

My silly solution to a very specific silly problem

Declan Juliano

My problem

- I need an easy way to turn an image (.png) into gcode(lots of x and y points like a connect the dots. For a Laser CNC machine)
- I'm tired of spending tens of hours fiddling around with online converters and getting mediocre results with simple images

The current solution

- Draw your image
- Import image into Photoshop
- Perform a trace to vector operation
- Save the vector
- Import the vector into lightburn(laser cutter sliver software)
- Export the gcode
- Import gcode into machine software
- Make the thing on the machine

My research into this topic

After researching the basics:

How svgs store data in an xml file.

How to trace a png to get a svg.

And what programs that can currently do this.

I found there are not many that I haven't already used.

The ones I used, that are the best just aren't that great

Cura - It is a slicer for 3d printers, and it works great for 3d, but for 2d images it converts them to a lithography, leaves crisp lines as a blotchy mess.

Inkscape - Used it for years, hurts my feelings and makes my head spin. (also very laggy and slow)

WE NEED A BETTER WAY!!

My Future solution

- Draw image
- Import image into my program
- Click export
- Make thing on machine

(profit)

Design Proposal and WHY

It surprised me that there weren't any free programs that do this. It may be a bit niche but anyone with a laser cutter or simple cnc machine would benefit greatly.

To accomplish

Simply use a tracing library to convert the png to an svg,

Then use some logic to convert the paths of a SVG into a gcode file. Using user defined positions and scale. So we can 'place' the SVGs in an accurate XY plane for the machine to work with.

My Program

I decided to actually code the thing

So here are some screen shots of my genius program.

Disclaimer

This was made in a day so it is very crude and no where near finished.

The GUI

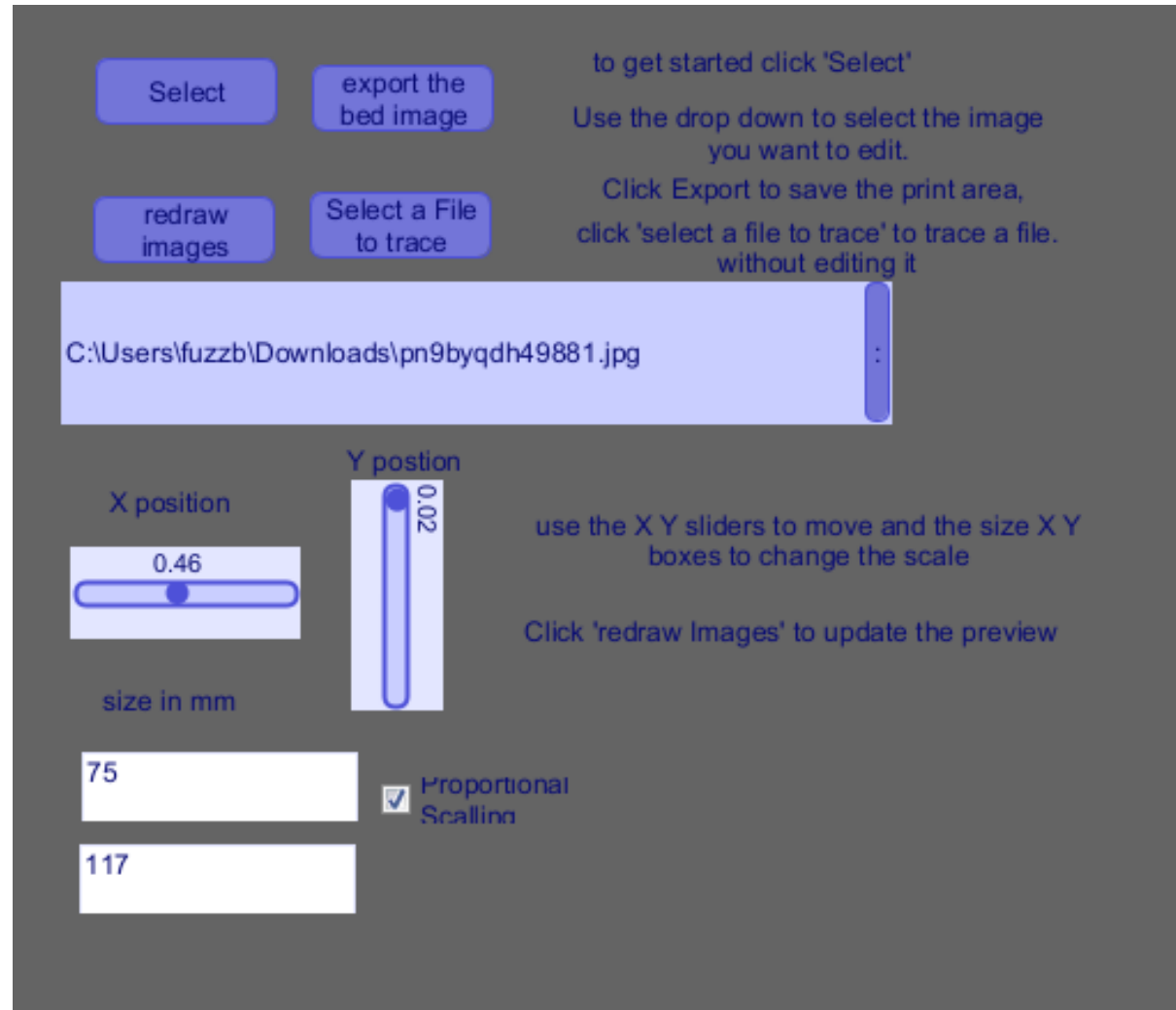
This very basic GUI gets the job done, it has numerous buttons to carry out different functions.

Select – calles loadImages()

Redraw – redraws the canvas and updates the dropdown with all images

Export – traces the bed and exports the svg

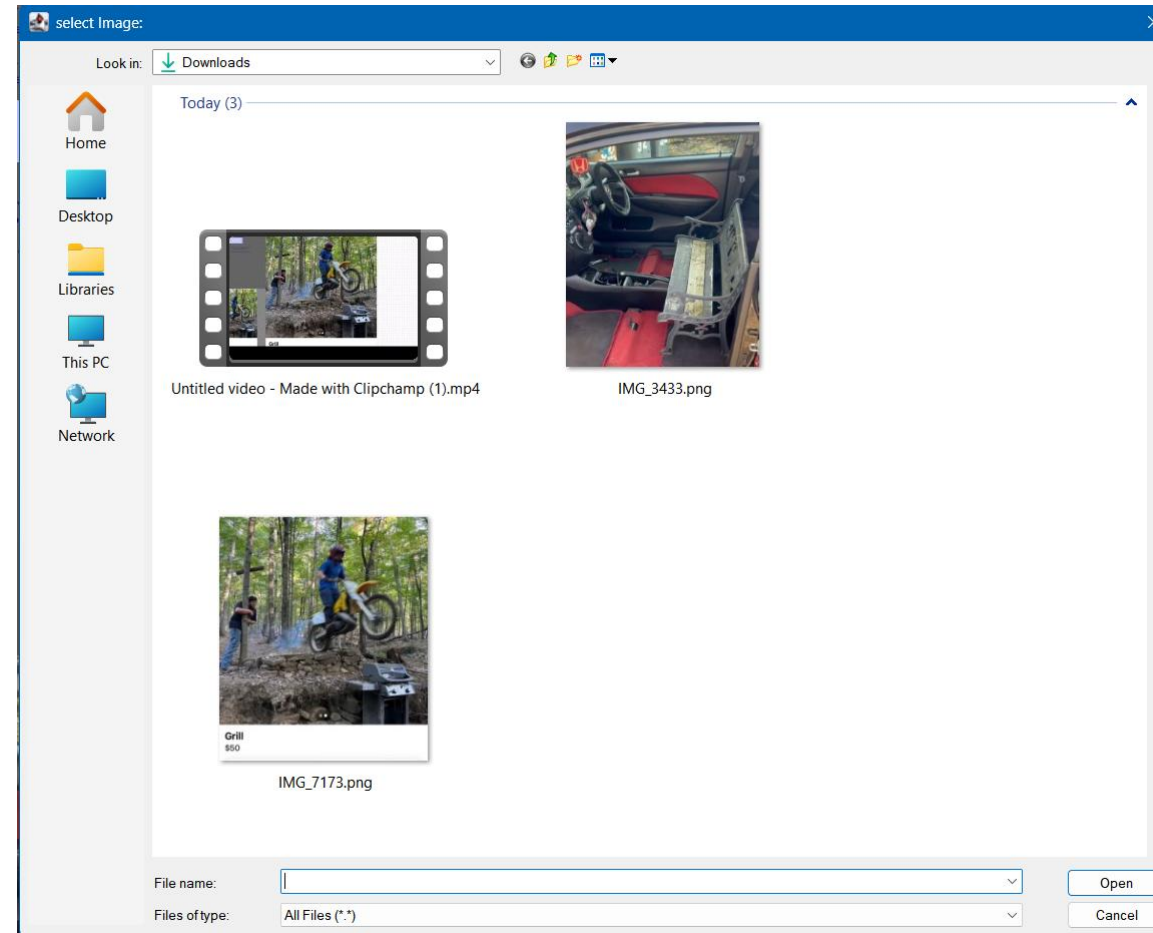
Using sliders and text boxes to modify the images, due to time constraints,



click and drag will be implemented in future releases.

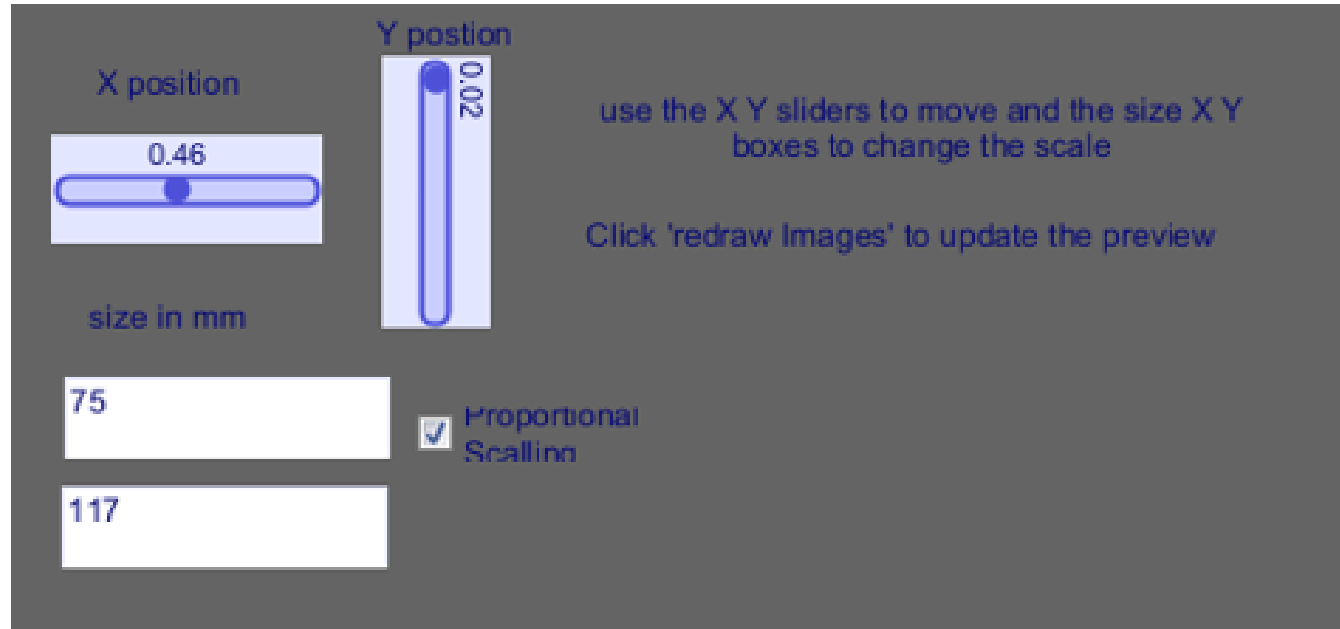
First step - Select Images

When the user clicks the select button a file window opens prompting the user to select an image



Second step – Modify the images

Select the image you want to modify from the drop down. Then use the sliders to move it in the X Y plane. And use the text boxes to set the desired width or height in mm.



The screenshot shows a software interface for modifying an image. It features two sliders: 'X position' with a value of 0.46 and 'Y position' with a value of 0.02. Below these are two text boxes for 'size in mm', with values 75 and 117. A 'Proportional Scalling' checkbox is checked. Instructions on the right state: 'use the X Y sliders to move and the size X Y boxes to change the scale' and 'Click 'redraw Images' to update the preview'.

X position: 0.46

Y position: 0.02

size in mm: 75, 117

☒ Proportional Scalling

use the X Y sliders to move and the size X Y boxes to change the scale

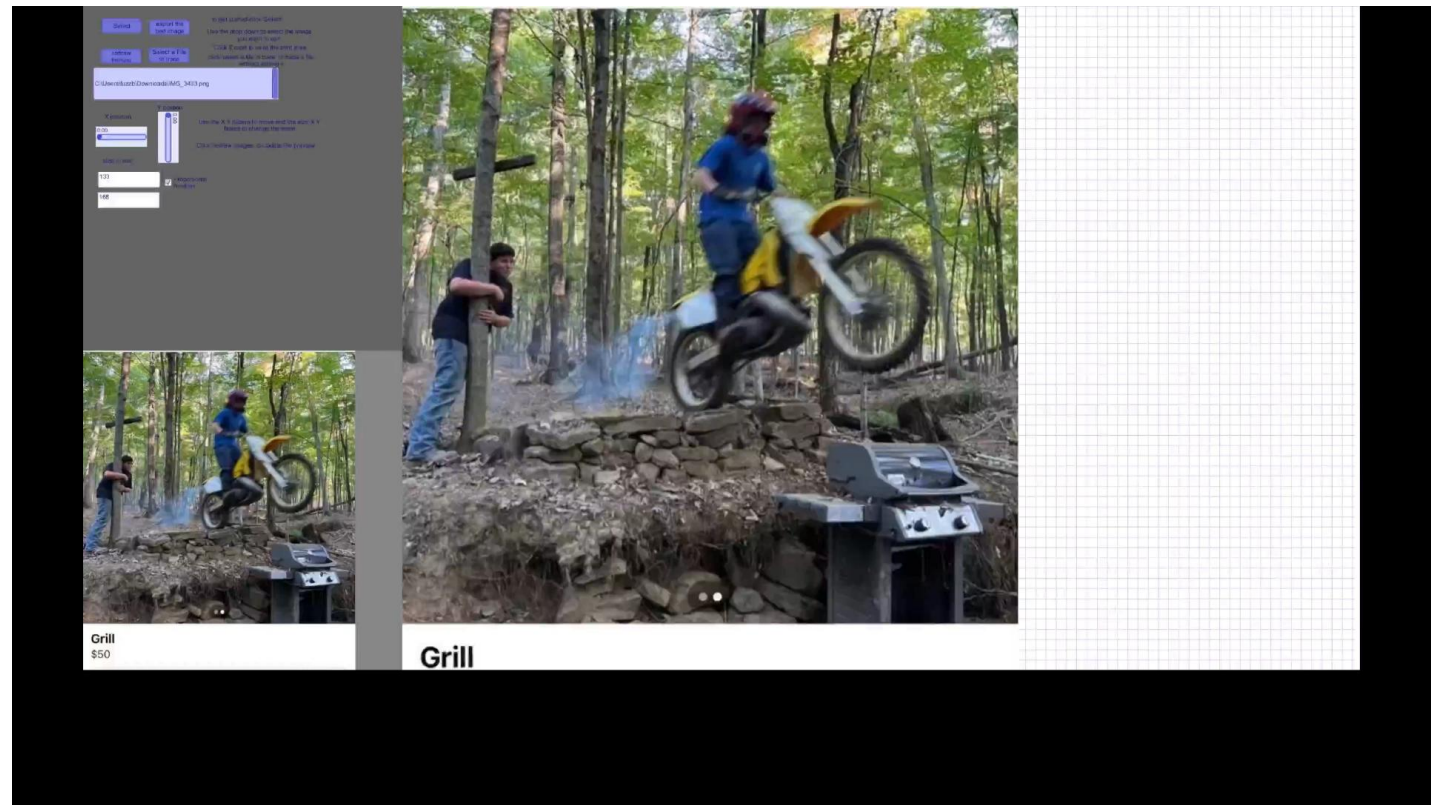
Click 'redraw Images' to update the preview

If you don't want the image to be proportionally edited then uncheck the "Proportional" check box. BEFORE changing dimensions.

Third step - Redraw Images

Once the user modifies the images, click redraw to update the bed. All images will be redrawn in their new places at their new size.

A massive chunk of code so have a video demonstration instead



Fifth Step - Export

This finalizes the user's input. Then saves the frame as a PNG with alpha, throws a prompt of where you want to save the image,

Then the code automatically traces the saved PNG as a SVG. Saving it to where you told it to.

Then we do some math to the SVG to get a gcode file out of it.

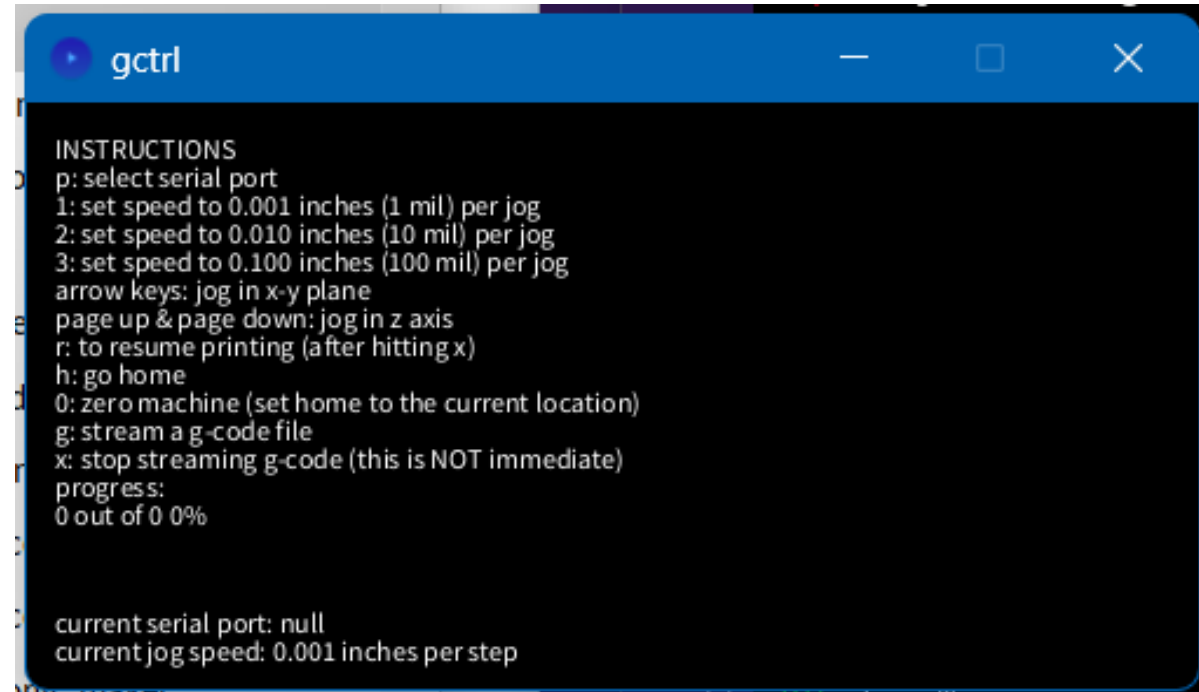
All in one button press.

Final Step – Execute the Gcode

After exporting. You can load the gcode onto your machine.

Within the same program.

A custom GCTRL window pops up with the gcode already loaded and works like normal.

A screenshot of a software window titled 'gctrl'. The window has a blue header bar with the title 'gctrl' and standard window control buttons (minimize, maximize, close). The main content area is black with white text. It displays a list of instructions for controlling a machine, including setting speeds, jogging, and streaming g-code. At the bottom, it shows the current serial port as 'null' and the current jog speed as '0.001 inches per step'.

```
INSTRUCTIONS
p: select serial port
1: set speed to 0.001 inches (1 mil) per jog
2: set speed to 0.010 inches (10 mil) per jog
3: set speed to 0.100 inches (100 mil) per jog
arrow keys: jog in x-y plane
page up & page down: jog in z axis
r: to resume printing (after hitting x)
h: go home
0: zero machine (set home to the current location)
g: stream a g-code file
x: stop streaming g-code (this is NOT immediate)
progress:
0 out of 0 0%

current serial port: null
current jog speed: 0.001 inches per step
```

Future Additions

- Click and drag for image manipulation
- Auto updating images after editing is finished.
- Clean up and make a better GUI
- Redesign GCTRL again

Open Questions

This was a relatively simple task. Which leaves me asking, why has this not been done before?

Where can I further streamline this process?

Resources

- My brain (past experiences in this topic)
- <https://stackoverflow.com/questions/8978265/java-api-to-convert-tiff-or-png-or-jpeg-to-svg>
- <https://github.com/jankovicsandras/imagetracerjava>
- <https://forum.lightburnsoftware.com/>
- https://www.reddit.com/r/hobbycnc/comments/a08lyh/what_is_the_best_way_to_convert_a_jpgpng_file_to/
- <https://forums.autodesk.com/t5/fusion-360-ideastation-archived/multiple-chain-selection-in-cam/idi-p/6796921>

All code can be found here <https://github.com/fuzzbuzzbay999/Gcode-slicer>