

Part scanning tool

November, 2016 -- Fabrica project

Overview

This document is a draft specification for a piece of software to be written.

The basic function of this software is to move a camera over the work area of a CNC machine, take photographs of the work area (scanning), and from those photographs, extract the 3D shape of the part to be worked on.

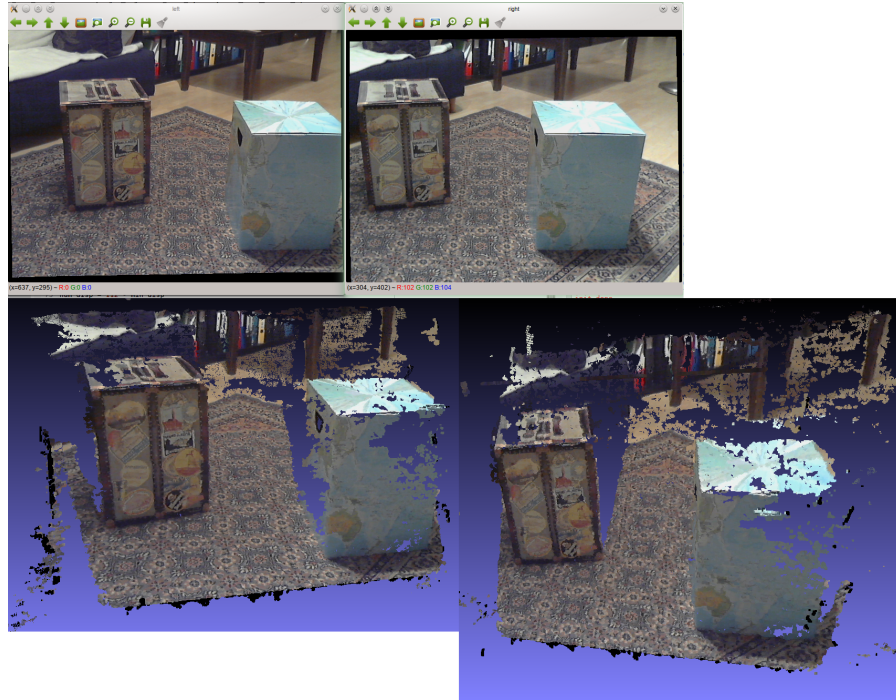
Additionally, the software should be able to use a precision probing tool to get exact dimensions on some of the part, as a means of setting the origin of a machining job.

The goal is to create an extremely simple and friendly tool for users of CNC mills and laser cutters needing to precisely locate and measure parts before machining them.

Implementation



The software runs on a dedicated Raspberry Pi, the client uses HTTP (Ajax) to talk to it, and it talks to the CNC controller (Smoothieboard) via the network too. It can be written in any language that has an OpenCV library (Python, NodeJS, Java ...).



From the client's perspective (web application in a browser, or other), things look like this :

- User wants information on what is in the work area
- Client requests an action to be executed (over HTTP)
- Tool performs action by requesting the CNC controller to move the machine or probe, taking images, analyzing them.
- Tool answers to client with picture of the work area, or a 3D representation of it, or probing data
- User sees data, decides what to do next
- Client requests an action to be executed (over HTTP)

- Etc ...

Usage

This is one part of a two-part system. This will interface with a web interface client (the other part) that will aid the user in finding the corner or center of things it wants to machine (This document only describes the server, not the client).

To put it simply, this helps set the zero position before starting a G-code file.

Setup

This only needs to be done once whenever setting up the system : the server will scan the entire surface of the work area to give the user and the server a reference of what it looks like without anything on it.



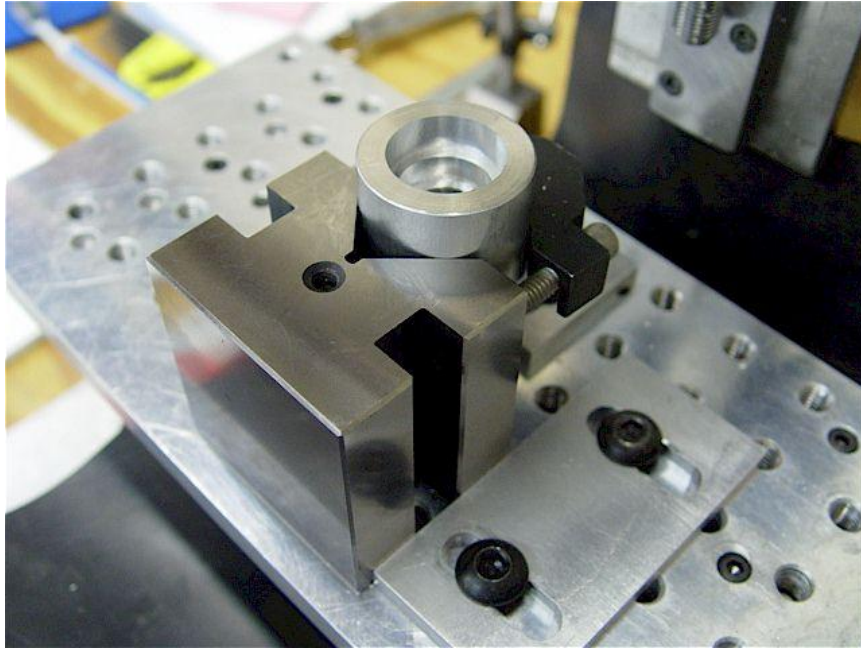
Part detection

The user can then install the part (stock) to machine on the work area. The server then takes a (composite) picture of the entire work area, and detects where the part is located. It then uses further pictures to create a 3D model of the part/stock.



Fixture detection

The user can then (if necessary) install the fixtures that hold the part in place. The server then scans the part again, and can by using a comparison, have separate 3D models for both the stock and the fixtures.



Detailed probing

Assuming the machine is equipped with a precision edge finder and/or probe, the user can then tell the interface what sort of part this is (block, cylinder, hole, complex part), the interface will propose probing points, and after user confirmation, the server then probes the machine to find the exact position of any given edge or surface, and from this generate a “sharp” 3D model of the stock, as opposed to the “fuzzy” one reconstructed from pictures.



Finally, the user can now, in the interface, select with very high accuracy, select the point at which the Gcode file should start working.

Future possible improvements include multiple cameras (on-toolhead and overlooking), as well as subtraction of gcode from scanned 3D shapes.

Interface

The client (web interface used by the user) uses HTTP (Ajax/JSON) to talk to the server. Because a lot of the tasks the client can request take much longer than typical HTTP timeouts, this works using sessions : a “job” is created, given an “id”, and the client polls to get information about the current status of the job or of any given on-going operation.

Here are a few proposals for interface calls, given as an example :

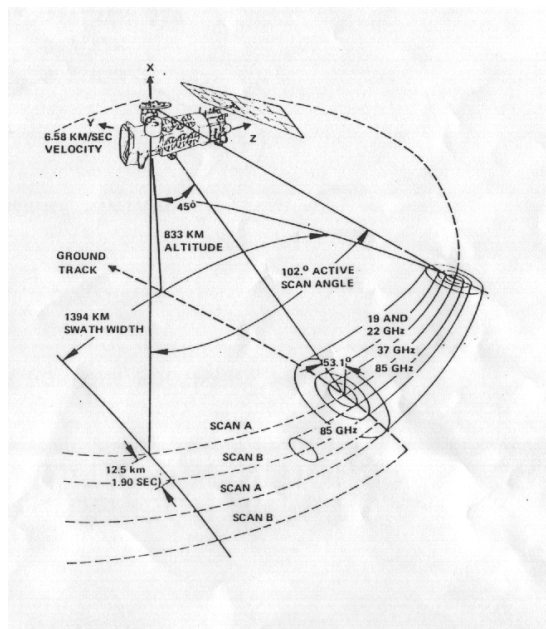
- `/create_job` returns `{job_id:'eab9c333d8'}`

- `/status {job_id:'eab9c333d8'}` returns `{status:'idle', data:['work_area', 'stock']}` or `{status:'scanning', progress:'32'}`
- `/start_scan {job_id:'eab9c333d8', type: 'stock', x:100, y:200, h:120, w:430}` returns `{error: 0}`
- `/get_data {job_id:'eab9c333d8', data:'stock'}` returns `{image_url: '/img/eab9c333d8-stock.jpg', stl_url: '/stl/eab9c333d8-stock.stl'}`

As part of any given job, the server also needs to request for the machine to move to specific locations or probe things. This is done by interfacing with the Smoothieboard via HTTP or telnet (recommended), and sending Gcode commands to it.

Scanning

Scanning refers to the operation of going over the whole work area, or a specific part of it, taking pictures at specific points (in a grid pattern).



Those images can then :

- Be stitched together into an overall picture
- Or processed into the 3D shape of whatever is on the work area (reconstruction)

Probing

Probing here refers to using a tool (probe) to find the exact position (often to within 0.01mm or better) of the surface or edge of a part.

Sonar

Another possible extension is the use of ultrasonic sonar to scan the depth of the work area to detect where parts are located (in an approximative manner).

Links :

- <http://hackaday.com/2017/04/26/get-up-close-to-your-soldering-with-a-pi-zero-microscope/>

TODO :

- Add visual 3D representations in interface using : openjscad
: <https://plus.google.com/u/0/+JeremieFrancois/posts/icDui2GWVTP?cfem=1>
-