kinetic vision

MakeUC - 2024 Challenge

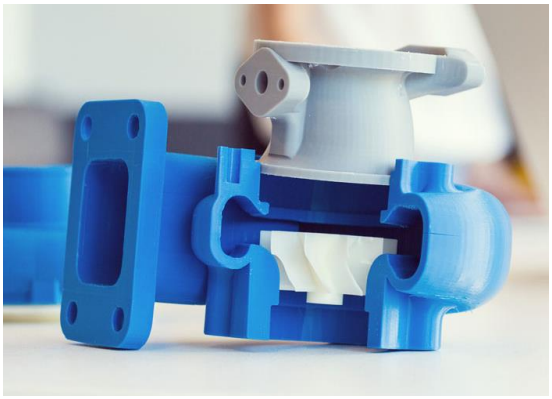
3D Printing, 3D Modeling

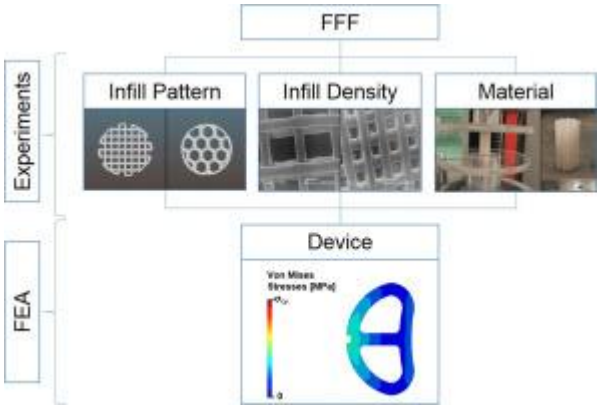
Kinetic Vision utilizes additive manufacturing, or 3D printing, in several aspects of our Engineering Department. Through rapid prototyping, modeling and simulation, and manufacturing design, 3D printing serves to accelerate and optimize our path to a final product. We have a variety of platforms in house to choose from for additive manufacturing, including:

- Fused Deposition Model (FDM) or Fused Filament Fabrication (FFF)
    - the most common hobby 3D printing with plastic filament
- Stereolithography (SLA), or resin printing
    - Also available for hobbyists, where an ultraviolet light cures a liquid photopolymer into a solid resin piece
- Selective Laser Sintering (SLS)
    - A growing powder bed is melted with a laser to solidify a model

The most common, and what our challenge deals with today, is **FDM printing**.

When using an FDM 3D printer, a designer makes a 3D model in a modeling software like Blender, AutoDesk Inventor, or OnShape. Once saved to a format like STL, it can be **sliced**. Slicing is the process of turning a 3D model into a set of instructions for the 3D printer to follow (e.g. go to center of pad, set nozzle to 205°C, extrude at 1mm/s, move 10cm up, then 5 cm right, etc.). These printer instructions are often in a format called **G-code**. G-code is very common in the machining world, and is used on a whole host CNC and 3D printing machines. Your challenge today is to develop a reverse-slicer.





The Challenge

Write an application or simulation that accepts G-Code and allows the user to view the expected 3D model. This should include all the details that the 3D printer will actually spit out, like infill and supports. This application can take any form that you can think of. For example, a command-line tool that accepts a file path and outputs an STL file, a game-engine-based 3D-printing simulation, or anything else that has input and output!

After creating your G-code reverse-slicer, make it your own! Add other features that could solve real-world problems! For example:

- Add the ability to reverse-slice multiple G-code inputs at the same time
- Parameter adjustment inside of the application
- Make a cross-section view to inspect the infill pattern

Getting Started

To get started, we recommend having mechanisms for *input*, *parsing*, *processing*, and *output*. However, the task is open-ended by design and there are **no hard requirements**! If you are looking for somewhere to start, please take inspiration from the following strategies:

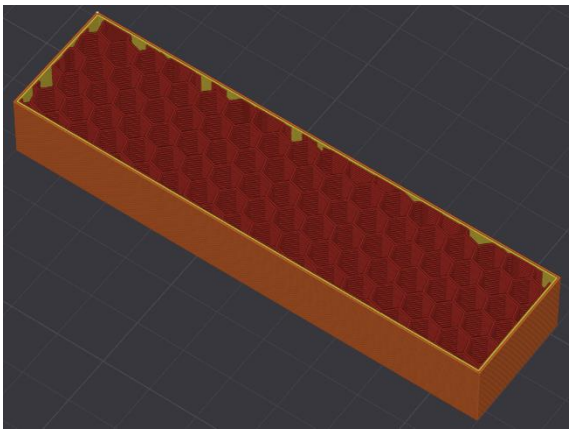
1. Input
    - File path to a local .gcode file
    - Reading stdin
    - A text field to copy+paste raw G-Code into
    - A file upload field on a website
2. Parsing
    - Use an existing G-Code parsing library (i.e., gcode-parser (JavaScript), gcodeparser (Python), g-code (Rust), or alternative)
    - For an extra challenge, write your own:
        - String searching
        - Parser combinator (i.e., parsy (Python), nom (Rust), or alternative)
        - Hand-written Lexer and Parser (example Python script)
        - Parser Generation (i.e, ANTLR or alternative)
3. Processing
    - Generate a 3D mesh according to the printer's instructions
    - Simulate plastic flow and deformation from nozzle extrusion
4. Output
    - File export (e.g., STL, OBJ, or GLTF files)
    - 3D Render (picture or animated)
    - 3D Model Viewer

Resources

You can find some example 3D models and their associated g-code in the following folder:

https://kineticvision-my.sharepoint.com/:f:/g/personal/sfasone_kinetic-vision_com/Ekcn-I9egyJIgPKEuRzptuwBYcdOkcIB5yWJqaTG66tXEQ?e=5d23WK

If you have any difficulty accessing these resources, let us know.



Judging Criteria

- **Functionality**
  - Does the tool successfully reverse-engineer G-code into a 3D model?
- **Accuracy**
  - How accurately does the reconstructed model reflect the original design?
- **Uniqueness**
  - Are there unique features or creative approaches that enhance the functionality?
- **User Experience**
  - Is the interface intuitive and easy to navigate?
- **Developer Experience**
  - Is the provided code and documentation clear?

Prizes

The winning team (up to 4 people) will be able to choose between two prizes:

- A **Bambu Lab A1 mini** 3D Printer to share
- A **$50 Microcenter gift card** for each member on the team