinate system, when controlling the gripper.

Also, using the 3D printed object to hold the extruder and the gripper at the same time makes the gripper lose the freedom of rotation. The gripper can only opens in x direction, this also lead to the limitation of the shape of the objects in our demo. Therefore, we choose cubes for demo, which is easy to grab.

The pump is used to control the gripper switching to either an open pose or a close pose. The pump is controlled by two ports. One port delivers a $12V$ voltage and the other gives a $3V$ voltage. When given the $12V$ signal only, the pump blows air and the gripper opens. When given the $12V$ signal and the $3V$ signal at the same time, the pump sucks air and the gripper closes.

However, theses two ports can not work under 3D printing framework and that is why we have to refit the hardware to achieve controlling the pump under 3D printing framework.

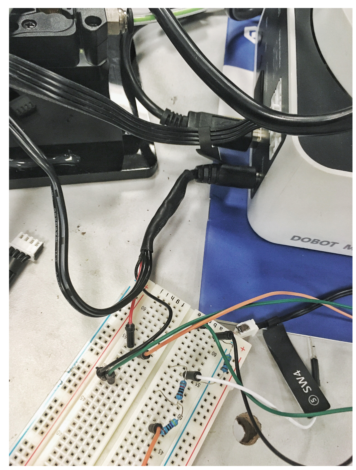

Figure 7: *12V from power supply*

We find out that the fan in the 3D printing system is using a $12V$ signal and we realize that the fan should be turned on during the whole 3D printing process. So, we use the voltage from the power supply of the arm to give a $12V$ power to the fan and also as the $12V$ signal that make the pump blow.

Here, we build a voltage divider on the breadboard, so that we can divide the $12V$ signal from the port, which originally used to control the fan, into $3V$. Then we can close the gripper using Gcode $M106$ and open the gripper using Gcode $M107$.

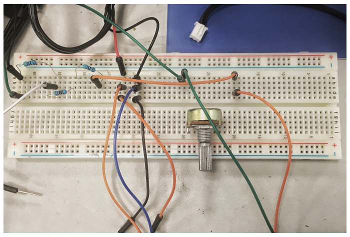

Figure 8: *The voltage divider*

i.2 Software Structure

In the software part, we have DobotStudio as a controlling system for Dobot arm, all communications are done using DobotStudio.To use the 3D printing function, we burn the 3D printing framework into the Dobot arm via DobotStudio, and then we can control the arm using Gcode.

The Gcode that used to implement the three stages is generated using a python program.

```
def testingstage(startpoint,width,height):
def container(startpoint,width,height,H):
def container_R(startpoint,width,height,H):
def rectangular(startpoint,width,height,H):
def cover(startpoint,width,height,H):
def cover_R(startpoint,width,height,H):
def endprint(z):
def solid_rectangular(startpoint,width,height,H):
def Xfunction(startpoint,width,height):
def StraightUp(startpoint, H):
def Edge(startpoint,width,height,H):
```

Figure 9: *Python Program for slicing*

This program mainly consists of two parts. The first part is used as a slicer and the second part is used to control the gripper.

As for the slicer, different functions are used to construct different types of objects. For example, the *container* function is used to construct a container in the first stage, and the *container_R* function is used to construct a container in the second stage. There are some differences between them. In the first stage, the container will be sealed, while in the second, the container will not be sealed but it will have a cap. Therefore the container have to be made in a way that the cap can be placed safely and stably.

```
def FindTra(destination,objects):
def OpenGripper():
def CloseGripper():
def FindThePlace(x,y,z):
def MoveAside(x,y,z):
def GoDown(x,y,z):
def GoDown2(x,y,z):
def GoUp(x,y,z):
def GoDestLine(x,y,z):
def GoDestDown(x,y,z):
```

Figure 10: *Python Program for grabbing*

The grabbing part uses a function $FindTra(destination, list\ of\ objects)$ to achieve grabbing objects in the list to destination.

The *FindTra* function call the other functions to find a trajectory for grabbing based on the information of the objects with known location.

ii. Problem overcome

ii.1 How can we control the gripper while using the 3D printing framework?

To achieve this, we first try to remove the fan, and use its port to control the air pump. Since we believe that the fan is only used to cool down the extruder and we have a temperature sensor to ensure the temperature, removing the fan seems to have no effect on printing. However, as soon as we remove the fan, the material becomes sticking in the middle of the supply pipe, which takes us three days to realize that this problem results from the absence of the fan. Without a fan, the material can be easily melted in the middle of the pipe. To solve this problem, we connect the power supply directly to the fan and make it work again.

IV. System Evaluation

i. Demo 1

In the first demo, we present the basic functions we implemented with the modified Dobot arm, which are 3D printing and grabbing.

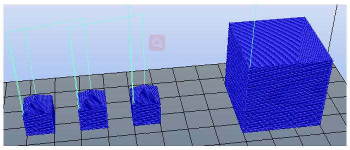

Figure 11: *demo_1*

First, we 3D print a container with some small objects. Then, we grab these objects into the container. Finally, we print a lid directly on the top of the container so that these small objects are locked inside.

There is no manual control during the whole process and the quadrate shape of the objects makes it easy to grab.

ii. Demo 2

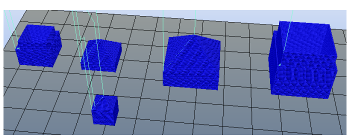

Figure 12: *demo_2*

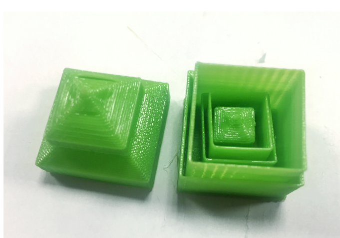

Figure 13: *demo_2*

At stage 2, it is much harder to cover the container with a lid. Since we combine the gripper with the 3D printer, we lose one degree of freedom of the gripper that we can not rotate the gripper to the best position to grab the the lid. Therefore, we have to arrange all the lids

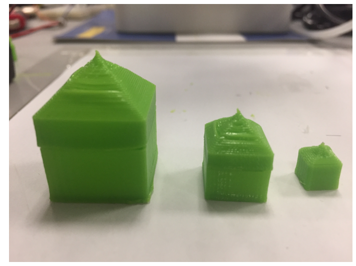

Figure 14: *demo_2*

and containers at the same y-axis as where the gripper is.

In addition, to cover the container with a lid successfully, we carefully designed the objects as shown in figure 12.

iii. Demo 3

In order to achieve real 3D printing, we need to know how the materials becomes under different conditions very well. We did experiments on the materials with different feedrates, different speed of the printer head's motions, different temperatures and different ratios of extrusion. Some results are showed below.

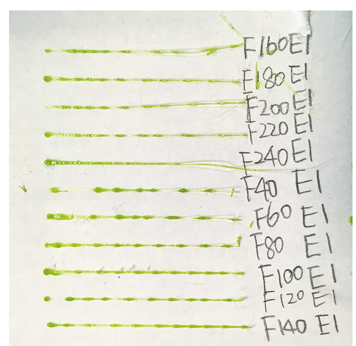

Figure 15: *Feedrate*

After these tests, we find out that we can get the best products with F10 and the ratio of extrusion is 0.3.

We designed two models for real 3D printing. One is the pyramid, and the other is a cube (Figure 18, 19).

Here are some results of real 3D printing.

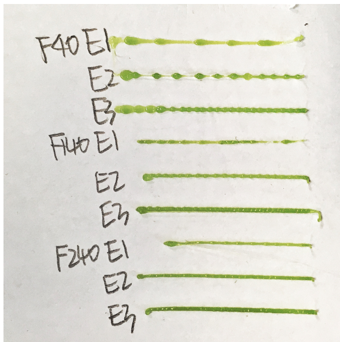

Figure 16: *Extrusion & Feedrate*

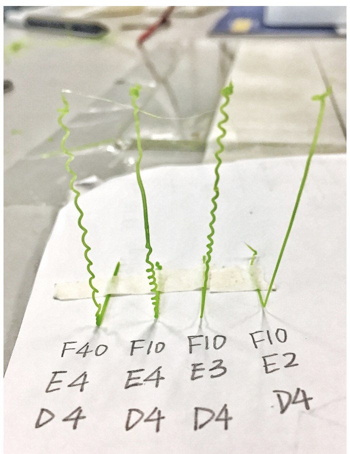

Figure 17: *Extrusion*

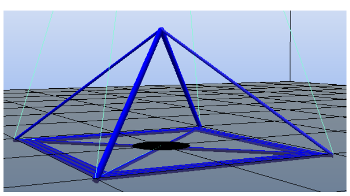

Figure 18: *demo_3*

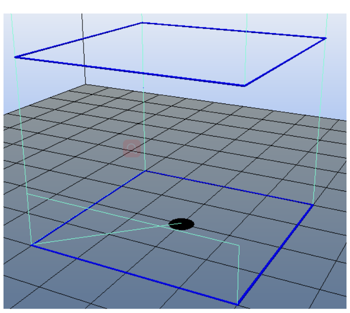

Figure 19: *demo_3*

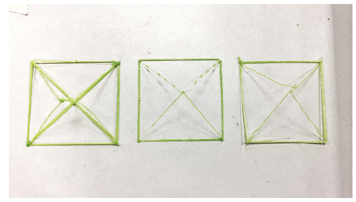

Figure 20: *demo_3*

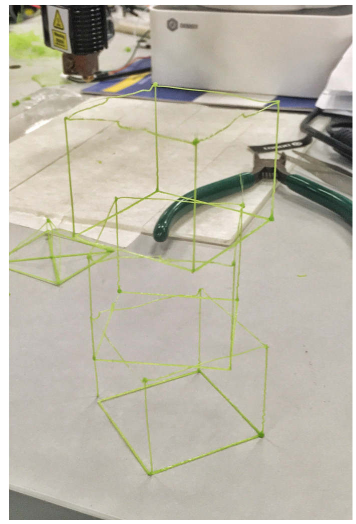

Figure 21: *demo_3*

V. Conclusion

Our idea of the first two stages is to build a multi-functional system that combines 3D Printing and grabbing. With the help of Dobot Arm and some open sources, we are able to implement a system that first build a 3D model according to given information and then turn on 3D printing mode to print out the objects and finally use the gripper to grab the target object and place it into the container.

In the third stage, we achieve real 3D printing of cubes and pyramids after a lot of tests, which shows the possibility of real 3D printing.

We implement what we expect to do in the first place, which is printing and grabbing at the same time. We also did some experiments in real 3D printing. Therefore, we believe that we accomplished our project successfully.

As for further work, we can try to do real 3D printing for all kinds of objects, and also make the products more stable.

References

Yoshikawa Y et al. Dallalibera F, Penaloza C I. Kinematic analysis of a 3d printable 4-dof desktop robot actuated exclusively by revolute pairs. *Intelligent Autonomous Systems 13. Springer International Publishing*, pages 1627–1638, 2016.

Gershon Elber and Myung Soo Kim. Geometric shape recognition of freeform curves and surfaces. *Graphical Models & Image Processing*, 59(6):417–433, 1997.

Deng G W. Hao X Q, Chen J J. Research on motion control algorithm of 4-dof multifunction robots. *Telecom Power Technology*, 2014.

Joaquim A. Jorge and Manuel J. Fonseca. *A Simple Approach to Recognise Geometric Shapes Interactively*. 2000.

Hung Y P. Luo H L, Chen I C. 3d printing and camera mapping - artwork: Digital buddha. *CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 3867–3870, 2016.

M. Patil M. R. Birajdar D. Recognition of offline handwritten devanagari characters using combinational feature extraction. *International Journal of Computer Applications*, pages 120(4):1–4, 2015.

Ma L. Sun J, Zhang J L. Research on modeling and simulation and trajectory planning algorithms for dobot manipulators. *Machinery and Electronic*, pages 72–75, 2016.

Jia Wei Xu, Seop Hyeong Park, and Xian Yun Fei. Geometric figure recognition by using multiple algorithms combination. 474-476:417–421, 04 2011.

Zhi Yang and Qi Zhou. The recognition the geometric figure based on the hough transformation theory. *Journal of Chongqing Polytechnic College*, 2002.