Contents

Chapter 1

Introduction

This series of projects is about awareness - about the tools and techniques we use to see the world around us.

These projects are part of a ongoing process nourished by different motivations during my last year as a Parson's MFA DT student. I found that awareness of my surroundings was the constant and repetitive presence on my works. This was probably motivated by my own limitations and lack of attention.

Looking at the horizon we find a *skyline*. A place where the earth and the sky touch and differentiate from each other. This contour is the confluence of time and space. The landscape is shaped over time by forces like the erosion or human hand; it is in constant change and fluctuation. This skyline is also defined by the particular position of the observer. Seeing through technical devices allows us to look beyond our perception into different qualities of time and space.

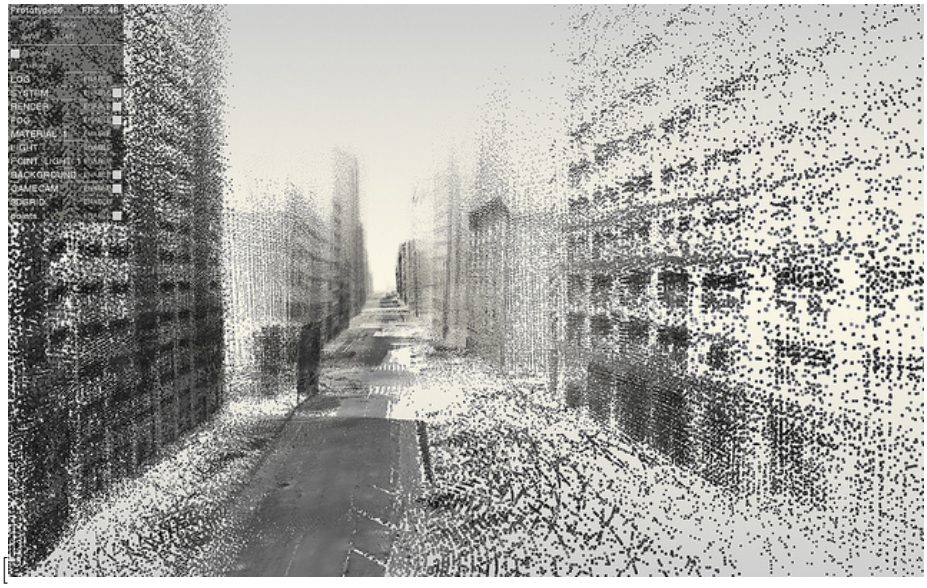
1.1 Landscapes

Landscapes, as a pictorial theme in art, speak overall about our intention to capture the world we live in. Landscapes preserve for the future a glimpse of our context, the background of our experiences, the identity of a nation, the horizon of an era.

As Rafael Cardoso said *"nature is not made up of landscapes. Although it contains mountains, valleys, plains, rivers, bays and much more, these natural forms only become 'landscape' when they are framed by the viewer's gaze and grouped into compositions. The landscape thus depends on the observer's gaze just as much as on the actual physical features been observer."* As another example, The Hudson River School, with its panoramic vistas, deep space, deft atmospheric effect and precise details define the national identity of North America scenery.

By learning to draw these sceneries, artists through time understand the laws of our perception: everything from the behavior of light to the laws of perspective were discovered by learning how to see the world. In that process we develop tools, like the camera obscura, camera lucida and lens techniques, that helps us see with more precision. Artifacts were designed to assist our perception. With them our awareness of the world has expanded dramatically. We learn how to see further and closer, to get a better picture of the macro context we are inserted into, together with the micro universe hided in details.

Because of the scale of our world, observation of space has always relied on technology. This forced us to develop and improve technology that helps us see where our eyes are blind -pushing the limits of our perception. Binoculars, telescopes, astrolabes, compasses, grids, maps, photographs and panoramic images are just some of the devices we create in this process of always looking further.



Nowadays we live in a world of images. They are easy to generate, edit and share. Effortlessly and instantaneously we produce an enormous amount of information about the world like never before in history. 3D Maps, a digital compass, GPS devices, and panoramic cameras live in a same device in our pocket. But does this mean we are more aware of our surroundings than ever before?

This thesis is a meditation about these devices. An on-going process of exploration around the theme of seeing assisted by apparatus.

1.2 The Artist as Toolmaker

Artists have been on the technological frontier since before the Renaissance. As Greg Turner and Ernest Edmonds said, *"the physical world of artifacts is very different from the conceptual world of the imagination, and artists often find themselves pushing technology forward, creating new artifacts either as part of, or in order to construct, their art. These new artifacts present new ways of using and thinking about other things."* @Art03towardsa

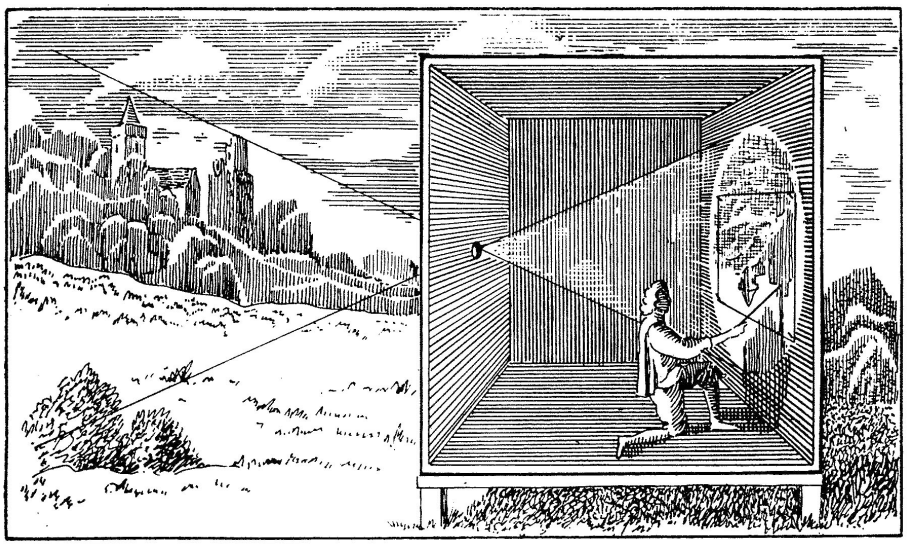


Figure 1.1: Camera Obscura

Most of these artifacts, developed by the creative process, end up as custom tools for expression. Some of them have become so popular that it is impossible not to relate them to a particular expressive field: the brush for painting, the chisel for sculpture, the notebook for poetry. Tools blend with the author to become extensions of their body and share the process of bringing imaginations to life. This can happen by learning how to use pre-existing tools, by experimenting with using tools in new ways, or by developing new tools. In the new media scene, the latter two occur often.

> *"As long as you are not defined by software, you are helping to broaden the identity of the ideas that will get locked in for future generations."* Lanier, 2010

Three decades after Xerox/Park research and the birth of the personal computer, digital media has become part of our time. Computers were developed as information machines in the early stages of what was called *Information Society*

(Beniger & Salvaggio). We have *"come to be viewed as an active processor of information."* The tools designed for this duty are the standard digital devices that we find in offices and homes, desktop computers, printers and scanners. All of these respond to standards becoming black boxes of Inputs/Outputs.

What was initially a tool for data processing slowly transformed into an expressive medium chosen by some experimental artists. As John Maeda once stated, computer technology *"is not a tool; it is a new material for expression."* New media artists in their search for new expressive and poetic potential make constant efforts to re-appropriate existing technology and at the same time push the limits of it into new directions.

This breaks with the passive paradigm of computer *"users"* to install an active attitude of creators. As Mitchel Resnick said, *"computers will not live up to their potential until we start to think of them less like televisions and more like paint brushes. That is, we need to start seeing computers not simply as information machines, but also as a new medium for creative design and expression."*. Resnick emphasizes the need for turning the Information Society into a Creative Society. *"The ultimate goal is a society of creative individuals who are constantly inventing new possibilities for themselves and their communities."*

In this spirit of self-made tools, I began a series of explorations searching for new poetic potential in between old and new apparatuses, and digital and analog media.

[

One of my biggest inspirations for these explorations was William Kentridge, a famous artist who is not usually considered a new media artist, though in his work he uses a camera and a projector to construct animations. Sometimes he

starts by filming himself performing different movements. Then, after editing the video, he projects it on top of a board and uses pieces of paper to animate frame by frame. In this process he is not only incorporating both digital and analog materials, he is sitting in the middle of a bigger apparatus. His studio becomes a tool in itself, and him a user and a vital part of it. This represents for me an excellent example of de-centralization of technology. In it, devices are reduced to a mere set of input and output devices that become part of a creative process that incorporates expressive potential from both worlds. This exquisite model of working teaches us about a flexible flow between *"real"* and *"virtual"* media, one where the artist can initialize an expansive dialog that integrates together analog and digital gestures.

As Klemmer and Hartmann warn us, *"one of the most sweeping — and unintended — transformations that the desktop computing paradigm has brought about is the extent to which the physical performance of work has homogenized."* The expressive potential of our bodies is restrained by the same gestures we use for navigating the web and writing an e-mail. In order to explore new frontiers, we need to integrate the body in new ways by modifying and re-appropriating the available technology.

Michael Polanyi refers to our physical body as *"the ultimate instrument of all our external knowledge, whether intellectual or practical experience is always in terms of the world to which we are attending from our body."* In this sense, digital creative processes that successfully incorporate the richness and rawness of direct manipulation of materials will result in a larger bandwidth of poetic and expressive potential.

Inspired by William Kentridge's methodology, I dedicated the fall of 2013 to explore the boundaries between digital and physical tools. I developed a wall plotter (vPlotter)[1] that later become a tool to think about technology and perception.

[1] http://www.patriciogonzalezvivo.com/2014/vPlotter/index.php

Chapter 2

The Process

2.1 Early experiments with the vPlotter

October 12, 2013 I assembled my first Plotter prototype based on Alexander Weber's code and designs[1]. This version used an Arduino to control the steppers' drivers and servo motors. This Arduino was receiving instructions from an OpenFrameworks application I coded through the USB serial port.

Because of my original intentions of plotting maps with this tool, I concentrated most of my efforts on resolving problems regarding precision. The movement of the plotter heads presents a lot of vibrations coming from the motors.

After experimenting with different markers, surfaces, electronic components (capacitors to compensate the excess energy between the steps of the motor) and software settings (such us augmenting/reducing/smoothing the steps per line), I discovered that the biggest enhancement was using a sponge to hold the pen. This soft material successfully absorbed most of the vibrations, generating constant straight lines.

October 14,2013 I'm happy to say the plotter is working better. Although after showing some plotting samples to Martin Mazorra (at that time my Print-Making teacher), we agreed on the value of the plotter as a bridge between analog and digital and not as a tool of precision. Since then I've start working on my own techniques to jump between mediums.

I started by using images related to the *"virtual world"*, like desktop icons, to plot (also by that time I was very interested in the desktop as a common metaphor for computer environments). These images were plotted on top of linoleum to then be carved and printed as a print relief.

[1] http://tinkerlog.com/2011/09/02/der-kritzler/

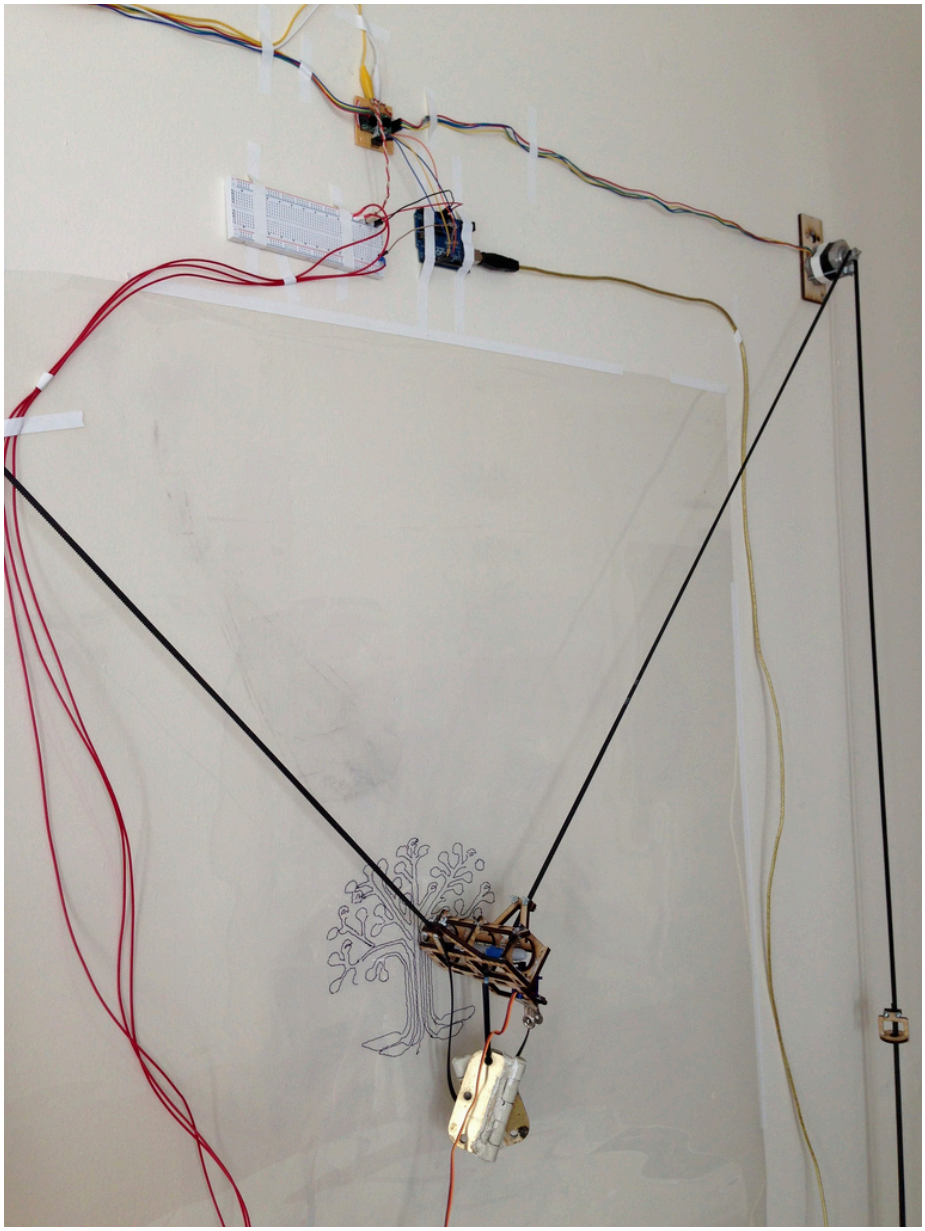


Figure 2.1: First prototype of the vPlotter

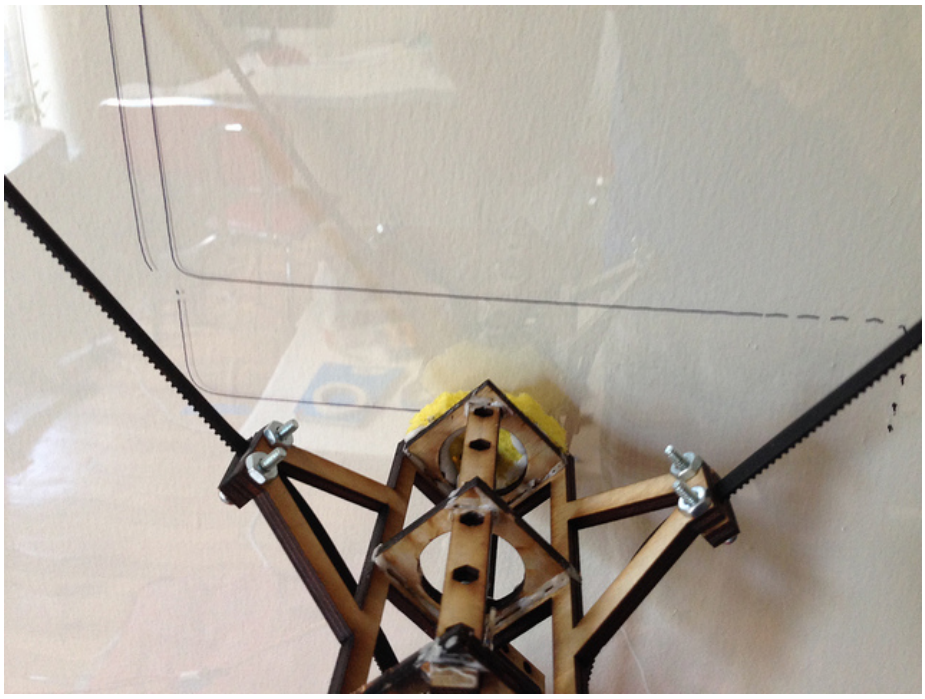


Figure 2.2: Plotter doing straight lines

October 21, 2013 After the first test with the icons I decided to go further adding more texture to the prints. I decided to work with wood because of its beautiful irregular texture.

Figure 2.3: Video-Woodcut

October 30, 2013 I became interested in how to achieve gradient patterns using just the line of the marker. Researching I found this interesting paper[2] proposing populating the surface of an image with points using a dithering algorithm, then applying a voronoi relaxation between them, and finally apply a Traveling Salesman Problem (TSP) solver to get a constant uninterrupted line between them. This approach seems to work pretty well on paper but is very difficult to carve on wood. Also I found that was necessary to add

2.2 Traces of the digital

November 16, 2013 What type of visual techniques are commonly related to digital mediums? I have been working on translating images from one medium to another. But in the end the work of my hands over the irregular surface of the wood erases almost all traces of the original virtual nature of the images. By

[2] http://archive.bridgesmathart.org/2005/bridges2005-301.pdf

generating images that clearly speak of their digital origins, I want to generate some intrinsic tension when they become clearly analog.

With that in mind I made my own slit cam combining my own GLSL Shader with the camera input.

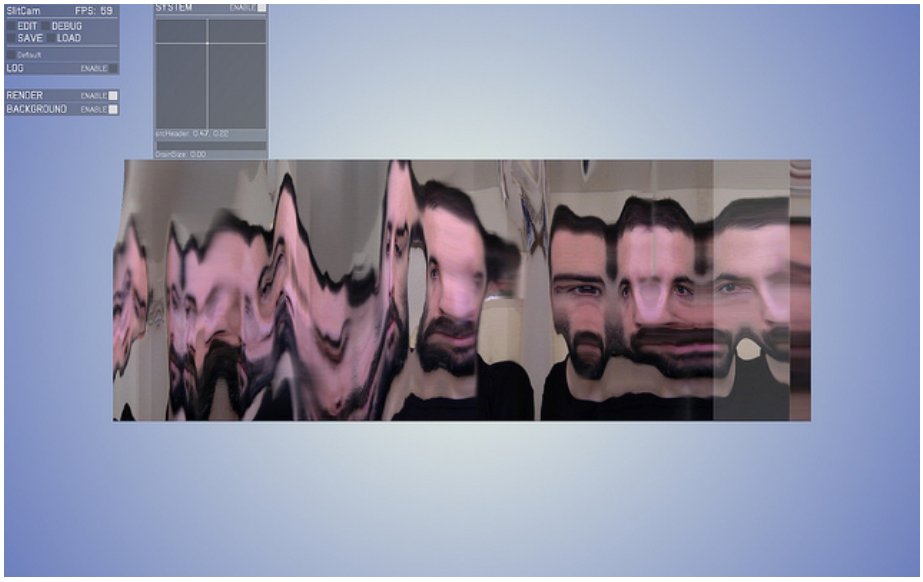


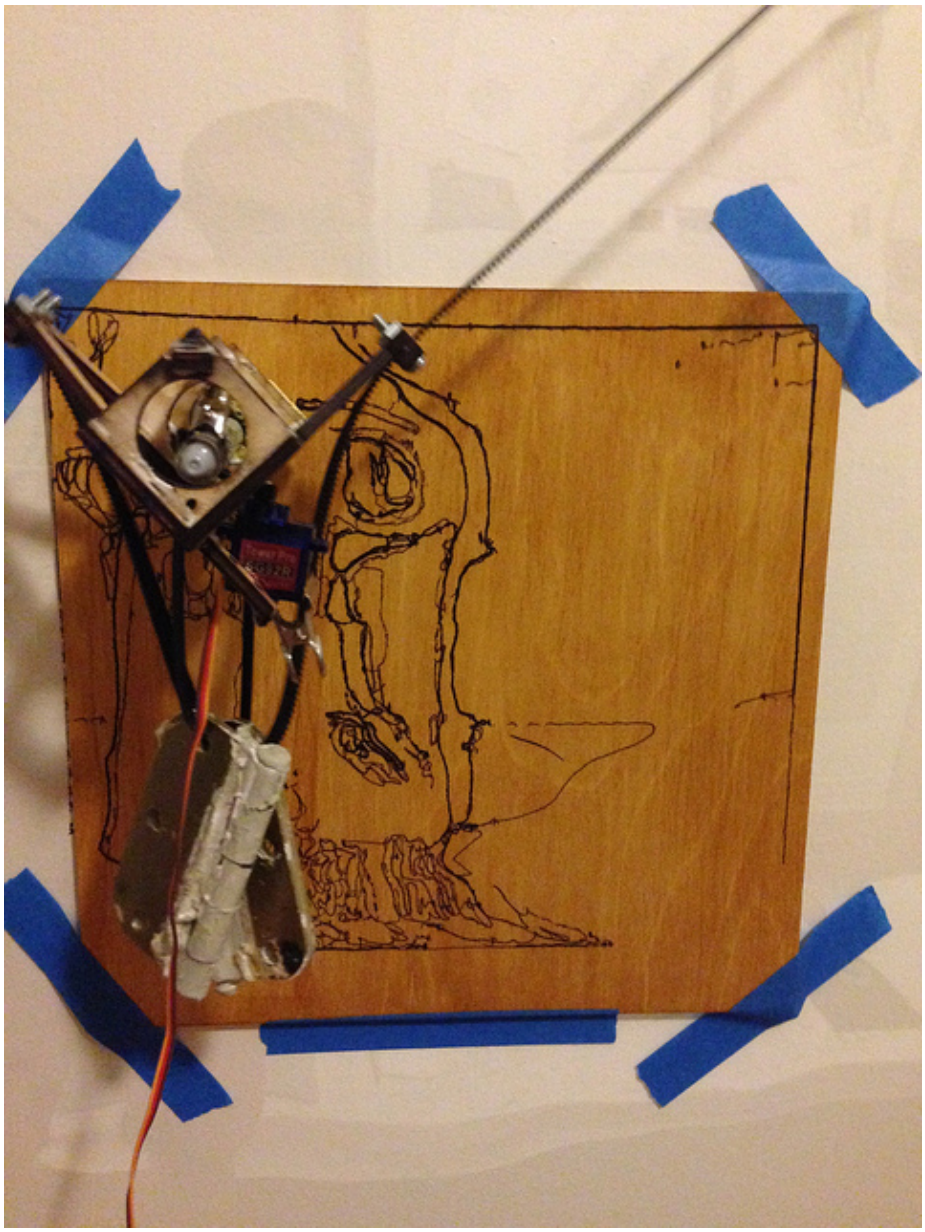
Figure 2.4: SlitCam

The second step was to extract the contours of the image applying a Computer Vision *"Canny Edge Detection"* algorithm.

After doing that I repeat the process of plotting on top of a peace of wood.

> *"At first you question the model (the seven irises) in order to discover lines, shapes, tones that you can trace on the paper. The drawing accumulates the answers. Also, of course, it accumulates corrections, after further questioning of the first answers. Drawing is correcting. […] At a certain moment - if you're lucky - the accumulation becomes an image."* John Berger, 2011

One iteration of the plotter was not enough. The jumps and spaces left by the Canny Algorithms weren't coherent enough. I discovered that running the edge detection several times and letting these traces accumulate resulted in more dramatic and rich borders.



In the print shop this process paid off. The digital nature of the slitCam, the repetition of the edge detection algorithm, the organic surface of the wood with their imperfection and the work of my hands combine in a fluid image whose nature is unclear.

December 2, 2013 I repeat the process. Makes me feel like an alchemist of mediums. The products of this process are dense and obscure at the same time

Figure 2.5: Me

superficial, incomplete.

2.3 Research and experiment about glitch

The term glitch usually refers to a technical error that leads to unexpected behavior in a system. As a clear example of "The New Aesthetics"[3], glitches speaks about the devices we use and the influence they have over us. But glitches become an aesthetic of their own. Digital artists like Kim Asendorf[4] and Adam Ferriss[5] develop glitches through code, seeking ways to exploit their aesthetic qualities. This movement is know as "Glitch Art"[6] and has many members, celebrities, conventions and festivals (ex: GLIT.CH[7]).

February 6, 2014 Intrigued by the poetic potential of this phenomena, I wonder about its limits. Do glitches manifest outside the digital? The work of Gerhard Richter[8] and Tintin Cooper[9] seem to challenge that idea. They are an example of how this aesthetic has filtered into non-digital mediums, as something recognizable and with its own agency. Both artists depart from photographs to then apply different treatments that evoke technical errors. Blurs and JPEG coding seems to emerge in the analog medium.

I feel inspired by these two artists. I have the feeling that by experimenting with the translation of mediums, I will understand more about the nature of the glitch. Like a research exercise led by the body. My hands and eyes will engage in a conversation with textures, movements and colors. I will just need to observe that process.

After weeks of collecting glitches in a Pinterest Board[10] one image is the chosen one[11], and I start to make my experiment. I grab a cardboard and start sketching the profile of it, then the whites and at last the blacks and grays. Now is the moment of the glitch. Following the directions on the image I add greens, reds and blue distortions. The colors don't mix in the right way. I remember about difference of pigment and light color theories. This is getting harder.

For most of the painting, I used the brush but its trails were too smooth to mimic the edgy shapes of the glitch. The spatula was a better tool for that. Spasmodic horizontal moves are better. I concentrate on that movement. I change the angle of the spatula in an attempt to distort the maximum amount

[3] http://booktwo.org/notebook/sxaesthetic/
[4] http://kimasendorf.com/
[5] http://adamferriss.tumblr.com/
[6] http://web.archive.org/web/20100818025410/http://www.wired.co.uk/news/archive/2010-08/17/glitch-art-databending
[7] http://rhizome.org/editorial/2010/oct/13/code-eroded-at-glitch/#c63266
[8] http://www.gerhard-richter.com/
[9] http://tintincooper.com/filter/collage
[10] http://www.pinterest.com/patriciogonzv/slit-glitch/
[11] http://robertdelnaja.tumblr.com/post/41636057857

of paint per movement. Instead the paints seems to "jump" over the surface. The wavy surface of the cardboard dictated by its own structural composition gets exposed. The glitch appears. It isn't the glitch I expected, but I think it's off course; it is a cardboard glitch.

Figure 2.6: Jumps over the cardboard

The jumps expose the structure of this system composed by acrylic paint over cardboard in the same way glitches expose the encoding algorithms of the pixels on digital images. Taking distance from the painting these jumps are all over, they quietly influence the image by adding texture to it. As the PAL/NTSC, super8 and old televisions they provide a singular and recognizable fingerprint to the image. A technological ghost scripted into the medium. I think of Instagram and all the cameras and films that they mimic.

"Any fixed contour is in nature arbitrary and impermanent. What is on either side of it tries to shift it by pushing or pulling." John Berger, 2011

The structure of the medium pushes over the image. In my paint the acrylic is dense and offers resistance. It pulls back. But digital images don't. They are like flags in the wind.

February 11, 2014 I start a second experiment. This one is about the non-resistance of digital images. Rows of pixels of a picture will be pushed by a column of pixels of a second one. The exercise will be better if the picture is

a texture, a photograph that can evoke the feeling of a surface. I also collect these types of images on a Pinterest board, I find them useful[12], like spices on kitchen shelves. One single image is distorted over and over again using different textures. The image never fights back, it is submissive and receives the distortion.

It's hard to tell how recognizable the structural ghost of the texture used to distort the image is. The texture is evoked mostly by the quality of the movement. The limitation of this experiment consists of the dimension of the displacement. It only happened in one dimension; the whole image is distorted just by one column of the second one (the one we call the texture). The eye barely reconstructs the underlying surface one row at a time, like slices of ultrasound examination.

Because the image has been distorted equally in one direction (left to right) it is easy to un-glitch. By pulling back in the opposite direction of the push, we can flatten this surface again.

February 15, 2014 My wife and I had the chance to make a short trip to Pittsburg. Watching the landscape change I saw mountains in a different way, now there were glitches. They expose an underlying system of tectonic plates; they are geo-glitches. Mountains have always evoked my emotions; I think about Berger's words - the pull and push of the landscape against the sky defining a contour. This contour is the background of our experiences. The tension of forces shape a scenario where our moments happen. Deserts, cities, forest, all of them have a distinctive profile, the exposure of the fabric of the earth, the cardboard mark of our location.

What happens if I try to un-glitch these geological irregularities? What happens if the tension of the contour that John Berger referenced, is raised to zero?

[12] http://www.pinterest.com/patriciogonzv/textures/

[

Re-using an algorithm develop together with Zach Lieberman[13] I could push back each column of pixels according to an arbitrary horizon. Like inverse sound waves that cancel each other, I visually compensate the forces of the earth over the contour of the skyline. The result is a peaceful landscape. A silent image. But looking closely, the forces of nature are trapped in a vigorous tension. The calmness is just a projection of our power. Our desire instrumented by the digital medium is temporary and superficial. This scene could explode any time, like a flooded river, nature will reclaim its power.

[13] http://thesystemis.com/

[

2.4 Ode to Slow Media

February 28, 2013 At the beginning of the Fall of 2013 I was interested in virtual environments, especially with the common adopted metaphor of a Desktop. If our life is becoming more and more digital, what are the consequences of always interacting from a "desktop"? How is our awareness influenced by the fact that we remain constantly in a digital working environment without closing time, days off, lunch breaks, sunsets, sacred spaces and celebrations.

I did a couple of experiments in this area. Trying to imagine a 3D environment that reacts to the time of the day, the task we are doing and the time we have been spending on it.

These were superficial interventions. For me, the problem was in the point of view in itself. We were still looking at the screen: shiny glowing images, digitally modified or generated. Mass produced in a glimpse of a second, instantly broadcasted, shared and copied. These images are taking over. We are staring at them. We are losing awareness by being distracted by the shininess of the digital imaginary.

> *"Videos, photographs and graphic images tend to fade in the white noise of mass culture, whereas carefully chosen image, an image made out of accurate, thoughtful brushstrokes (or any other carefully considered technique for that matter), an image that carries the wight of human touch - of human presence, of repeated analysis, of intense gazing - a full-resolution image, life-size and in real time - can be*

> *just as miraculous today as that fresco in the remote late medieval chapel."* Marc Valli

Time to experiment with another media. Time to think with my hands and body. For me, drawing and painting can trigger a dialog of thoughts better than any other medium.

Drawing are conceived as the natural and intuitive tool for ideas to be born. The double nature of connecting the material, sharable and tangible 2D world of paper with the invisible, cryptic and silent world of the mind allows ideas to shape themselves in a dialog that goes from the paper to the creator. Impossible new thoughts find ways to reality though the pencil, and by leaving their mark, form the basis for new thoughts. As matches lighting each other, drawing becomes the space between spaces, the conduction between ideas and thoughts.

> *"Since drawing can mediate between perception and reflection, it plays a constitutive role in the production and communication of knowledge"* Gansterer, 2011

There is also something about the time. It takes time to mix the right proportions of pigments to match a color. It takes time to get the right proportions of a model. It takes time for the hand to trace a contour. It takes time for the brush to fill a space. It takes time for the paint to dry out. Analog mediums are slow. But it is that slowness that holds and opens a space in time to think, observe, contemplate and meditate. It is that slowness of analog mediums which holds thoughts and awareness.

> *"The drawn hypothesis attempts to extend the space of the conjectural, deferring or delaying the effects of consequential thinking in order to keep the if in flight."* Cocker, 2011

March 27, 2014 Painting and drawings are carefully curated images. Slowly generated. They also require time to observe them. They reveal their content through time, to a patient observer. Contemplation is a slow viewing practice, one that also allows thinking, that opens a dialog with the author.

> *"Painting has offered an option - it is almost a lifestyle option - to the contemporary artist. It's slower rhythms allow for a more intimate connection to human perception (which is true both from the point of view of the artist and the viewer). For example the need for models connects painting to the intricacy of human relationships."* Marc Valli

At this point, I was more sure on the direction of my work. The tools I was making should be slow. Should use the timelessness and tirelessness of machines

to enhance and expand our perception, not to speed it up and block it. My machines should be a provocation to this world of fast images. They should slow the observer down. Make them wonder and question.

Machines originated as an assistive technology and it's important to make a clear point of that.

After winning World War II deciphering enigma our relationship with machines shifted. Machines become powerful entities that should be assisted by humans to process information. On *The Psychology of Human-Computer Interaction*, Card, Stuart K., Thomas P. Moran, and Allen Newell propose a human information processor model called GOMS (for Goals, Operators, Methods, and Selection rules). Besides the major breakthrough that this represents for its time, what is very interesting to note is the conception of human. From the epistemological frameworks and school of psychology the conception of human is based on a mechanical reduction of the human mind. Users are meant to be complex data processor entities that by connecting humans with the tireless power of the computer can bring us to a new era, but as Lanier says: *"This ideology promotes radical freedom on the surface of the web, but that freedom, ironically, is more for machines than people."*

> *"The role psychology might be expected to play in the design of the user-computer interface is suggested by the results it was able to achieve for military equipment during World War II. At that time, it had become apparent that a strong limiting factor in realizing the potential of man- machine systems, such as radar sets and military aircraft, lay in the difficulty of operating the equipment. Out of a wartime collaboration between natural scientists, engineers and psychologists came major advances, not only with respect to the man-machine systems being designed, but also with respect to psychological theory itself. Examples of the latter include the theory of signal detection, manual control theory, and a methodology for the design of cockpit instrument displays. That with psychological attention to human performance airplanes become more flyable encourages us to believe that with psychological attention to human performance computers can become more usable"* Card, Stuart K., Thomas P. Moran, and Allen Newell. 1983

The human-computer dream was one where humans can be plugged into computers in an intuitive and efficient way. By intuitive and efficient these fields understand: without too much previous explanation and without the human making mistakes. Because the psychological model behind this research is a reduction of the human psyche to terms of electronics circuits (buffers, memory, processors, inputs and outputs) the beginning of the human-computer interaction represents a project where engineers and psychologists trace the basis of how to design computers for human-computers.

Machines as assistive technology.

Machines to make us more aware.

To help us see beyond our perception without distractions.

Calmer and slower technology.

One that goes at our rhythm, to let us see, to make us think.

2.5 Skyline #01

March 28, 2014 In our personal cognitive development and in our evolution as a species, drawing is our first technique for generating images.

Drawing sets the basis for abstract thinking at the same time as we are learning to see. In this process we initiate a dialog back and forth between our eye and our tools.

This dialog also could be between things we have seen, tracing them from memory, or with things that doesn't exist yet, drawing from the imagination. In this way, memories and ideas get shaped on paper.

Cartography behaves in the same way, tracing contours and tracking paths along landscapes. Sometimes the terrain is not real and the map is from something in our mind. The work of Matthew Rangel[14] is a good example of drawing as a topographical search.

> *My work is centered on the ways in which human constructs of land influence our experience of a place and I explore this notion through an embodied process of visualizing the land. My prints are inspired by the extended walks that I initiate and often comprised of influential documentation I gather that becomes integral to my experience. This includes maps, observation-based drawings, photographs, historic research, and oral narratives I gather from ethnographic field research.*
> Matthew Rangel

March 30, 2014 I start using my vPlotter Tool[15], this time open to be guided by it. I want to see through the eyes of this tool. I start by drawing skylines.

One after the other. The plotter doesn't get tired. In fact it doesn't perceive the time. It simply obeys the series of instructions.

[14] http://matthewrangelstudio.com/
[15] http://www.patriciogonzalezvivo.com/2014/vPlooter/

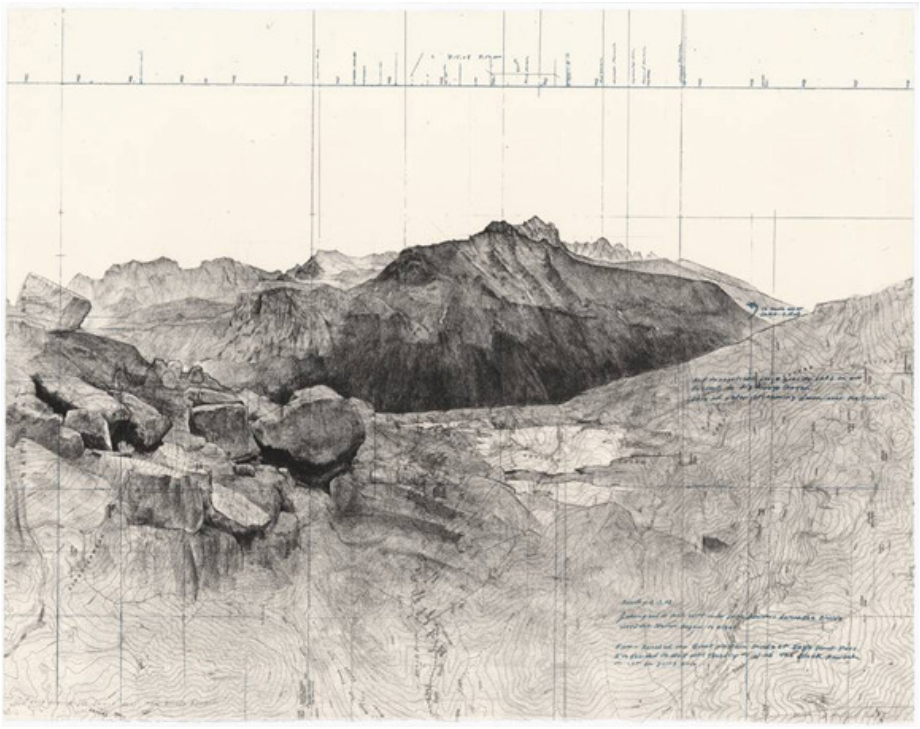


Figure 2.7: Chasing Peaks by Matthew Rangel

Now it's clear to me, how machines expand our perception to one where tiredness and exhaustion don't exist. That means I can draw by stretching the time.

As a long exposure image in photography (see Dan Holdsworth photo[16]) I can see something invisible to my perception. See through time.

[

> *"I try to encourage the viewer to think about their position in relationship to the world, trying to create some kind of communion with the viewer, instead of dictating something, or surprising them."* Dan Holdsworth[17]

I'm interested on working beyond our perception of time and space. Like in maps and long exposure pictures that necessarily imply the use of technology as an extension of our experience.

Skyline #01 is a drawing machine that slowly reveal an image by tracing constantly the contours of the images coming from a camera. The lines add one over the other one. The machine will behave like a long exposure drawing.

[16] http://www.danholdsworth.com/
[17] http://www.danholdsworth.com/

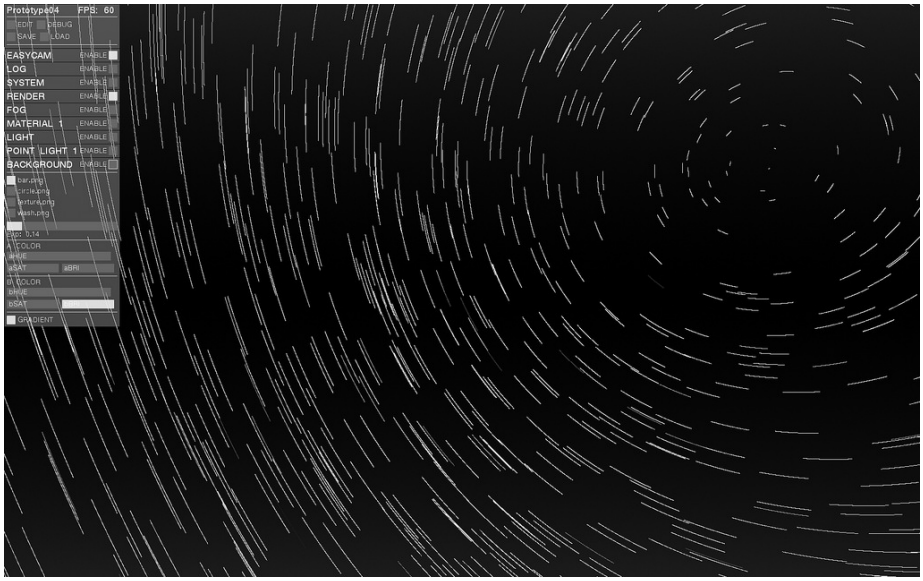


Figure 2.8: algorithmic long exposure

2.6 Skyline #02

If *Skyline #01* sees through the flow of the time. **Skyline #02** expand our perception of space.

March 13, 2014 By moving we change our point of view. We leave behind a landscapes in the search for a new horizon.

Journey diaries or traced maps are a log of the transformations that occur during a trip, a record of the past choices.

Our perception is attached to the present. We usually fail to noticing subtle changes. In a journey, the changes on the horizon becomes invisible. The skyline changes so slowly that is imperceptible.

By using machine memory we can recreate the points that constitute a journey. Google Street maps contain an enormous set of panoramic pictures. Since 2007 our landscapes and skylines have been rigorously archived. By using this collection of memories I can reconstitute the transitions in a journey from point A to point B. In my case from Union Sq. (Manhattan) all the way up to Hudson (the town where The American Landscape was born).

To let the user observe this transition of miles and miles in just a glimpse I decide to print the stitched horizon in a big roll of paper.

For this I re-purposed a mini thermal recipe printer connected to a RaspberryPi and running an openFrameworks program that I wrote which collects and

Figure 2.9: map and horizons

composes the horizons extracted from Google Street View.

For this process I develop series of digital tools in C++, like the drivers for the printer[18] and TFT Display[19], together with the C++ algorithm to extract information from Google Street View database.

March 23, 2014 Trying to improve the stitching between panoramic views I came across some encrypted depth information inside Google Street View database.

April 1, 2014 I successfully incorporate the PiTFT[20] display into the project.

April 16, 2014 After some failed enclosure prototypes...

[18] http://www.patriciogonzalezvivo.com/2014/ofxThermalPrinter/
[19] http://www.patriciogonzalezvivo.com/2014/ofxPiTFT/
[20] http://www.patriciogonzalezvivo.com/2014/ofxPiTFT/

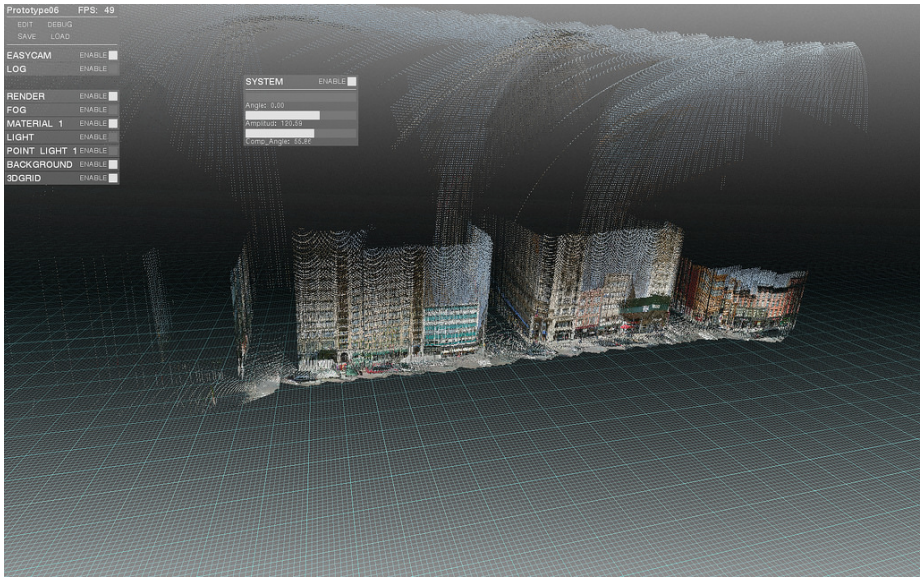
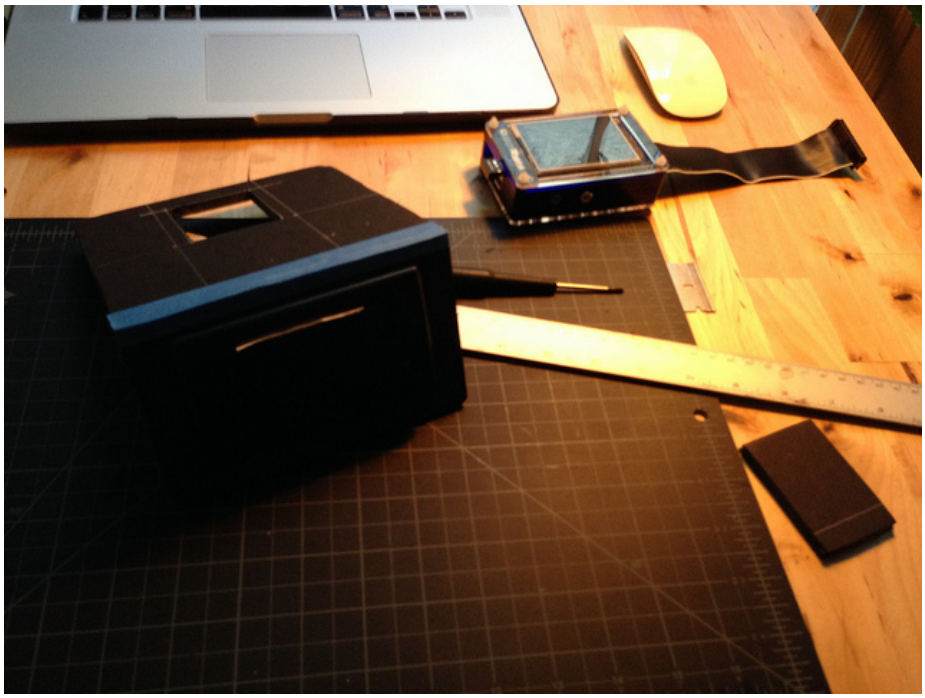
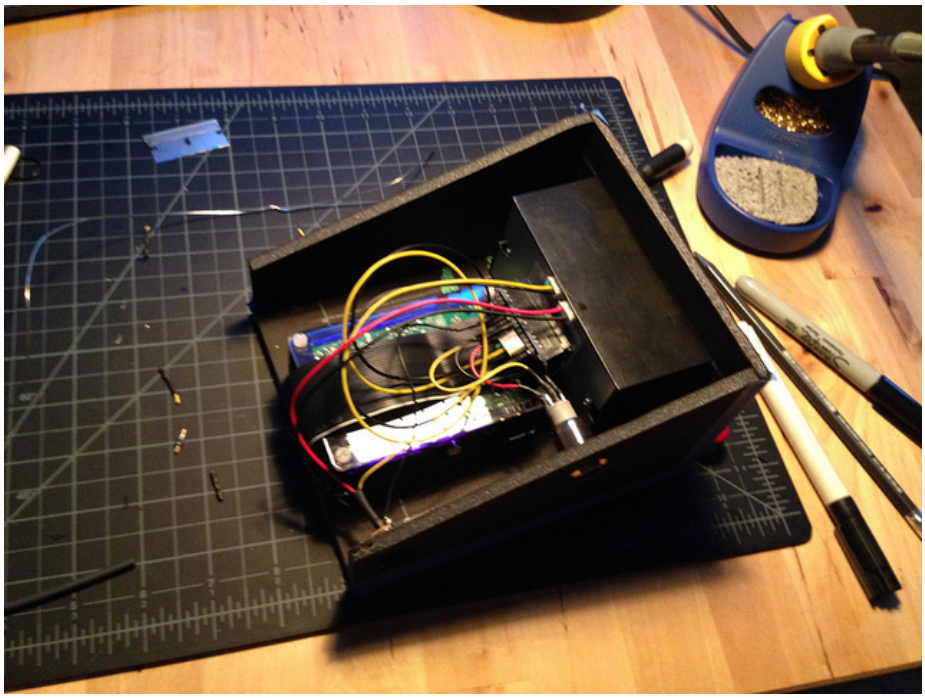


Figure 2.10: DeepInformation



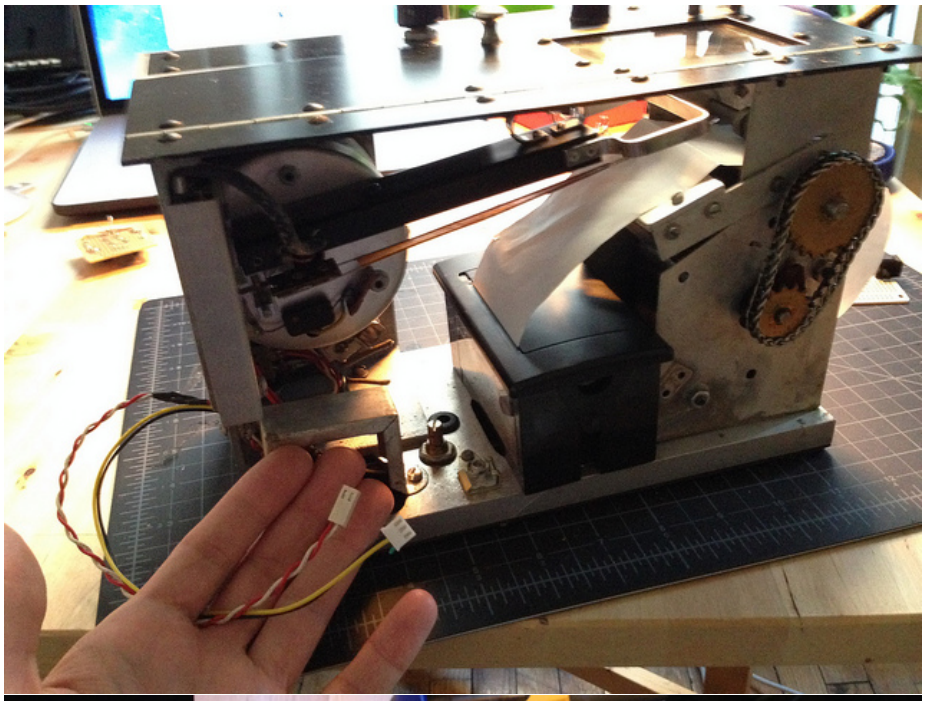


...I discover an old EKG machine that seems to be a perfect host.

With some adjustment the Mini Thermal Printer[21] and the TFT display of the RaspberryPi[22] are added. The skyline-recorder is almost ready.

[21] http://www.patriciogonzalezvivo.com/2014/ofxThermalPrinter/
[22] http://www.patriciogonzalezvivo.com/2014/ofxPiTFT

Figure 2.11: EKG Machine



Chapter 3

The Tools

3.1 vPlotter

3.1.1 Basic principes

The vPlotter is essentially a vertical drawing machine that consists of a marker suspended between two motors which moves it over a wall surface.

The math involved in the movement of the marker over the surface is extremely simple. It requires knowing the distance between the motors to calculate how much string distance each motor has to give to the marker (plotter head). This distance can be calculated using Pythagoras' theorem.

Both motors are attached to the wall at the same height to the floor separated by a given distance. The more distance there is between them, the larger the drawing area. I usually separate the motors by 1.5 meters (1500 mm).

```
distance_between_motors = 1500;
```

So the motors' positions can be represented as:

```
M1.x = 0;
M1.y = 0;

M2.x = M1.x + distance_between_motors;      // 1500
M2.y = 0;
```

Knowing this we can calculate the position of the plotter head using Pythagoras' theorem. Imagine the triangles composed by the two motors and the plotter

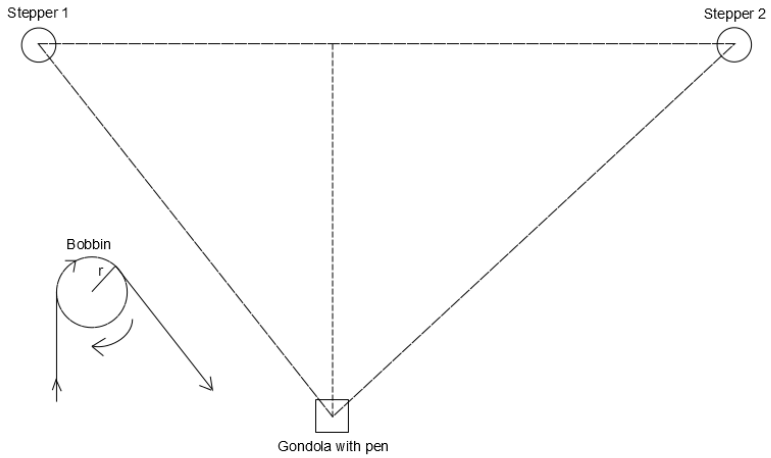


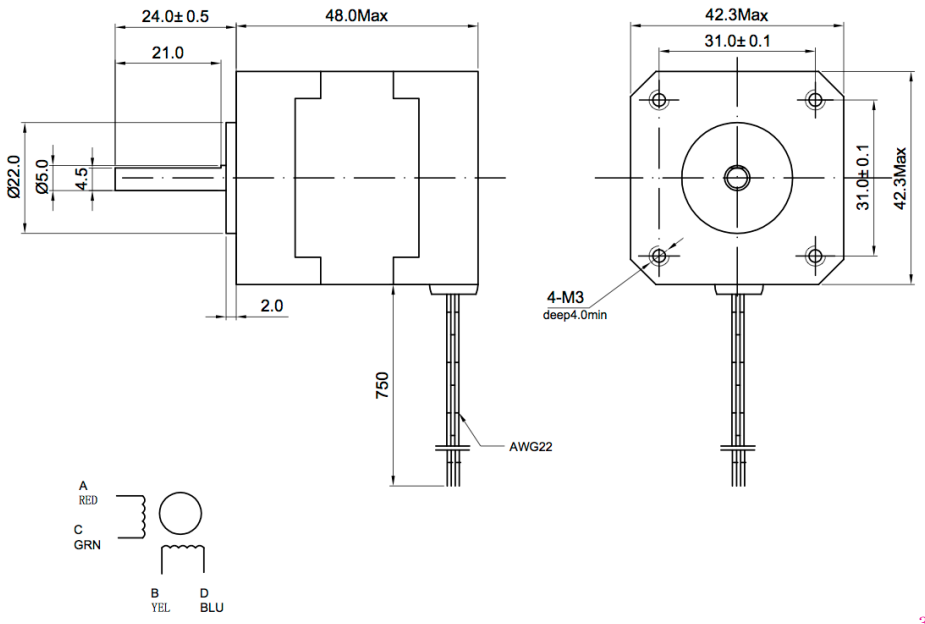
Figure 3.1: GRAPH

head as two right triangles. For each motor we need to find the distance (square root) to the marker. By releasing this amount of string to the plotter head, gravity will make the rest, by tensing the string.

```
distance.M1 = square_root( M1.x*M1.x + M1.y*M1.y );
distance.M2 = square_root( M2.x*M2.x + M2.y*M2.y );
```

To have precise control on the motors' movements, I'm using two NEMA 17 steppers motors[1] controlled with a Pololu A4988 Stepper Driver[2] for each one.

[1] https://ultimachine.com/content/kysan-1124090-nema-17-stepper-motor
[2] https://ultimachine.com/content/pololu-a4988-stepper-driver-heatsink-kit



[3]

Nema 17 motors have 200 steps for rotation but by enabling quarters of steps on the Pololu drivers we can obtain up to 800 steps for rotation which allows for much more precision. Next we need to calculate the circumference of the pulleys to determine how many millimeters we have for each step. In my case I'm using two Timing Pulley GT2 of 20 tooth[4] which have a bore of 5mm and match the GT2 Timing Belt[5] that holds the plotter header. From that we can calculate the amount of steps using these formulas:

```
STEPS = 800;
DIAMETER = 5; // 5mm
CIRCUMFERENCE = DIAMETER * PI; // 15.7 mm
m2s = CIRCUMFERENCE / STEPS; // 0.019625 mm for step
```

Keeping track of how much string each motor has moved we can estimate with a precision of 0.019625mm the position of the plotter head. I'm keeping track of the number of steps in a structured variable called MotorVal :

```
struct MotorVal{
  long M1;
    long M2;
};
```

[3] https://ultimachine.com/content/kysan-1124090-nema-17-stepper-motor
[4] https://ultimachine.com/content/timing-pulley-gt2-20-tooth
[5] https://ultimachine.com/content/timing-belt-gt2-custom-length

To calculate the plotter drawing head we can use absolute parameters or relative. This means the an absolute position or a position relative to the previous one. In order to have accurate absolute positions is necessary to know the starting position of the plotting head. For that I use a physical mark on the center of the drawing area to manually adjust the plotter header at the beginning of each drawing work. Other drawing machines like laser cutters or CNC machines have sensors to estimate the initial position.

Another important feature of this drawing machine is the ability to start and stop drawing by moving the marker closer to or further from the wall surface. This is done using a Micro Servo Motor[6] mounted on the plotter head. Servos use frequency of pulse to calculate the angle of the motor. They can go from -90 to 90 deg (0 - 180 deg), alternating between 544 to 2400 pulses per second. By moving the marker towards or away from the surface, we can control when it is drawing lines and when it is just moving. This, together with the absolute or relative coordinates, defines 4 types of commands:

```
enum Command{
  MOVE_ABS = 0,
    MOVE_REL,
    LINE_ABS,
    LINE_REL
};
```

Using these commands each line of a drawing is translated to instructions to the motors.

```
void vPlotter::print(vector<ofPolyline> _paths){
    for(int i = 0; i < _paths.size(); i++) {
        for(int j = 0; j < _paths[i].getVertices().size(); j++){
        addInstruction( ((j==0)?MOVE_ABS:LINE_ABS), _paths[i].getVertices()[j] );
        }
    }
    addInstruction(MOVE_ABS, printingArea.getCenter());
}

bool vPlotter::addInstruction(Command _command, ofPoint _pos){
    ofPoint t;

    //  ABSolute or RELative positions??
    //
    switch (_command) {
        case MOVE_ABS:
        case LINE_ABS:
```

[6] http://www.adafruit.com/products/169

```
            t = _pos;
            break;
        case MOVE_REL:
        case LINE_REL:
            t = _pos + currentPos;
            break;
    }

    //  Make instruction
    //
    Instruction inst;
    inst.set(_pos);
    inst.cmd = _command;
    inst.target = getStepsFor(_pos);

    //  Add
    //
    instructions.push_back(inst);
}

MotorVal vPlotter::getStepsFor(ofPoint _pos) {
    MotorVal val;
    val.M1 = M1.distance(_pos)/m2s;
    val.M2 = M2.distance(_pos)/m2s;
    return val;
}
```

To ensure consistency, I'm adding an extra instruction that moves the head of the plotter to the center of the printing area. This prevents the user from having to manually move the head of the plotter to the physical mark on the center of the plotting area.

These instructions are then processed in another thread sending each motor the number of steps it has to move and in which direction. Controlling real motors means being constrained by real behavior - that means being aware of the physical stress (thermal and electrical) together with the time a task takes to be executed. This is the function that executes the instructions, that calculates the pauses and pulses to the motors.

```
void vPlotter::threadedFunction(){
    while( isThreadRunning() != 0 ){
        if (isPrinting()){
            if(getInstructionsLeft()){
                if (exeInstruction(instructions[0])){
                    if(lock()){
                        instructions.erase(instructions.begin());
```

```
                    unlock();
                }
            }
        } else {
            bPlotting = false;
            stopThread();
        }
    }
}

bool vPlotter::exeInstruction(Instruction _inst){
    MotorVal t, s;
    float sd,pd;

    //  Set the right PEN
    //
    sd = stepDelay*1000.0f;
    pd = penDelay*1000.0f;
    switch (_inst.cmd) {
        case LINE_REL:
        case LINE_ABS:{
            if(penState == PEN_UP){
                softServoWrite(SERVO_PIN,penPosDown);
                usleep(pd);
            }
            penState = PEN_DOWN;
            break;
        }
        case MOVE_REL:
        case MOVE_ABS:{
            if(penState == PEN_DOWN){
                softServoWrite(SERVO_PIN,penPosUp);
                usleep(pd);
            }
            penState = PEN_UP;
            break;
        }
    }

    // Tmp Variables
    t = target = _inst.target;
    s = steps;

    // set directions
    MotorVal dir;
```

```
    dir.M1 = (t.M1 > s.M1) ? +1 : -1;
    dir.M2 = (t.M2 > s.M2) ? +1 : -1;
    digitalWrite(DIR_PIN_M1, (t.M1 > s.M1) ? DIR_UP : DIR_DOWN);
    digitalWrite(DIR_PIN_M2, (t.M2 > s.M2) ? DIR_DOWN : DIR_UP);

    //  Make steps
    //
    while (s!=t){
        if( t.M1 - s.M1 != 0){
            digitalWrite(STEP_PIN_M1, HIGH);
            s.M1 += dir.M1;
        }

        if(t.M2-s.M2 != 0){
            digitalWrite(STEP_PIN_M2, HIGH);
            s.M2 += dir.M2;
        }

        steps = s;
        currentPos = getPosFor(s);
        usleep(250);
        digitalWrite(STEP_PIN_M1, LOW);
        digitalWrite(STEP_PIN_M2, LOW);
        usleep(250);
        usleep(sd);
    }
    return true;
}

//  Reverse kinematics
//
ofPoint vPlotter::getPosFor(MotorVal _steps) {
    float a = _steps.M2 * m2s;
    float b = _steps.M1 * m2s;
    float c = motorsDistance;
    return calcPointB(a, b, c);
}

ofPoint vPlotter::calcPointB(float a, float b, float c){
    double angle = acos((b*b+c*c-a*a)/(2*b*c));
    return ofPoint(b*cos(angle)+M1.x,b*sin(angle)+M1.y);
}
```

These functions are the core operations behind the drawing. To make the vPlotter a proper tool I designed a reliable and flexible infrastructure based on openFrameworks (C++) and Node (JS) on top of a Linux OS running on a

raspberryPi.

3.1.2 Hardware

- 2 x Nema 17 Stepper Motor[7]
- 2 x Pololu A4988 Stepper Driver[8]
- 2 x Timing Pulley, GT2, 20 Tooth[9]
- 6m of Timing Belt, GT2, Custom Length[10]
- 1 x Micro Servo[11]
- 15x15 inch of Plywood
- Raspberry Pi[12]
- GPIO Ribbon Cable[13]
- Adafruit Half-size Perma-Proto[14]
- Optional PiTFT[15] screen for visual debug

RaspberryPi is a small and cheap board capable of running a complete Linux system. The RaspberryPi has some General Input's and Outputs (GPIO) that will let us connect this board to the stepper motors' drivers (Pololu) and the mini servo motors.

Because the servo uses frequency of pulses it will use the GPIO number 18 that has PWM capabilities. For the two motor drivers we use the GPIOs 4, 17, 23 and 22. Each driver will require both a *step* and a *direction* pulse to know when and how to make the rotation. These values can be modified at the top of the `vPlotter.h` file

```
#define SERVO_PIN 18

#define STEP_PIN_M1 4
#define DIR_PIN_M1 17

#define STEP_PIN_M2 23
#define DIR_PIN_M2 22
```

[7] https://ultimachine.com/content/kysan-1124090-nema-17-stepper-motor
[8] https://ultimachine.com/content/pololu-a4988-stepper-driver-heatsink-kit
[9] https://ultimachine.com/content/timing-pulley-gt2-20-tooth
[10] https://ultimachine.com/content/timing-belt-gt2-custom-length
[11] http://www.adafruit.com/products/169
[12] https://www.adafruit.com/products/998
[13] https://www.adafruit.com/products/862
[14] https://www.adafruit.com/products/1148
[15] https://www.adafruit.com/products/1601

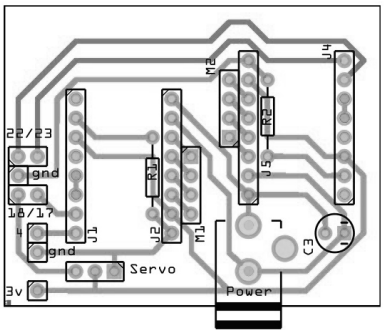


Figure 3.2: Electronic Board

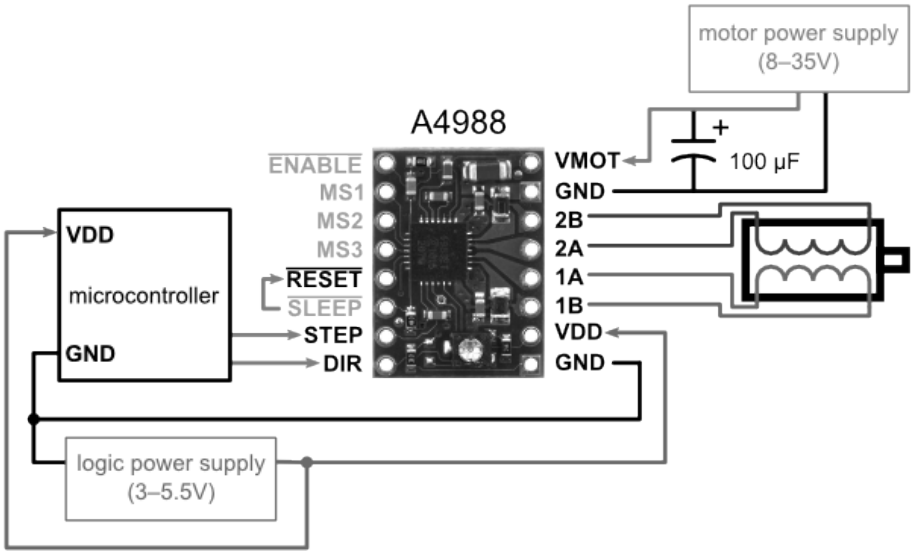


Figure 3.3: A4988

3.1.3 Software

The RaspberryPi uses a derived version of the Debian Linux distribution as its operating system. This software architecture is famous for it stability, security and modularity. Each new program installed onto the computer becomes part of a robust ecosystem of applications that can be re-combined in different ways.

In the case of the vPlotter the C++ code I wrote using openFrameworks constitutes the core application capable of being executed directly from the command line as follows:

```
vPlotter -i file.svg
```

It is possible to modify some setup parameters like the distance between motors by doing:

```
vPlotter -d 2000 -i file.svg
```

This application can print vector files (known as SVG format) and can also listen for instructions on a OSC port (for example the port 1010101).

```
vPlotter -o 101010
```

Another feature of this application is the possibility of receiving visual feedback of the instructions on the pull by adding the option -x

```
vPlotter -x -o 1010101
vPlotter -x -i file.svg
```

It's possible to combine vPlotter with other tools using Unix pipe commands like this:

```
potrace -s image.png || vPlotter -i
```

In this case we are using the program 'potrace'[16] to transform an image to vectors (-s) to then print it using vPlotter. Using similar techniques, it is possible to transform a picture of a document to text to then assign some style and fonts to then vectorize and finally print with the plotter. The combinations are infinite.

But not all users are comfortable running linux commands as part of their creative practice. That's why I design a Node.js server that provides an web API to administrate calls to the vPlotter. Users can ask for jobs in

[16] http://potrace.sourceforge.net/

the plotter queue together with the setup parameters by making a call to `rpi.local:8080/status.json`. Using the same API users can `add` tasks to the queue or change the setting options of the vPlotter. A hypothetical user with intermediate knowledge of HTML/CSS/JS can use the plotter remotely by sending and receiving calls to this API.

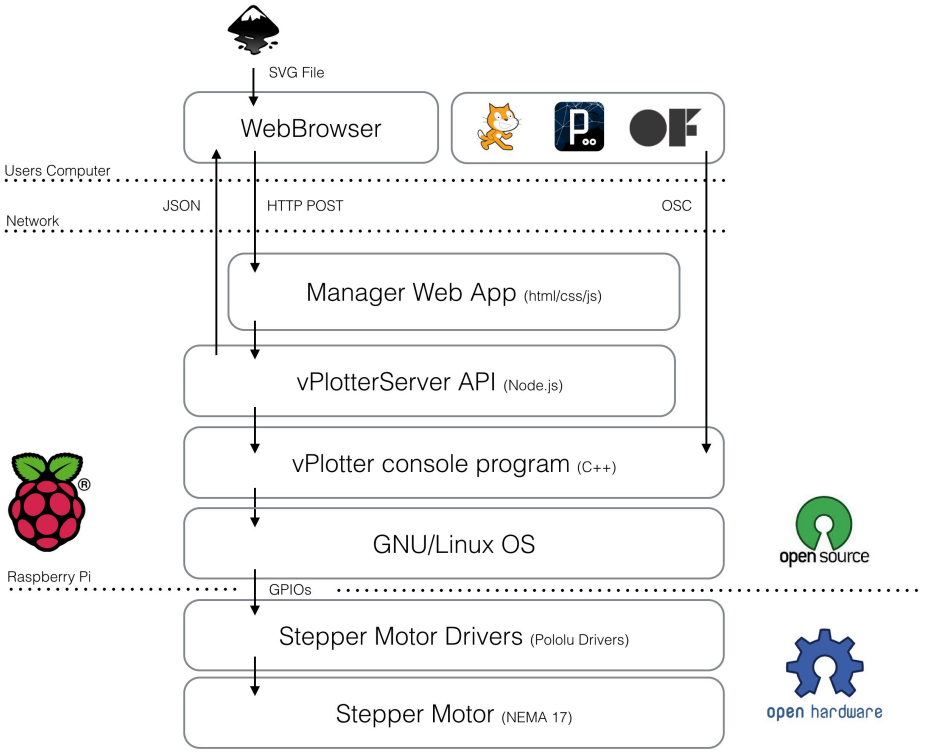


Figure 3.4: Modular design of the vPlotter

It's possible that the user does not have the technical knowledge to use the API, which is why I have also incorporated on the server a web site from which it is possible to drag & drop files to the queue. I've designed a tool that can be access ed from a wide spectrum of architectural layers.

3.1.3.0.1 Software Installation

1. In order to compile and install the vPlotter binaries in the right place you will need to install openFrameworks[17]. For that please follow this guide[18]. Be prepared, it is going to take a while.

[17] http://openframeworks.cc/setup/raspberrypi/Raspberry-Pi-Getting-Started.html
[18] http://openframeworks.cc/setup/raspberrypi/Raspberry-Pi-Getting-Started.html

2. Once you have your openFrameworks installed and working (please check it by compiling some of the examples), clone this repository inside the `/home/pi/openframeworks/apps` folder.

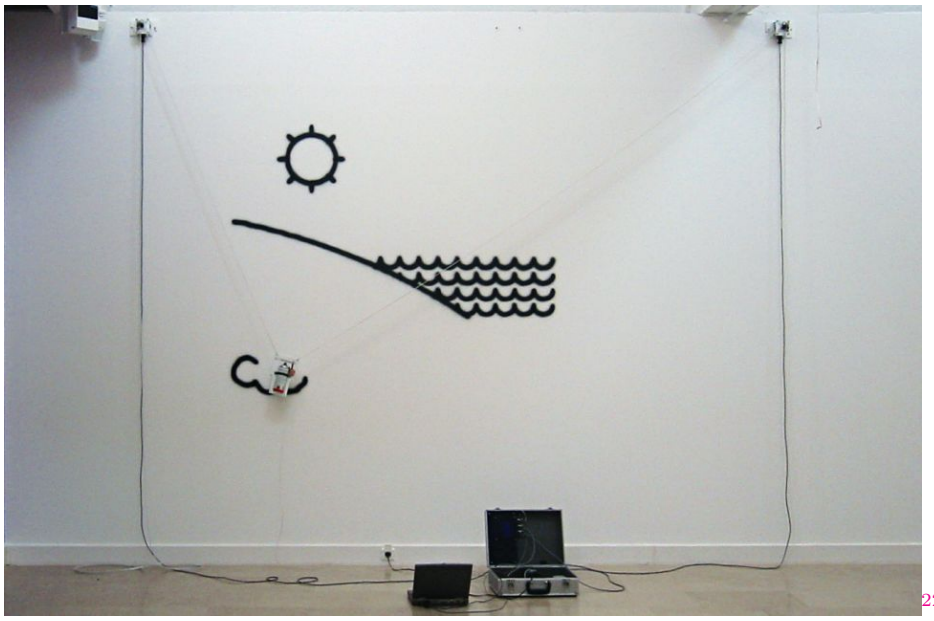
```
cd ~/openFrameworks/apps
git clone https://github.com/patriciogonzalezvivo/vPlotter
```

5. The last step is to run the installation script which will install Node.js[19] for the server, WiringPi[20] to communicate through the GPIOs, compile `vPlotter` and install `vPlotterServer` as a local daemon.

```
cd ~/openFrameworks/apps/vPlotter
./install.sh
```

3.1.4 Precedents & Research

Wall Plotter has been around for a while. My work is continuing what others have started. The first documented wall-plotter was develop by [Jurg Lehni]((http://juerglehni.com), and was called Hektor[21]. Jurg is a fascinating artist committed to tool making, especially related to drawing.

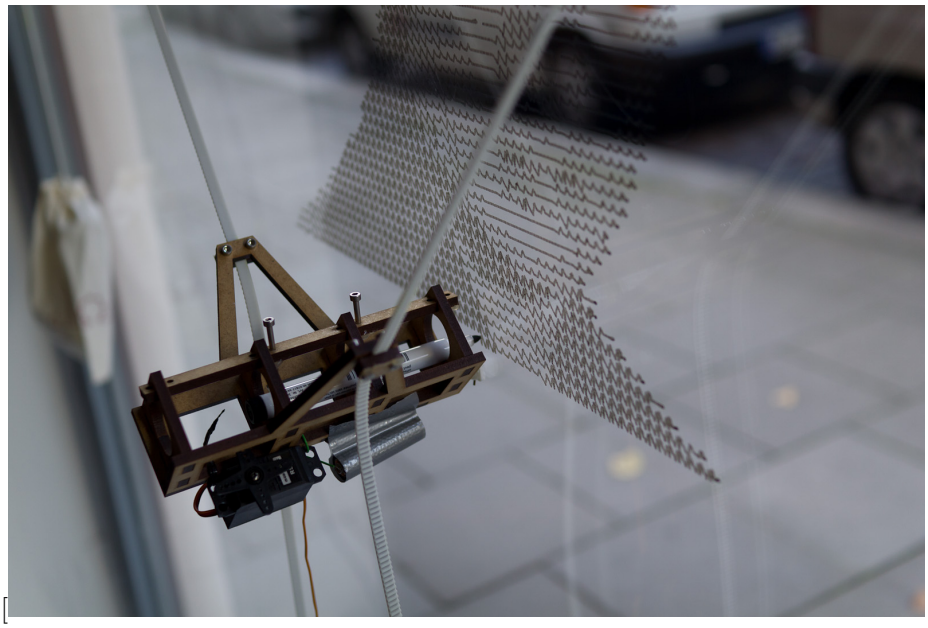


[22]

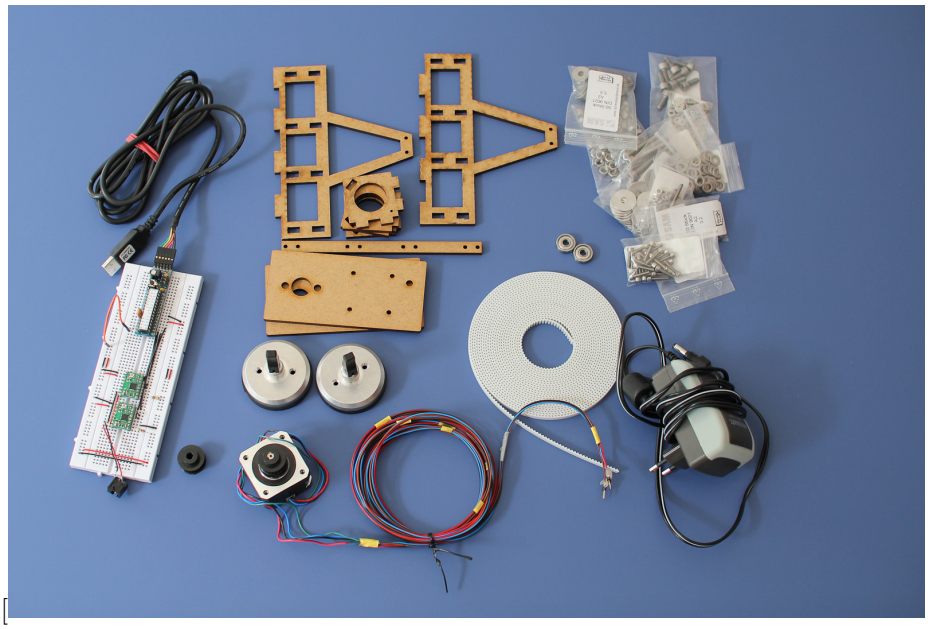
In my case I start working based on Alexander Weber's[23] wall plotter called

[19] http://nodejs.org/
[20] https://projects.drogon.net/raspberry-pi/wiringpi/
[21] http://juerglehni.com/works/hektor/
[22] http://juerglehni.com/works/hektor/
[23] http://tinkerlog.com/

Kritzler[24]. He's done amazing work documenting the process together with comprenhensible code[25]. This version is designed to run with an Arduino plugged into a computer. My first effort on this project was porting the code to openFrameworks to then compile and run on the RaspberryPi, but the basic core logics behind the algorithmic kinematics are pretty much the same as what Alex wrote.



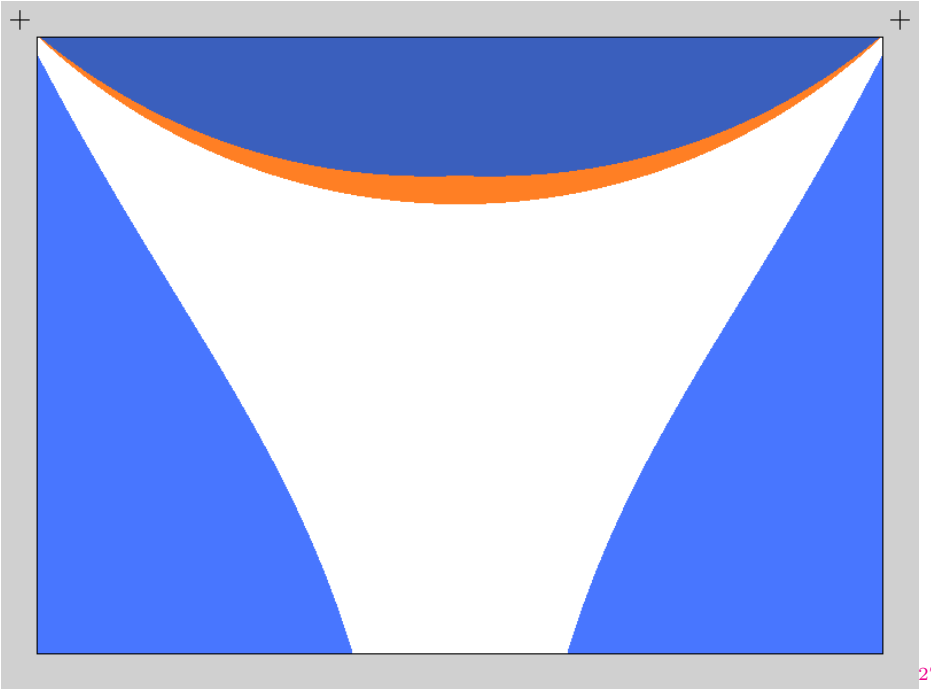
[24] http://tinkerlog.com/2011/09/02/der-kritzler/
[25] https://github.com/tinkerlog/Kritzler



While I was experimenting with the Plotter I rapidly discovered how important it is to calculate the right printing area. The variations on the tensions on the strings that hold the plotter head are vital to obtain good results on the marker traces.

Doing some research I found some valuable documentation on this website[26] of how to obtain both resolution and tension coefficient. Since they are presented in Python I translated them into C++ and added to some methods and functions that calculate the best area given a specific distance between the motors.

[26] http://2e5.com/plotter/V/design/

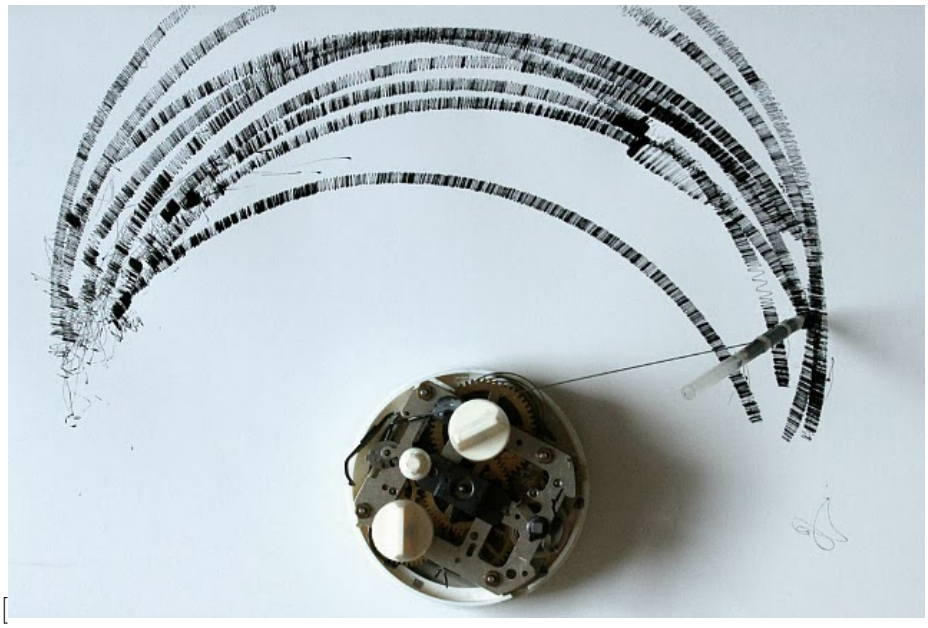
 [27]

The presence of mathematics concepts in drawings have a long history. The laws of perspective and proportions are one of the first technical tools we use. In my process of making a drawing machine I learned how to connect and work together with physics and mechanical principles.

Sometimes drawing tools aren't born from a long history of techniques and apparatus. They are discovered as the repurpose of everyday machines. This is the case of the drawings made by Echo Yang[28].

[27] http://2e5.com/plotter/V/design/

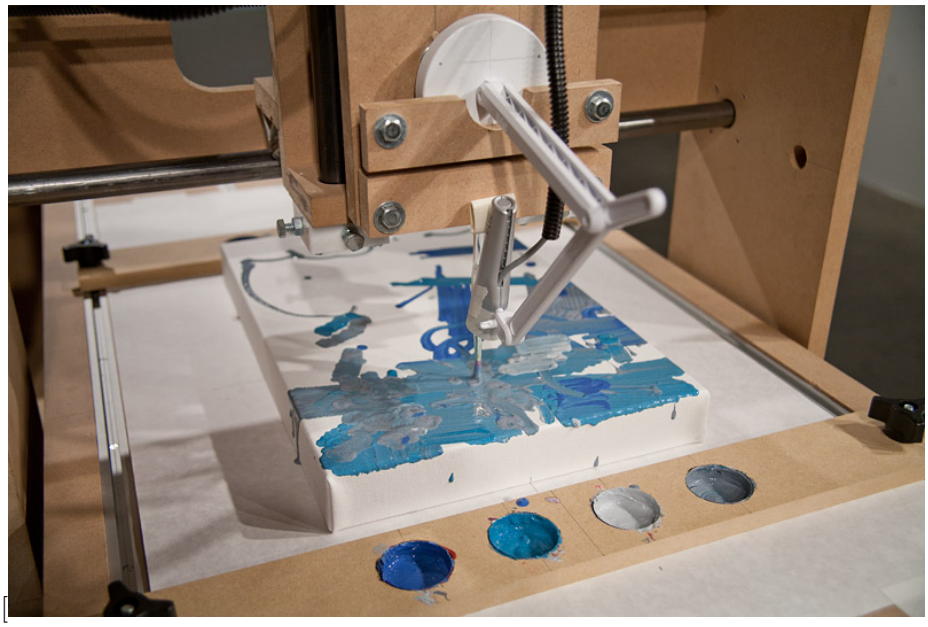
[28] http://www.junk-culture.com/2014/02/designer-transforms-old-school-analog.html#more



[

There are other examples of drawing machines, ones that formulate a question not about the tools but about the user. These are the cases of sophisticated robots armed with regular brushes and analog paint. These robots mimic our use of these tools in the search of becoming autonomous agents. eDavid[29] designed by Oliver Deussen and Thomas Lindemeier is an example of a machine that performs accurate drawings, while the robot designed by Benjamin Grosser[30] explores some "expressive" visual representations of sound.

[29] http://www.informatik.uni-konstanz.de/en/edavid/news/
[30] http://bengrosser.com/projects/interactive-robotic-painting-machine/



3.1.4.0.2 More references to drawing machines

- Viktor[31]
- Norwegian Creation's Drawing Machine[32]
- SADbot[33]
- Harvey Moon's The Drawing Machine[34]
- http://www.creativeapplications.net/maxmsp/drawing-machine-maxmsp-processing/
- Der Kritzler[35]
- http://www.polargraph.co.uk
- Tristan Perich's Drawing Machine[36]
- Paul Ferragut's Wall Drawing Machine: http://vimeo.com/20526787 http://vimeo.com/35647507 http://vimeo.com/26900180

[31] http://www.we-find-wildness.com/2011/01/juerg-lehni-viktor/
[32] http://www.norwegiancreations.com/2012/04/drawing-machine-part-2/
[33] http://www.instructables.com/id/SADbot-the-Seasonally-Affected-Drawing-robot/
[34] http://www.unanything.com/Drawbot.html
[35] http://tinkerlog.com/2011/09/02/der-kritzler/%20Polargraph
[36] http://tristanperich.com/Art/Machine_Drawings/

- Douglas Irving Repetto's Giant Painting Machine:http://music.columbia.edu/~douglas/portfolio/GPM_sanmateo_peoria_sf/ http://music.columbia.edu/~douglas/portfolio/GPM_milano/
- Sprite Mod's White Board Plotter[37]
- DrawBot:http://www.marginallyclever.com/category/drawbot-my-creations/ https://github.com/i-make-robots/DrawBot http://web.me.com/sami6877/unanything/Site/Secret_Work/Entries/2010/6/1_Drawbot.html http://blog.makezine.com/2010/07/30/drawbot-takes-its-time-to-sketch/
- AS220 Lab's Drawbot[38]
- Make Magazine's Drawing Machine[39] http://makermedia.cmail4.com/t/ViewEmail/r/351C80A070878603/6423113868B805CEC9C291422E3DE149 http://blog.makezine.com/2011/02/24/make-it-last-build-3-rigging-up-your-drawbot/
- Muralizer[40] http://www.kickstarter.com/projects/1910641777/muralizer-it-prints-on-walls http://blog.makezine.com/2009/10/29/muralizer-prints-art-on-the-wall/
- InternBot[41]

[37] http://spritesmods.com/?art=whiteboard&f=had
[38] http://www.as220.org/labs/drawbot/instructions.html
[39] http://makezine.com/makeitlast/
[40] http://www.muralizer.com/blog/
[41] http://jamesprovost.com/blog/introducing-internbot/