CMSE 501 / DASC 501 - Introduction to Computational Sciences

Raspberry Pi STL format Recognition

Term Project Report

Rasp Snakes (Group 12):

Shams Torabnia – 0070136

Hammad Rahman – 0071057

Shukhrat Khuseynov – 0070495

GitHub: RaspiSnake repository (for now private)

Date: 20.12.2019

Table of Contents

Definitions (Template sections 1-2)

3D topology

3D topology is a way of representing three-dimensional shapes by modeling their structure geometrically with nodes and vectors in terms of simpler figures. Quadrilaterals (four sided polygons) and triangles (three sided polygons) are most commonly used to represent the 3D models.

CAD software

Computer-Aided Design (CAD) software is a type of program used to create and work on both two-dimensional and three-dimensional shapes by designers and engineers to represent the physical objects and mechanics related to their field. 3D models of CAD are highly employed in manufacturing and automotive industries.

File types

There are many different file types used for 3D modeling with their own strengths and weaknesses, such as 3MF, AMF, IGS, and STP. But the most widely used file type is STL (Standard Tessellation Language), which is usually viewed and modeled in a CAD software and can be deployed for 3D printing. An STL file constitutes a 3D shape represented geometrically in a mesh model with a topology of triangulated surface.

Robotic 3D printing

Robotic arms are more widely used nowadays for printing the 3D objects. They are programmable and their main part is actuator which moves accordingly to print a model, which was given as an input.

Raspberry Pi

Raspberry Pi is a Linux based small computer ARM V8 CPU with different configurations. The official OS of this hardware is Raspbian, a version of Debian, which has basic functions of the main Debian with limited abilities. Small computers such as Raspberry Pi gradually find vast usage in different industries. The ability to use a small computer embedded in an industrial setup, for instance a robot, makes it a favorite tool for designers.

Business Understanding (Template sections 1-2)

The aim of this project is the development of a program, namely RS Design, which recognizes STL file types in Raspberry Pi for the purpose of robotic 3D printing.

Although Raspberry 4-4GB hardware with Raspbian OS is a convenient and logical alternative for the standard computers to use within the robotic actuator, it has memory limit, which makes the usage of complex programs on it for 3D processing and viewing almost impossible. Our application, which is a simple version of CAD software, could be used instead by taking less space in the memory thanks to Python. The output will pave the way towards exhibiting 3D graphics of an STL file on Raspberry Pi, which is motivating because of the lack of 3D libraries in Raspbian OS.

Programming (Template sections 3-8)

Data

Since this is more data generating project, the only data used in the application is its input, which is a graphical file in STL format to be processed and shown on the screen.

Python structures and modules

Many different structures of Python, such as functions, integer/float and string variables, global variables, lists, numpy arrays, were used in the code.

Regarding the modules, the main ones were numpy-stl module for processing the meshes of STL files, tkinter module for the implementation of graphical user interface (GUI) of the application, and matplotlib module (particularly its pyplot, mplot3d, and 3D navigation functions) for the visualization purposes. The numpy and math modules were used for calculating the geometrical parameters of 3D processing functions. The os and sys modules were used initially to deal with saving and updating problems of the interface.

Main platform

Using the built-in tkinter module of Python, the GUI of the main platform was implemented. When it is launched, the RS Design program starts with a splash screen, whose background picture is obtained from a GIF image, using Tk window and canvas functions of tkinter. After launching the main window, File toolbar with buttons appear on the left, the canvas of 3D plot is in the middle, and Create toolbar with figure creation buttons appear on the right. Below the canvas there is also a frame with navigation tools and under it there is a button showing the STL parameters of the plotted shape. All of the buttons, icons, and other GUI elements were added by using basic tkinter functions. Moreover, the button opening the STL file uses a standard way for that, which was embedded using the filedialog function. Among the File toolbar buttons, the save button asks a name for STL file to be saved. If there are any unaccepted characters in the name, they are deleted from the name's string to be saved correctly. Lastly, when the figure creation buttons are pressed, a pop-up window, asking from the user the parameters for the figure to be plotted, appears. All the elements of the interface were given the values of locations and sizes as an input to be placed correctly and harmoniously.

STL processing

For the main functionality of STL processing there is a "RaspS.stl" file to be worked on throughout the program. It is aimed to save the progress and be updated, whenever it is asked by the user. Also "Empty.stl" file is used to make the plotting canvas new. Moreover, three fundamental functions for the STL recognition were added to the code initially: find_mins_maxs(), translate(), and copy_obj() functions. In a nutshell, they measure the correct maximum dimensions for the proper plotting, translate the meshes to the relevant axes, and copy a 3D topology from a file to save its meshes for the further usage.

As it was mentioned above, there are file buttons on the left end of the main platform, which contain several essential functions for STL processing. If the button for creating a new file is pressed, then the corresponding meshes are calculated using numpy's zeros function, converted to ASCII and saved directly as "RaspS.stl". When the button for opening an STL file (with some 3D shape) is pressed, using the fundamental functions described above, the opened file is combined as ASCII with the existent plot of "RaspS.stl" by concatenating their meshes. Similarly, the save button, which was discussed before, and exit button do the function as their name implies.

Additionally, there are also figure creation buttons on the right end of the main platform, intended to plot few geometric figures to the existent plot, depending on the button, such as 3D box, sphere and cone. The figures are plotted according to the user's entry. The meshes are calculated using the numpy arrays. Their parameters were designed by calculating the geometric properties and the vectors of each figure. The similar algorithm of concatenating and saving to “RaspS.stl” is used.

Output

As a result, a simple 3D processing application, RS Design, can now be used to recognize and view STL files on Raspberry Pi to print them with the robotic actuator. There might be small bugs remaining, which can be easily eliminated.

Deployment (Template section 9)

Currently, there are lots of mechatronics projects, where Raspberry Pi is used as their controller. It would be great, if there is a way to have a graphical simulation of the path generated by Raspberry. RS Design software will be deployed primarily in such projects in Koç University’s labs.

Feedback (Template section 10)

Further improvements

If we had more time, we would have added also other formats of 3D modeling for recognition and processing. Moreover, more figures could be added for plotting to make this feature more diverse. Since this project is related to the research areas of some group members, probably it is not only limited to the mentioned developments, but also adding mesh toolbar and path toolbar with the general program improvement is planned. We also plan to make our GitHub public later on.

Group work

Shams implemented the ideas and the main algorithm of the project. He also did the proposal and presentation. Hammad was helping Shams with his work, even after dropping the course. Shukhrat mainly worked on GUI and a different approach of plotting 3D figures, which we decided not to include in the final version. He also prepared the report.

While implementing the core code, we were working together on GitHub, which was very convenient and interesting. It is also useful for sharing our project with others and for further developments in terms of external advices.

References

[1] StereoLithography Interface Specification, 3D Systems, Inc., July 1988

[2] StereoLithography Interface Specification, 3D Systems, Inc., October 1989

[3] SLC File Specification, 3D Systems, Inc., 1994

[4] "STL File Format for 3D Printing - Explained in Simple Terms". All3DP. 17 November 2016. Retrieved 5 May 2017

[5] Hiller, Jonathan D.; Lipson, Hod (2009). "STL 2.0: A Proposal for a Universal Multi-Material Additive Manufacturing File Format" (PDF). Cornell University. Retrieved 5 May 2017.

[6] Millman, K. Jarrod; Aivazis, Michael (2011). "Python for Scientists and Engineers". Computing in Science and Engineering. 13 (2): 9–12.

[7] van der Walt, Stéfan; Colbert, S. Chris; Varoquaux, Gaël (2011). "The NumPy array: a structure for efficient numerical computation". Computing in Science and Engineering. IEEE. arXiv:1102.1523

[8] https://docs.scipy.org/doc/numpy/reference/

[9] https://buildmedia.readthedocs.org/media/pdf/numpy-stl/latest/numpy-stl.pdf

[10] https://docs.python.org/2/library/tkinter.html