

Pepr3D

Authors: Bc. Štěpán Hojdar, Bc. Tomáš Iser, Bc.
Jindřich Pikora, Bc. Luis Sanchez

Supervisor: Mgr. Oskár Elek, PhD.

Consultants: doc. Ing. Jaroslav Křivánek, Ph.D., Ing.
Vojtěch Bubník (Prusa Research s.r.o.)

Faculty of Mathematics and Physics
Charles University

Contents

1	Introduction	2
1.1	3D printing basics	2
1.2	Prusa environment	2
1.3	Multimaterial printing	3
1.4	Our project	3
1.5	Related works	3
1.6	Challenges in this project	3
1.6.1	Handling the geometry during editing	4
1.6.2	Exporting the finished objects	4
1.6.3	Performance	4
2	Use case	5
3	Execution of the project	6
3.1	Todo by Jindra	6

Chapter 1

Introduction

1.1 3D printing basics

3D printing is a new technology that has seen rapid development in the last years. It comes in many different forms, melting plastic, fusing metals, shining UV on photopolymers, etc. Fused Deposition Modelling (FDM) is the most popular and accessible to the general public and for the purpose of this project, when we talk about 3D printing, we will always mean FDM printers, unless stated otherwise.

FDM printing is a relatively simple process - a printer head melts the plastic filament and deposits it on a preheated platform layer by layer, from the bottom towards the top. The printer has to regulate the temperature of both the filament in the head and the moving platform for the deposited material to bond correctly. Several types of filaments are used, namely PLA, ABS, PET and others.

1.2 Prusa environment

The Prusa environment is very similar to the general description we provided in the section 1.1. For the purpose of our project, the most important concept in the Prusa environment is the slicer. The slicer is a program that receives the 3d model the user wishes to print out and creates the instructions for the Prusa 3d Printer - a G-code file. The file is then transferred to the printer, which then executes the commands in the G-code file. The slicer has to plan the movement of the head for the whole print. This includes several crucial things:

- * Covering the whole area of each layer
- * Reinforcing the walls of the object to make them sturdier
- * Filling the inside of the object with a rougher print, because it won't be visible when finished
- * Planning the path so the head can stay in one Z level - an "Eulerian path".
- * Switching the materials for multimaterial printing (more in 1.3)

Prusa develop their own slicer - a forked branch of an open-source program called Slic3r (<http://slic3r.org/>), called Slic3r Prusa Edition (<https://www.prusa3d.com/slic3r-prusa-edition/>). This slicer can do all we listed above very well.

1.3 Multimaterial printing

Multimaterial printing is a very new concept, even in the fairly new world of 3D printing. Many of the simpler and cheaper 3D printers can only print one material models - one color for the whole object. However, many users would like to print models that include more than one color. Even though the more advanced printers are capable of combining up to four different materials into one print, the process to achieve this is rather cumbersome for the end user - the user has to manually split the 3D mesh of the object into parts that he wishes to have a different color.

For example, if we are printing a dragon, want the dragon to be black and have white teeth, we have to take the dragon model, and split off each individual tooth. Then tell the slicer that the remaining file - the toothless dragon should be black and the teeth should be white.

This model splitting has to be done in a full 3D editing software like Blender or 3ds Max, which is difficult to control for newcomers and overly complex.

1.4 Our project

Our project aims to make printing a multi-colored object a lot easier, by developing an application that will allow the user to simply paint on the 3D model (i.e. the dragon) with different colors (i.e. color the teeth white), then simply click export and generate the files of the split-off models automatically.

Our application should allow for free hand painting as well as some forms of guided painting - bucket fill and some smarter tools, for example a bucket fill that studies the object's geometry and stops the filling if it detects a sharp edge (i.e. the transition of the tooth into the dragon).

Our aim is to make the application for desktop PCs, with main development time being focused on the Windows operating system. However, we are trying to use software engineering tools that can also be ported to a plethora of other platforms like Linux based OS, Mac OS and mobile, if the need should arise.

1.5 Related works

Based on the analysis of the experts from Prusa Research s.r.o, there, at the moment, does not exist a software that does what this project is trying to achieve.

The closest existing software is Autodesk Meshmixer (<http://www.meshmixer.com/>), which is very complicated and is not targeted for FDM printing specifically. As such, it includes a lot of features that are not important for the FDM users and end up being confusing.

Microsoft 3D Builder (<https://www.microsoft.com/en-us/p/3d-builder/9wzdncrfj3t6?active>

1.6 Challenges in this project

This section should briefly familiarize the reader with some of the parts of the application we think will be difficult to implement correctly, before we present the full program specification.

1.6.1 Handling the geometry during editing

We want our application to be able to emboss text on the surface of the object, detect edges and stop painting the color during bucket fills, allow the user to paint fine details on a rough triangle mesh. All of these things require some degree of subdividing the triangle mesh to allow the user to create small details. We think that this potentially could involve some difficult problems - we have to allow the user to subdivide the triangle mesh enough to actually allow him to create fine details on the surface. However, the if the user goes overboard with the subdivision, the model will be too complex to print or even handle inside a desktop PC.

1.6.2 Exporting the finished objects

After the user is done painting, the application will have to separate the designated objects and areas into distinct meshes. This is potentially a very complicated task to do correctly for non-convex meshes. For example: if we are writing a text on a ball, we really only want the text to be carved deep enough into the ball for the printed material to hold firmly, we do not want it to be too deep. However, if we want the dragon's teeth to be white, we would prefer the whole tooth to be white, not just its surface. The distinction between these two cases could be non-trivial.

1.6.3 Performance

Handling complex geometry is a very taxing task for the user's computer. This application is targeted on beginner-level customers of simple and non-expensive 3D printers. Therefore the application cannot be too hardware demanding - it has to run smoothly on an average 3 year old PC or Notebook. We expect it is going to be hard to ensure this is the case.

Chapter 2

Use case

Chapter 3

Execution of the project

3.1 Todo by Jindra