3D Casting Aluminum and the FOSSCar Project

Julia Longtin, Lead Developer, FOSSCar Project
fosscar.faikvm.com | https://gitorious.org/~juri
JuliaL@gmail.com

Julia Longtin: About Me

- Independent hacker/software/hardware engineer working out of College Park, MD, USA
- Acquired MCSE at age 15 & burned it by 18 to work on free software/hardware
- Previously developed/maintained/contributed to:
  - the Linux kernel
  - FAI, TinTin++, Yacms
  - Svgalib, Mach64, Voodoo3, SNES joystick
  - OpenEMR
- Currently maintainer of:
  - GnuGIFT (image recognition system) (we can use lots of help too!)
  - ImplicitCAD (Haskell-based, free CAD software)
  - LinuxPMI
  - CreateVM
- Building 3D printers from scratch for years
- Current focus: FOSSCar / Aluminum printing
- Long-term goal: applying 3D printing to bio-hacking

FOSSCar Project and 3D Printing Aluminum: Roadmap and Goals

- Implement a cast-aluminum based methodology for creating 3D printing-based parts with ImplicitCAD as the principal software design tool
- Use these processes toward the design of 3D-printed cars and car parts
- Develop fully electric cars based on Security-by-Design principles such as metadata-minimizing technologies and encryption
- Implement free software and free hardware ethics and legal standards

Current LostPLA techniques: basics

- Build forge large enough for whole molds
- Cook out molds with forge
- Melt aluminum in forge
- Pour aluminum into molds

[picture]

Current LostPLA techniques: shortcomings

- Large forge needed for removing plastic — inefficient for reuse in melting aluminum!
- Excessive energy consumption — intentionally using heat insulator as mold material!
- Propane is required — and is potentially very dangerous!

Why Microwaves?

- Spent 1.5 years on hot-end for direct layering of aluminum — epic fail!
- What else is out there? → LostPLA
- Microwaves are an attractive replacement for propane-based methods
- Started with microwave forge
- Developed microwave mold cooking process
- Mold process was more successful more quickly
  - Cooking just the plastic, not the mold, is much more efficient!
- For now, using small propane forge with microwave mold process
- Eventually, improve microwave forges and build multi-emitter microwaves to use them with
- Someday, automated operation can make process even safer

3D Casting Aluminum: Microwave Forge

Proof-of-concept microwave forge.

Microwave Forge: Materials

Safety Equipment

- Face masks
- Gloves
- Eye protection

Tools

- Potato masher
- Measuring cups
- Large, well-vented microwave with emitter at top
- Trowel
- Large steel bowl
- Sheet of tempered glass
- paintbrush

Consumables

- 6+ containers, high-heat furnace cement
- 1 bag of perlite
- about 0.5 kg silicon carbide grinding powder (1200 grit)
- water
- cardboard
- newspaper
- painter's tape
- large static bags

Microwave Forge: Cement-perlite mixture

Mixing

- Mix fire cement, water and perlite in large steel bowl
- Ratio of ingredients depends on water content of fire cement:
  - Start with fire cement
  - Add water until easy to mix with masher
  - Add perlite until no longer easy to mix
- Final mixture looks like oatmeal and mashes like peanut butter
- This material serves as structural support and heat insulator for forge

Baking

In a conventional oven,

- Two hours at 100 °C → Two hours at 200 °C → One hour at maximum heat

Microwave Forge: Inner layer

Chamber mold

- A five-sided box with cardboard and painter's tape:
  - Open top / base just large enough to fit crucible (the steel cup)
  - Sides about 3 cm taller than crucible
- Cover with cut-up static bag followed by a layer of newspaper.
- Seal tightly with painter's tape

Initial solid layer

- Cover outside of chamber mold with 30 mm cement-perlite mixture
- Coat sides up to $1 - 2$ cm below height of cup, plus bottom
- Bake normally, except for tearing away mold after first stage in oven

Microwave Forge: Outer layers

- Measure inside of microwave
- Apply and bake 30 mm layers
- Forge, including lid (next slide), must fit inside microwave!

Microwave Forge: Seal and lid

- For lid to match chamber, both lid and tops of chamber walls must be flat!
- Add a layer of cement-perlite mixture to tops of walls
- Place newspaper on tempered glass, the chamber upside-down on that, and bake
- Place a mound on newspaper+tempered glass, more newspaper atop that, and the upside-down chamber atop that
- Press together to force the lid to mate to the chamber, and bake again

Microwave Forge: Susceptor application

[picture of water, sugar, susceptor powder]
[caption]

- Create a susceptor paste:
  - 5 parts powdered sugar
  - 3 parts susceptor powder
  - 2 parts water
- Paint thick layer onto inner layers of lid and floor and walls of chamber
- Bake at 100 °C for one hour
- Add 10–20 more layers until walls are smooth
- Each layer, bake at 100 °C until caramel sugar odor dissipates

Microwave Forge: Proof-of-concept melt

- Put aluminum scraps in coffee cup crucible
- Cook on high for 2:40 (1500 watt microwave)
- As always, remove with fire gloves!

Molten aluminum in forge

Cooled aluminum

Propane Forge Construction

- Microwave forge: good proof of concept but requires more development
- Propane forge: expedient for testing aluminum casts until microwave forge
- Built using standard techniques:
  - Fit open-topped cardboard box to a milk crate
  - Use Red Bull can and coffee can as molds for flame entry and crucible chamber
  - Create forge by pouring layers of RapidSet® cement

3D Casting Aluminum: Propane Forge

Propane forge used for testing the mold pouring process

3D Casting Aluminum: Microwave Molds

[picture]
[caption]

Microwave Molds

Preparing Molds for Pour

- Create sprue extenders by cutting top and bottom of Red Bull cans
- Use duct tape to seal extenders and keep sand out
- After brief heating in oven, place molds in box and fill with sand
- Once molds submerged in sand, place box near forge

Using the Propane Forge for Melting Aluminum

- Use lighter to light propane torch, and propane torch to light larger burner
- Propane tank about two meters from burner for safety
- Tune flame with propane regulator until sound is steady
- Slowly add solid aluminum to crucible until entirely melted ($\sim 1\,\text{hr}$ or less)
  - Should heat to $\sim 800 - 1000\,^{\circ}\text{C}$ with red-hot crucible emanating flames
  - Temperature gun may not go high enough but still a useful guide
  - Stir continuously with long steel rod

Pouring Liquid Aluminum into Molds

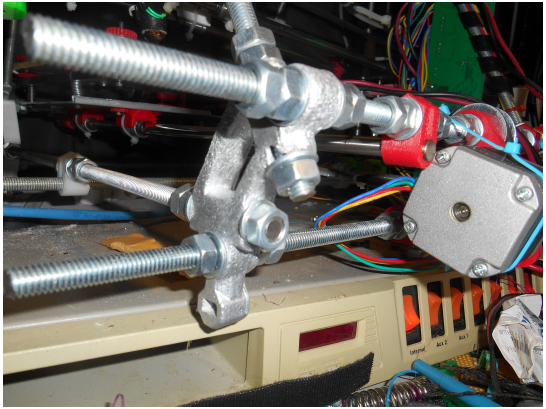
- Lift crucible with tongs and pour until sprue extenders are full
- Wait for flames to die down
- Allow to cool and harden for $8 - 12$ hrs

Separating Cast Parts from Molds

- Break apart mold material with hand tool
- Clear mold material out of object's crevices
- Use smallhacksaw to detach metal sprues
- File down imperfections

Sample Cast Parts: Elks Vertex with Foot

- Structural component of 3D printer
- First 3D printer to upgrade itself to cast aluminum



Sample Cast Parts: Extruder Body

Two versions of almost viable extruder bodies, with some lessons learned:

- Cook till burning plastic smell dissipates; medium power avoids overcooking
- Need to allow air out — planning to experiment with straws
- Must optimize gate dimensions for removal of parts with hacksaw

Sample Cast Parts: Suspension System Component

Fail part!

- Mold was packed in vermiculite $\rightarrow$ leakage at sprues!
- Flat parts may require more clever gate arrangement

ImplicitCAD: Why?

ImplicitCAD is free software that allows for the design of 3D objects for printing.

- Written in Haskell (functional programming language with a concise syntax good for general purpose computation)
- Functions parametrically
- Sets object-specific parameters including diameter and other geometric attributes. ("teeth"/offsets for interfaces with other components/complex object features using shells)
- Generates reusable functions easily — good for concisely encoding dimensions of parts
- Can use simple code function to rotate/specify/distribute across the base of an object
- For ex: a gear with five equally distributed teeth in relation to obj. center

ImplicitCAD: Why?

What about AutoCAD?

- Closed-source, expensive ($4195), no Linux UI
- Not easily customized using reusable code
- Sections have autolisp extensibility component but since parts of system are written in proprietary language and not distributed, flexibility is limited

FOSSCar project uses only free software and hardware.

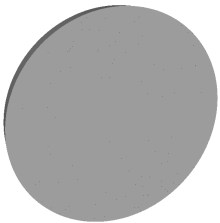
- We needed to design and compile objects from scratch $\rightarrow$ ImplicitCAD
- However, complex, buggy, and ill-maintained (at outset)
- Progressive debugging throughout the project as a result ("chief necromancer" to "maintainer" in 3 months)

Using ImplicitCAD to Compile Designs

- Derivative of OpenSCAD's language (graphical implementation — : $-\backslash$)
- ESCAD (extended SCAD) is the language interpreted/compiled by ImplicitCAD
- ImplicitCAD compiles objects to STL format via Raytracer
- Raytracer allows definition of an object as a series of equations and then determines the boundaries of that object as a series of triangles
- (Slic3r translates STL specificities into G-code for printing)
- Wrote a personalized test suite for ImplicitCAD
- We deployed MeshLab for Viewing Compiled Objects

ImplicitCAD Example: Disc

Render



Implementation 1

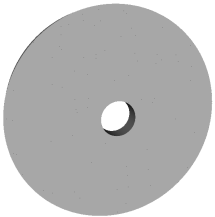
```
module cylinder_3d(height, diameter)
{
  linear_extrude(height)
    {
      circle(r=diameter/2);
    }
}
cylinder_3d(height=10, diameter=120);
```

Implementation 2

```
module disc_2d(diameter)
{
  radius=diameter/2;
  union()
    {
      circle(r=radius);
      circle(r=radius);
    }
}
module disc_3d(diameter, height)
{
  linear_extrude(height)
    {
      disc_2d(diameter);
    }
}
disc_3d(height=10, diameter=120);
```

ImplicitCAD Example: Bead

Render



Implementation 1

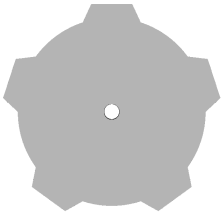
```
module bead_3d(height,
               diameter,
               hole_diameter)
{
 linear_extrude(height)
  {
   difference()
    {
     circle(r=diameter/2);
     circle(r=hole_diameter/2);
    }
  }
}
bead_3d(height=10, diameter=120,
        hole_diameter=20);
```

Implementation 2

```
module bead_3d(height,
               diameter,
               hole_diameter)
{
 difference()
  {
   cylinder(r=diameter/2, h=height);
   cylinder(r=hole_diameter/2, h=height);
  }
}
bead_3d(height=10, diameter=120,
        hole_diameter=20);
```

ImplicitCAD Example: Gear

Render



Example Implementation

```
module gear_tooth_2d(radius, angle, arc, arcdiff, height)
{
  polygon([[(radius)*cos(angle), (radius)*sin(angle)],
           [(radius)*cos(angle+arc), (radius)*sin(angle+arc)],
           [(radius+height)*cos(angle+arc-arcdiff), (radius+height)*sin(angle+arc-arcdiff)],
           [(radius+height)*cos(angle+arcdiff), (radius+height)*sin(angle+arcdiff)]]);
}
module gear_2d(diameter, hole, duty=1/2, arcdiff=1/5, teeth, tooth_height)
{
  radius=diameter/2;
  union()
    {
      difference()
        {
          circle(r=radius);
          circle(r=hole/2);
        }
      for( angle= [ (2*pi)/teeth: (2*pi)/teeth: 2*pi] )
        gear_tooth_2d(radius, angle, (2*pi)/(teeth/duty), ((2*pi)/(teeth))*arcdiff, tooth_height);
    }
}
module gear_3d(diameter, height, hole, teeth, tooth_height)
{
  linear_extrude(height)
    {
      gear_2d(diameter, hole, 0.5, 0.1, teeth, tooth_height);
    }
}
gear_3d(height=10, diameter=120, hole=10, teeth=5, tooth_height=10);
```

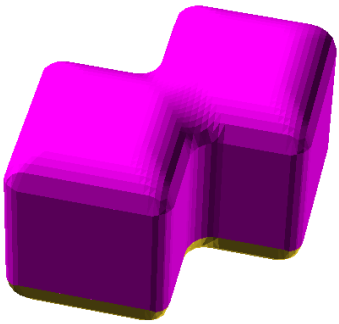
Unicorns!

ImplicitCAD allows much more than our specific use cases.

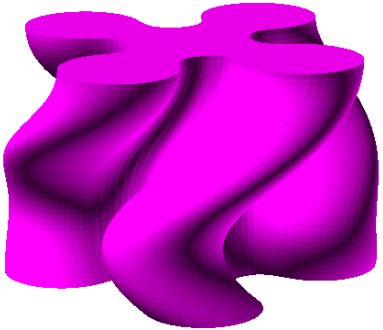
Beyond union, difference, etc.:

- Assign vars in loops
- Functional programming, e.g. `map`
- Arbitrary extrusion twist functions
- Round anything

Thus complex objects are expressed very intuitively!



```
linear_extrude (40, r=8)
  union (r=5) {
      translate ([-10,-10])
        square (30, r=3);
      translate ([ 10, 10])
        square (30, r=3);
  }
```



```
linear_extrude (height = 40, center=true,
                twist(h) = 35*cos(h*2*pi/60))
  union ( r = 8) {
    circle (10);
    translate ([22,0]) circle (10);
    translate ([0,22]) circle (10);
    translate ([-22,0]) circle (10);
    translate ([0,-22]) circle (10);
  }
```

Problems with ImplicitCAD

Three problems from outset

- Software is full of bugs
- Older versions took long time and generated large files even for simple objects
  - 17 MB, now down to 400 KB
  - $>$ 400 min, now down to $<$ 1 min
- Previous maintainer didn't deploy a test suite

I embarked on an ongoing debugging process through the course of creating viable designs for aluminum parts.

ImplicitCAD Bugs I

ImplicitCAD Bugs II

ImplicitCAD Bugs III

3D Model Prototypes

3D Model Prototypes

3D Model Prototypes

Applying Aluminum Casting to Cars

Overview

- Idea is to facilitate a Homewbrew manufacturing system for repairing, engineering, improving and personalizing cars
- Use free software, hardware, design tools, and actively published 3D models to engineer a fully-electric car

Alan

- 3D printed collection of parts that focuses on compatibility with traditional car components and portability wrt current auto manufacturing industry
- **Retrofit kit** — Compatible to fit a diversity of car models

Reinventing Cars: Security-by-Design

- Design the car from the ground-up to address serious security failings in car computing systems and mitigate unaddressed privacy concerns generated by the industry
- Encryption, reduction of reliance on surreptitios tracking technologies, data/metadata collection mitigation

Reinventing Cars: Internal Computer Network Design

- Redesign of internal networks to minimize attack surfaces through strategic sequestration of networking
- Currently: unnecessary interconnectivity across internal networks, communication/RF leakage/design-related exploits, integration of modern communication tools needs to be secured
- "Blue team" for car computer systems and communications
- Basic conceptual framework outlined but details of implementation obviously subject to revision, overhaul, review as project unfolds (we'd love to talk to anyone with ideas on these matters)

FOSSCar: Standards and Outline

- Legal/Licensing:
  — Canary, GPLv3, TAPR OHL, [. . .]

ADA

- Securing and limiting directions of comms between MCUs
- Prohibiting unnecessary connectivity via as-needed policies for data transfer across MCU "bridges," for separately purposed internal networks

ADA: Outstanding Issues

- Federal (or other regulatory) standards wrt design mandates potentially preseting obstacles
- Database systems collecting information via various infrastructures
- Devices silently transmitting data

- Specific network design details must met safety standards, incorporate electric vs. non-electric design considerations
- Very closed-source, proprietary, security-by-obscurity-based field at the moment (cease and desist!)

Join Us!