LIBRE GRAP HICS

2.4

Capture
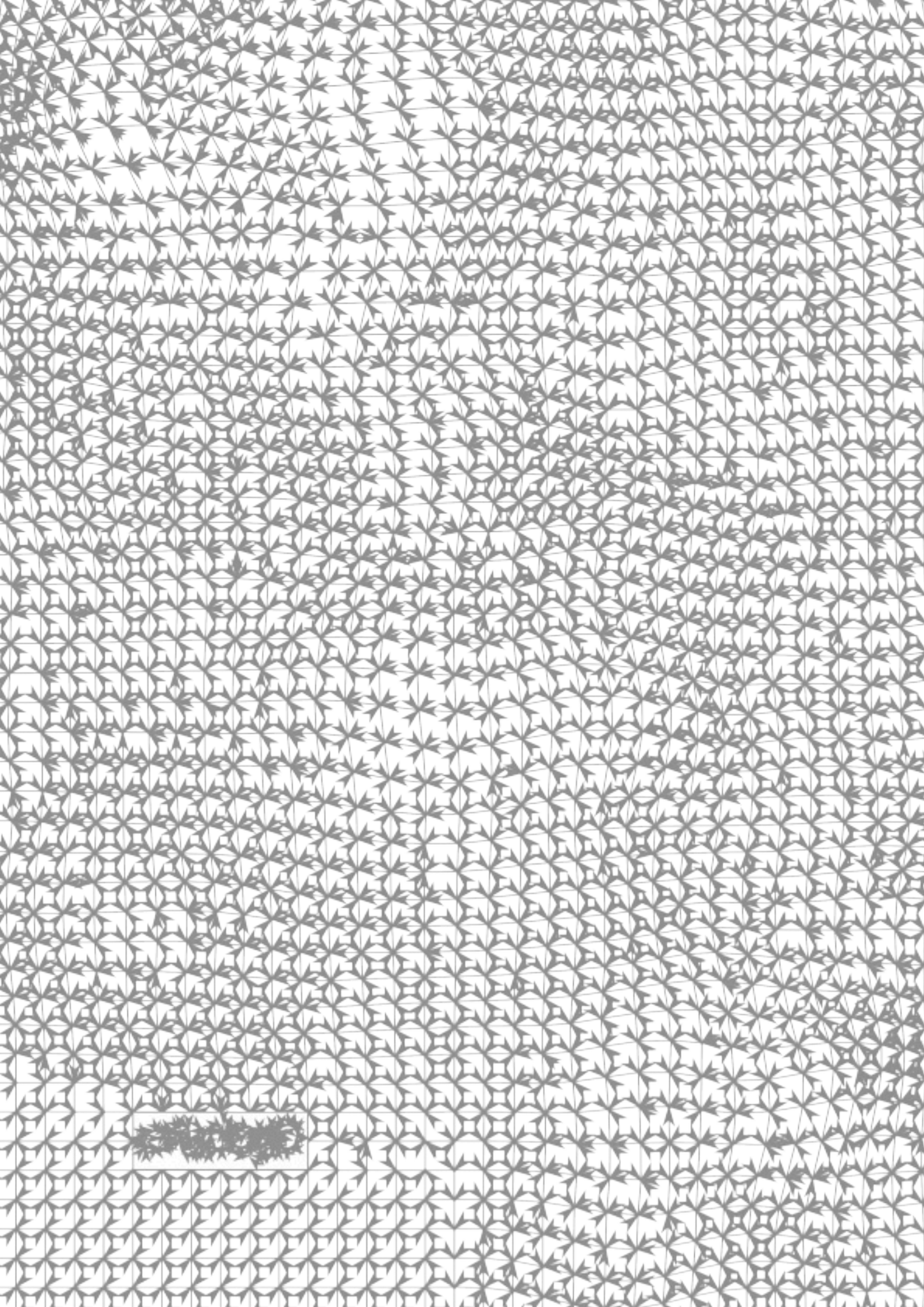
Index

Capturing capture

GINGER COONS

ginger coons is a member of the Libre Graphics Editorial team.

Informally, capture can be described as the fixation of information on a persistent medium; we can frame it as a way to register temporally ephemeral and/or spatially restricted objects or events. In this sense, capture becomes archival, and the scope of capturing is widening: as photography is now ubiquitous in the hands of everyone with a phone or cheap camera, we move to denser aspirations, with 3D scanning as the next milestone.

As human actions move to the digital medium, they become targets of capture. Computers acclimatized us to the notion of "error-free" storage. From occasional capture of our moments, we now continually capture more and more aspects of our lives, actions and trails by making backups, storing logs, updating timelines, archiving e-mail messages and syncing devices. Data storage is crystal clear, with a minimal margin for glitches or noise (save the occasional hardware failure). We can take snapshots of our own computer systems so that we can roll back undesired developments—capturing the moment is part of anyone's computing experience. Capturing their state and freezing it into a register of a specific moment in time, that we can go back to anytime—moments like carbonite.

It appears that the more we capture, the more information we realize we emit. The "quantified self" perspective amplifies this to its limit, by proposing the capture of all possible outputs, from conversation subjects to micro-behaviours in our bodies. But does this make us grow? But can yet more data enlighten us?

Contemporary artistic practices address that question. From the utilitarian purposes of the Internet Archive's Wayback Machine, which allows people to see previous versions of web pages, we can contrast the Printing Out the Internet project by Kenneth Goldsmith, which appropriately highlights the shortcomings of capture and archival in an almost absurd mission statement to capture the World Wide Web into an analog medium. Such a statement highlights the impermanence of the Internet, the medium which was supposed to become the ultimate archive by uniting the world's storage and computing power.

The still recent NSA surveillance scandal engraved the heavily political aspects of ubiquitous capture into mainstream awareness. It's tough to adjust to the mindset that many more actions are now subject to capture by third-parties: real-world visual or audio recording gave way to blanket logging of all possible activity; it's telling that Google has put the brakes on its mission to catalog the world's knowledge to focus on the monetization of captured habits. The act of capturing therefore embodies a strong, and ambivalent, aspect of morality, offering the potential to safeguard history, as well as the threat of deep surveillance.

The Capture issue plays with these topics, with the hope that the projects showcased and ideas articulated give you pause for reflection, in a world that's increasingly recording.

Outils Libres Alternatifs

NOTEBOOK

Outils Libres Alternatifs means "Alternative Libre Tools" in French.
This non-profit initiative gathers Sarah Garcin, Bachir Soussi Chiadmi
and Raphaël Bastide around a shared goal: to promote and broadcast
free/libre and open source tools for creativity. Workshops are
the perfect way to focus on a specific creative practice.

The first OLA workshop (OLA #0) took place in Paris, in May of 2015.
Each workshop is organised according to this protocol:

1. The team of Outils Libres Alternatifs chooses a date and a place to organise a 3-day workshop.

2. A creative practice is chosen, from illustration to publishing, from 3D editing to sound making or VJing. For OLA #0, this practice was "Publishing."

3. We contact a master, someone who has a technical and creative approach to the chosen practice, someone who wants to share this knowledge. OLA #0 invited Aurélie Delafon, sub-editor of *Le Tigre*, a beautiful French magazine laid out in Scribus.

4. The workshop is ready to start, with all kinds of creative participants with assorted skills, status or age, ready to experiment during 3 days. Speed talks are planned to allow everyone to share their own practices or hand crafted tools!
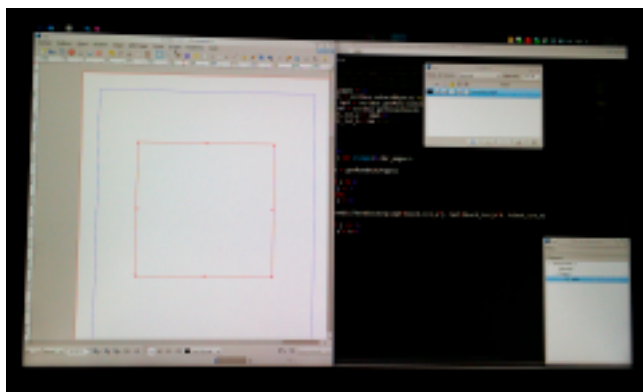
5. Three days of experimentations with Scribus, Python scripting, discovering SVG and libre typography in the spacious premises offered by La Générale.

6. Documents begin to appear, generative layouts and illustrations, articles, counterfeit newspaper, textual bug reports…

7. During the last hours of the workshop, a publication is printed and assembled, gathering all the participations. During an "apéritif" (an informal public presentation with drinks), the workshop and its subjects are introduced to the visitors, neighbours and guests.

8. Time to organise a new event in October. This time the practice will be about 3D editing, and a new master will be invited!

http://outilslibresalternatifs.org

Dry layout

MANUFACTURA INDEPENDENTE

Ana Isabel Carvalho and Ricardo Lafuente
make up Manufactura Independente,
a design research studio based
in Porto, Portugal. Their design practice
orbits around the principles
of Free Software and Free Culture.

http://manufacturaindependente.org

Layout programs, in general, lack automation and modularity. After another issue of the Libre Graphics magazine, where most of our time was spent in front of Scribus, we decided to address the topic. We'll focus on Scribus since it's the tool we mostly use for layout.

Design workflows contain a great deal of repetition, and layout suffers particularly from this. In layout work, mundane tasks—such as importing images, creating text boxes, importing text, applying text styles, setting up text flows, distributing content—take up significant time and offer little space for creativity or enjoyment. It is a particularly mechanical work that we wish could be automated whenever possible.

In coding circles, there's a common best principle that's usually summarized as an acronym, DRY—meaning "Don't Repeat Yourself." It refers precisely to the best practice of automating common tasks and minimising repetition or redundancies in code. It makes sense that repetitive tasks are scripted or automated to make layout a less robotic experience.

Besides this, layout software presents a bubble-like way to work. Scribus follows an application-centric model, where most elements are managed from inside the tool. When text is imported, it is copied over to the Scribus document. This means that, from that point on, any change to the text needs to be made within the layout software—changing the original text file will have no effect. Contrast this to the way DTP software treats images, which are usually linked to their source files. The approach of linking text to its source file is a feature we've been missing for years in layout editors.

By now we haveve pretty much covered the two things we miss the most in our layout tool—automation and modularity—and we can dive into what we see as the ideal scenario: having the ability to map text boxes to text files, written in Markdown or even HTML.

This would go beyond importing text as we have it now: the text box would be linked to a text file, and reflect any file changes. Each time the file is modified, the layout would update itself to reflect the new state (including semantic notations from the editor such as bold, italic, quotes, titles and subtitles).

With this magazine as a use case, it would mean any last edits, additions and fixes to the text could flow directly into the layout. Since there are always last minute edits to be

made, which happen in tight timeframes where work is rushed, it's common that the final corrections are done in the layout document and not in the source text files. The effect is that the source files are not in sync with the layout document, making it much more difficult to re-use the magazine content without needing to open Scribus to get the latest version of the text.

If there is a subsequent change in the text file, we have to be attentive to the commit logs to manually check the changes and repeat them on the Scribus document. A proper way to link text files could allow external contributors to participate in the editing process without the bottleneck of firing up Scribus to integrate edits into the layout document.

All of this can probably sound like a nitpick, but the underlying issue is one of the core dilemmas of digital publishing: the relationship between content and styling. In DTP software, content and styling are tightly integrated into each other.

Scribus does offer a powerful internal text editor, but it struggles to represent paragraph and character styles, and thus is not a proper replacement for a link-based model. (While Scribus does have a HTML importer, what it does is to copy the HTML file's content to a text box—not exactly the functionality we outlined. Nevertheless, it's a rather handy feature if preserving markup and styling is important.)

An opposite paradigm has been developed for the web: HTML and CSS embody the paradigm of total separation.

In its early days, HTML would be used to store not only content, but also all the graphical rules to render that content (such as fonts, colors or visual spacing). CSS was brought in to deal with the inevitable confusion, and effectively propose a separation between the actual content and the directives it should be styled with.

There is also a sense of modularity in how content is written and rendered, with the editing tool (text editor or web IDE) being distinct from the viewing tool (the browser). The content/style distinction became important and useful: editors can focus on the content, whereas designers can work on the style, all without stepping on each others' toes.

This workflow becomes tricky to implement in a single-application model. With DTP applications, the text is contained inside the layout file, as well as the style settings.

Editors are forced to edit text directly inside the DTP application, even though a simple text editor would suffice for most edits. Even designers might not need to launch the full application to make a small color tweak and export a PDF file (another feature request: export Scribus files to PDF from the command line).

In an ideal scenario, the layout file would be laid out in a format that could be opened by other, smaller applications. However, the complexity of DTP makes for file formats that are complex, hard to understand and often opaque.

This is unfortunate, since it would make sense that DTP would also benefit from the Unix way of doing things: having small tools that can be chained together.

We'd like to do with Scribus what we already do with ImageMagick: use and make small tools for jobs that can be easily automated.

The TeX model also springs to mind. It avoids the "huge GUI-centric application" problem by doing all its rendering on-demand, but has a steep learning curve in order to get to grips with its automated layout settings and the ways to fine-tune or hack them. This balance highlights the challenges behind creating a powerful, DRY-oriented layout tool that is also approachable by non-specialists, a holy grail of sorts.

However, there's promising developments in tools that propose alternative models for desktop publishing.

- **Flat/Even** by Juraj Sukop proposes a code-based solution to create layouts, using Python scripts that generate PDF files. The code aspect immediately enables generative experiments with layout.
  http://xxyxyz.org/flat

- On another front, **Html2Print** aims to create complete print layouts using only HTML, CSS, JavaScript and a web browser.
  http://osp.kitchen/tools/html2print

All that said, Scribus is still our main tool for everything layout-related: it produces flawless PDF files, takes good care of complex text and handles dozens of pages without hitches. Version 1.5 is just out, and we're rooting for the Scribus team to keep rocking with a fantastic tool.

New Releases

Antimony 0.8.0

A novel approach to 3D prototyping using a node-based interface, Antimony just put out a new version with a good set of improvements.

http://www.mattkeeter.com/projects/antimony/3

Chainmail Bikini: The Anthology of Women Gamers

An anthology of comics by and for female gamers. An excellent compilation of stories and viewpoints on games and gender, prominently featuring our favourite typeface, PropCourier Sans.
The PDF version is available, with the print version out in September 2015.

http://chainmail-bikini.com

Computed Layout

An ongoing collection of the next generation of layout paradigms, focusing on CSS Print, generative layouts and alternative publishing tools.

http://computedlayout.tumblr.com

Freeze+Press

A new F/LOSS-driven small press, dedicated to the explorative use of software for printmaking.

http://freeze.sh

HTML 2 print

A new practical proposal for using web technologies to create print layouts.

http://osp.kitchen/tools/html2print

Hybrid Lecture Player

A web platform that turns lecture documentation into multi-format publications.

https://mcluhan.consortium.io/

Conversations

A collection of interviews and conversations focusing on the more diffuse aspects of free software and free culture. Edited by Femke Snelting in collaboration with Christoph Haag.

http://conversations.tools

MediaGoblin 0.8.0

Dubbed "A Gallery of Fine Creatures," this new release of MediaGoblin brings our favourite media gallery closer to full federation of content, Python 3 support and Gstreamer improvements.

http://mediagoblin.org

RAM: Retales Analógicos de Medialab

The first edition of Medialab-Prado's fanzine is out, in an effort to capture the many facets of digital culture that Medialab-Prado is known to nurture and grow, this time under the subject of Habitability.

http://medialab-prado.es/article/fanzine

Synfig Studio 1.0

The up-and-coming F/LOSS animation editor marks its 1.0 release with a respectable list of improvements and fixes.

http://synfig.org

Valentina 0.3.3

The F/LOSS pattern drafting software for fashion design pushes on with a new version.

http://valentinaproject.bitbucket.org

Evasive Maneuvers

ANTONIO ROBERTS

Antonio Roberts is a digital visual artist
based in Birmingham, UK.
He produces artwork that takes
inspiration from glitches in software
and hardware, errors and the unexpected.
http://hellocatfood.com

We are obsessed with capture. On an aesthetic level we have been attempting to capture aesthetic qualities of things for thousands of years through drawing, painting and, relatively recently, photography. Advances in digital technology now allow to capture more than just aesthetic qualities of a thing. Now we can measure things, analyse them, and make decisions based on statistics and quantifiable data as opposed to qualitative personal opinions. Through this we have gained an incredible insight into the world around us, and can study everything from weather patterns, genetics of species and so much more. This gigantic planet and beyond now seems so much more comprehensible, now that we can understand it in terms of numbers and patterns.

Developments in digital technology have turned the focus increasingly onto the individuals. We want to understand not just how an environment evolves, but also the people that move through it. We want to know how they interact with it, why they do so, what their intent might be, what they might do next, what their emotions are and study little incidental quirks that could reveal more than a person intended. Like weather and climate data before it, the hope is that by collecting enough data about individuals we can begin to understand them more, and make predictions about them.

Although it has the potential to be useful, collecting this amount of data can have dangerous implications. Cctv cameras became the ever watchful unblinking eyes that are found littered throughout cities. These have only been enhanced by movement detectors, facial recognition software and GPS data on our phones. Thanks to constant misuse and abuse of these systems and the data that they collect, we have become distrustful of those collecting data. Now we are, more than ever, looking for ways not to be analysed but anonymised.

Artists, too, have been utilising their skills to highlight the culture we live in, where our every move and keystroke is being captured. This artwork, whilst informative, is at times eerie in the amount of data it reveals.

In 2009 Kyle McDonald undertook the live Twitter-based project Keytweeter. For this project, McDonald would tweet every 140 characters he typed, with a few exceptions like passwords and e-mail addresses. He also removed, at the request of others, text that would reveal sensitive details about third parties. McDonald states how, in undertaking this project, he was more aware that others may be "listening in" on his conversations. McDonald himself was explicit and transparent about his project. The source code was released and he alerted everyone that he was tweeting his keystrokes in all of his communications. This transparency allowed him to carry out the project, and his day-to-day life.
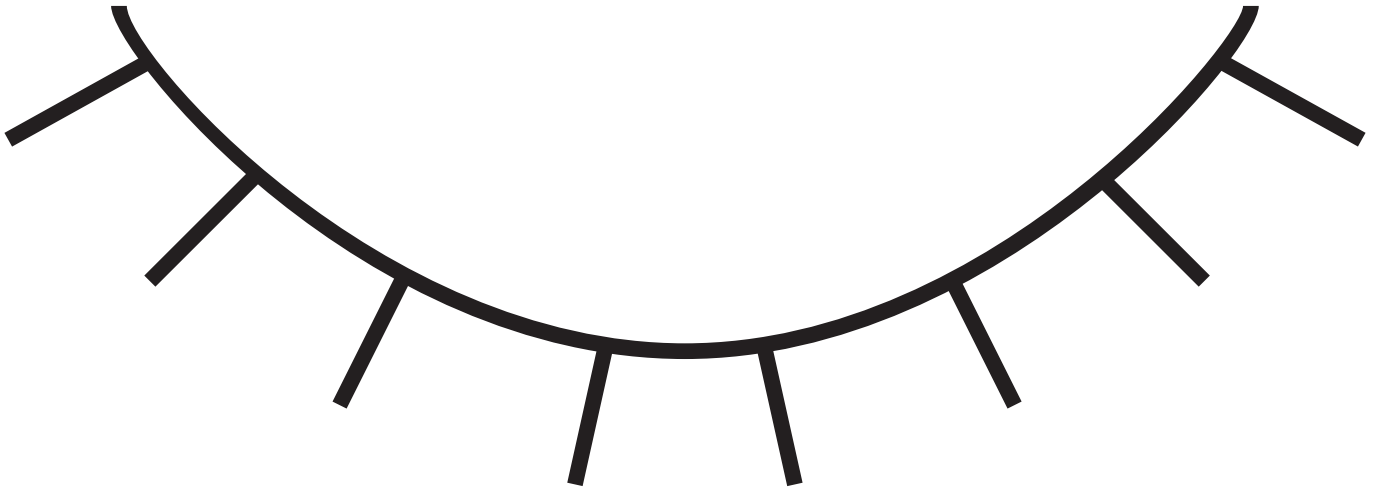
His other projects, such as Scrapscreen, Important Things, and other graphical projects like IOGraphica and Graffiti Markup Language, which similarly capture and display graphically potentially sensitive data, all avoid crossing the line into being invasive through this same level of transparency. Users are aware of the data that is being collected, how it's being collected and can decided for themselves what is done with it.

All of these projects could be deemed invasive, but what they are doing is what automated software has been doing covertly every day. Bots scan your e-mails, tweets, status updates, images and videos for information that is used to identify you. They claim not to do this to spy, but to help bring you more relevant information.

It is now not just text that is being mined for data. Facial, gesture and emotion recognition software is being used in a variety of invasive ways. Features like Facebook's auto tagging feature can be useful for identifying friends and family in photos, but become dangerous as we are in the dark about how this data is being used. Similarly, advertisers are employing the use of emotion tracking software such as Affdex to detect emotional responses to advertisements.

The aforementioned technologies themselves are not malicious. They may well have been developed with an intention to further study the world, but they become malicious when we are ignorant about how they are being used and how the data they capture is being employed. Only when there is transparency, as seen in Keytweeter et al, can we become comfortable with data being mined.

How, then, can we escape this data capture?

Alongside the very technically complex open solutions from the likes of the Tor Project and Blackphone, artists and designers have been addressing the issue of avoiding being captured in creative ways.

Sang Mun's ZXX typeface from 2013 in one such approach that obscures text from being read by optical character recognition processes. By adding noise to glyphs—be it in the form of haphazardly placed pixels and shapes or overlays of other glyphs—the hope is that text would become indecipherable to computers, much like the aim of CAPTCHAS.

Adam Harvey's CV Dazzle explores how fashion can be used as camouflage from face-detection technology. Instead of opting for camouflaging the whole face by wearing a mask, Harvey's solution selectively obscures parts of a face using removable tattoos and face-obscuring hair attachments so that computer vision software becomes confused and eventually fails.

Matthew Plummer-Fernandez created a system in 2014 for encrypting and decrypting files made for 3D printing. When a user wishes to distribute an STL file, they can use his software to apply an algorithm that encrypts the file by randomising the vertex positions, which makes the 3D model appear to be an abstract shape. When a user wants to decrypt it, they need to apply the same algorithm, or the model becomes visually corrupted.

A common theme in these approaches to escaping computer vision is the addition of noise. Current computer vision software is only able to capture data by analysing media and finding common patterns. The addition of noise disrupts this software, which allows the obfuscated element to go undetected.

Whilst these approaches may have a reliable level of effectiveness now, it is only a matter of time before computer vision software adapts to recognise glitched and corrupted files. We will begin to enter an arms race, leading to an ever increasing amount of noise being added to try and hide our messages.

Tonight we're making web sites like it's 1999

ERIC SCHRIJVER

Eric Schrijver (Amsterdam, 1984) is
a graphic designer and a
performance artist. He is inspired
by programming culture. Eric
teaches Design for new media at
the Royal Academy of Art
in The Hague, and is a member
of the design collective Open
Source Publishing.

http://ericschrijver.nl

Visiting *ttypp.nl/arch* is a time machine. This is where the webpages of TYP/Typografisch papier are archived. TYP was a forward-thinking magazine initiated by Dutch graphic designer Max Kisman, published on paper, on floppy disks and online. Surfing through the online editions, one is transported 20 years back, to a period when the World Wide Web had only just begun to arrive on people's computers. The enthusiasm displayed by designers looking for ways to exploit the new medium is contagious. On the transient space of the internet, it seems like a small miracle that this cultural moment is still accessible.

Surfing around web sites from the nineties on the Internet Archive, one notices that quite a few do not display anymore, among which many sites that use technologies like Shockwave or Flash. web sites using HTML tags, the native tongue of the browser, have generally aged much better. This is no accident: backwards compatibility has always been important for web browser vendors. Since browser vendors have so little control over the markup people write, browsers are very forgiving about the markup. Within HTML5, this tradition has been codified as the parsing of malformed or non-conforming HTML has been standardised.[1]

For many of the first web sites, HTML was not just the format in which they were delivered—it was the format in which they came about. The first web-sites were "hand-crafted HTML": created as a series of HTML pages, with occasional updates (the person designing the site might then charge for each update!). This did not mean coding was necessary: tools like Adobe Dreamweaver provided a visual view and a code view. The democratisation of Content Management Systems (CMS) like WordPress and Joomla changed the equation. In these systems, a general design is encoded into a template, and the contents for individual pages are stored in a database that is easily editable by the user. For clients, this saves time and money. The downside is that a CMS requires shoehorning every page into templates: these early HTML pages offered much more freedom in this respect, as potentially every page could be modified and changed according to the designer's whims.

This suggests that HTML has additional properties which not only make it the right format for delivering and archiving web sites: it looks like HTML files also provide a very powerful authoring format. The logic of CMS's (and indeed, the intended logic of CSS) is to pull form and content apart. Yet traditionally, the intelligence of designers has resided in creating links between form and content.

1. Some misplaced puritanism has caused HTML standards writers and browser vendors to remove the blink tag. This might have to do with a narrative in which Geocities-style, amateur driven web design had created a chaos from which we all had to be saved by standards loving professionals—in this sense, the blink tag becomes a *pars pro toto* for an approach to web design built on Comic Sans and MIDI files, that the "professional" web users suspect they can kill off by sacrificing blink. But it stands as a curious omission in what has basically been a technology that has been remarkably caring for its past.

Moving beyond the template, and allowing authors and designers to modify the design of each specific page, is what working in separate HTML files enables.

If such an approach were to be viable today, new tools will have to be developed. With tools like Dreamweaver fallen out of grace, it looks like the only tool we have now to edit HTML files is the code editor. Yet the popularity of database driven CMS's stems from the fact that they can provide different interfaces for the different people involved in creating a web site. A developer might need a specific view, an editor might require a distinct angle, as might a designer—even if one person combines these roles. New tools will have to be able to provide different views for editing the HTML document. Instead of generating the HTML, as conventional CMS's do, these tools should work *on* the files.

Having HTML files as the source upon which tools can be used also has implications for interoperability. This is a pressing issue in digital publishing. Currently, many parties all make their own database-driven solution. The design will be encoded in a custom template format. The text will be encoded in a custom markup format — often based on Markdown, but never exactly the same. This makes it very hard for third-party service providers to interact with these systems, which makes it harder for an economy to form around digital publishing.

In recent years there has been a return to creating solutions built on static HTML files. This is because hosting HTML files is easier, cheaper and more secure than hosting a dynamic system. Since the web site does not need to store information in a database, the "attack surface" of a web site hosting static files is much smaller. The web site does not need to expose an editing interface that can be hacked. Also, a dynamic system will be have to kept up to date to fix known security holes. No such maintenance is needed for the HTML files: and because they do not use any specific capacities of the server, the cheapest hosting solution will generally suffice.

These enviable properties have led to a proliferation of scripts designed to create such HTML files: static site generators. At the time of writing, some 398 were listed on the *staticsitegenerators.net* registry. These static site generators suffer some of the same drawbacks as conventional CMS's: they often work with a template through which all content has to be pushed. The advantage, then, is that such tools are always well equipped to generate indexes. The question of how to syndicate, index and provide navigation for a collection of static HTML files—that is the second challenge for HTML as an authoring format.

+rwx data and pipes: Better design for otherwise non-writable infrastructure and unreadable data

BIRGIT BACHLER AND WALTER LANGELAAR

As a result of recent endeavors in working with data as creative material, the three projects discussed in this piece approach processes of capturing data and release it in new forms through varying methodology and experiment, within different modes of subjectivity. The basic question we're asking is whether data-related platforms or practices exclusively constitute us as subjects, or whether we can claim back some agency in this mode of production by means of experimentation and subversion within its confines.
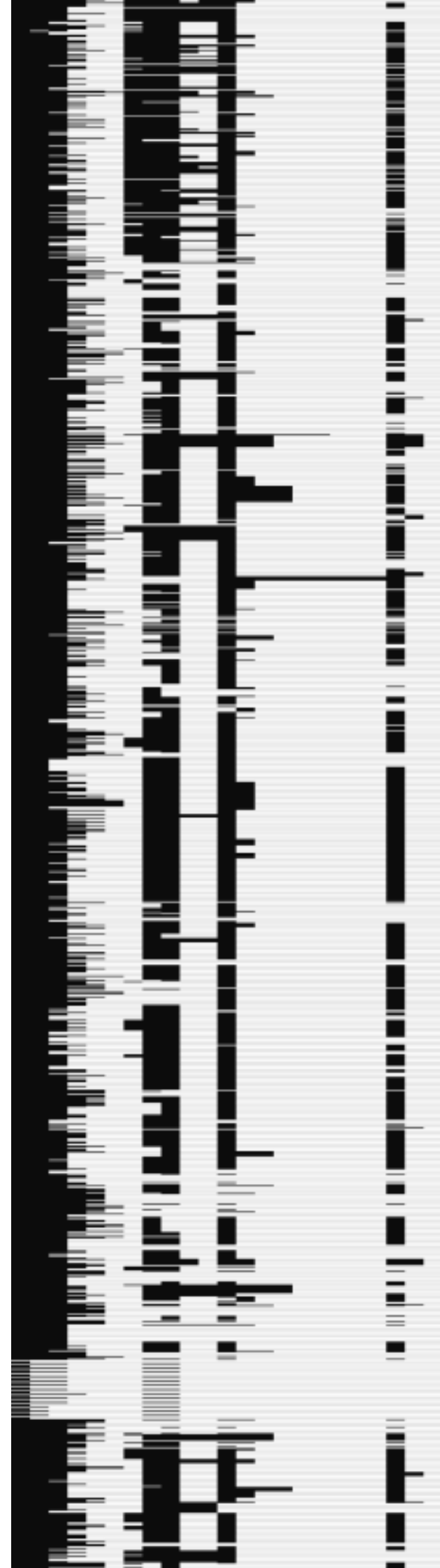
They make money from your data. Why shouldn't you?

Commodify.Us is a web application that allows people to visualize and license their Facebook data directly to marketers. Facebook continuously captures data from their users, going well beyond content uploaded intentionally like status updates, comments and images; it also continually stores location data, unpublished comments, deleted friends and machine-accumulated ad words. The "Download your information" function of Facebook also includes this invisible data. After exporting it from Facebook and uploading it to *Commodify.Us*, users can choose to license their anonymized data under private, fair use, or commercial licenses and effectively cut out the middle man. The project intends to correct an imbalance of power in the use of personal information by enabling users to enter a market for their data. As authors and owners, users can become active participants in new streams of creative output.

Appropriating the language and style of start-up culture, *Commodify.Us* poses as a fake Internet start-up employing the aesthetics and infrastructural backbone of a typical contemporary tech company. The fact that it is actually an activist, artist-run initiative is unclear at a first glance. By re-releasing social media datasets in the realm of an alternative model for private information marketing, we aimed to liberate the user from the datafied subjectivity generated by Facebook's algorithms and UI/UX design.

A significant value offered by the Commodify.Us platform is the power to manage our own data. The simple act of downloading our own data from Facebook, and then uploading it to Commodify.Us

supports us to rethink what all this information is. What once was just abstract data suddenly becomes material that we can manipulate. Alongside this realization arrives the understanding that this material was made by our interactions with all these platforms, and that other people are spying on us and making money out of it all.[1]

In the wake of the project's launch and after its first batch of users came online, we ran a series of workshops and presentations to further our research into the mass capture and commodification of user data. Participants were given the possibility to experiment with personal and/or open Facebook or Google+ profile datasets, to further quantify and visualise social media data and essentially establish a currency conversion of their online presence, potentially viable for further marketing or for use as creative material and critical reflection. Moving away from initial paranoia into more empowered modes of interaction with either personal data or open sets, the surge in workshop participants' awareness surrounding data as a means for production of subjectivity resonates with Fuller and Goffey's discussion of data as a "grey medium":

Civil libertarians only get half the story right in their well-meant concern with the vast extension in scope of data gathering and mining. From some points of view, the growing volume of personal data available does indeed look likely to threaten a totalitarian encroachment on basic human rights. But it should also be considered as an important element in optimizing the functioning

of market processes. In this respect, consideration of data mining would lead to an exploration of the rather less well-understood use of patterns and pattern recognition as an element in the process of modeling and production of subjectivity, as a component in the selection and extraction of new forms of the supply of subjectivity for the demand of markets.[2]

Put your Big Data on my...

In order to counter some of the above mentioned methods and tactics employed to use data-modeling in the production of marketable subjects, we can start thinking about new ways of shifting perspectives in this manufacturing process.

Big Data Wall Fillers is an attempt to make abstract datasets more accessible for domestic use, by processing them into decorative wall paper and screen savers. The project plays with the domestication of bulky datasets by re-interpreting the raw data for wall paper design samples.

The Open Data movement has promised wide data accessibility to the masses: Governments, telcos, companies of any sort are opening their digital inventories in the hope for contributing to a greater good. In *Big Data Wall Fillers*, a side-effect or symptom arises that allows us to question the subjectivity of Open Data datasets and file formats. If presented in a visually appealing form in a gallery setting, viewers feel tempted to interpret the visuals and discover relations to the original datasets, engage in discussions about certain figures with fellow visitors.

Presenting dry datasets in a seemingly familiar format lowers the threshold of lay users to engage with complex data, and encourages them to add their own opinions and interpretations.

Wall paper designs are usually regarded in a solely decorative context, as those patterns do not usually carry any deeper cultural meaning.
This work attempts to add a layer of meaning to the designs, leaving a vivid trace of the often orphaned spreadsheets on the walls in our homes.

Surrealist WiFi

While the above projects make use of already captured data, *SSID_Exquis* intervenes right at the moment of capture itself and alters the access modes of the usually read-only environment in the commonly known WiFi part of the 802.11 radio spectrum. Positioned as a performative and participatory work in public contexts, the project blurb reads as follows:

SSID_Exquis is a gamified exercise in collaborative poetry, manifested in WiFi. Its participants can publicly broadcast wireless network names, or SSIDS,

by contributing to a collectively assembled list. From this list, a series of wireless LAN beacon frames is generated and transmitted periodically to announce their presence in the surrounding 802.11 radio spectrum. The resulting flood of publicly accessible wireless networks is logged in a continuous fashion, constituting timestamped cadavres exquis.

Technically, the project functions as a captive portal which presents its users with a web page containing info about the work and some HTML form fields allowing for the contribution functionality. Whereas a captive portal would normally be used to constrain a user's movement on the network (or in the case of malicious use; capturing sensitive data from an unsuspecting user), in *SSID_Exquis* the aim is to make a WiFi radio spectrum writable by allowing each participant to have their say in the form of beacon frames.
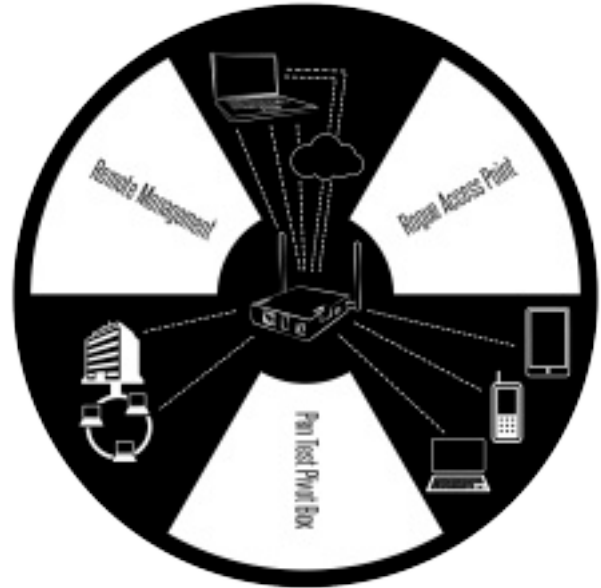
From an anthropological perspective, *SSID_Exquis* builds upon the popularization of niche-tech subculture surrounding topics related to "hacking", penetration testing and wireless network auditing—a post-Snowden collage of sorts. The project is structured around an "infusion" built on the so-called WiFi Pineapple MarkV device, a hardware and software platform which has a dedicated following via the HAK5 Youtube channel and forums. In essence, the platform is a cleverly put together mashup of very hackable WiFi radio chips, and a browser-based interface which allows for in-depth modification of its configurations. Ironically, *SSID_Exquis* subverts an earlier project built on this platform: dubbed Occupineapple, it was originally intended for protesters trying to get their word out via a sequence of broadcasted SSIDs.

These recent projects juxtapose the datafication of subjects with the subjectifying nature of large scale data gathering and mining: the use and re-use, visualizing, re-mapping and reverse engineering of digital data are ways to understand the complex processes within contemporary media. Interface design generally simplifies complex processes and enhances usability of systems, but simultaneously separates the lay user from the media professional.

We aim to familiarize our audiences with complex datasets and software processes through a rather playful approach. Facebook users do contribute to a large social data set, but access to all data is limited by Facebook's website. Encouraging users to download their sets offers the possibility to interact with one's own data without the eyes of Facebook capturing every single move and click. Similarly, a visually appealing rendering of a large dataset makes room for interpretation and speculation and eases the objective character of large spreadsheets. To engage an audience, workshops invite participants to bring their own data sets and learn how to use software like Processing to render them into large format prints.

This previously unfamiliar interaction with data immediately opens up discussions on data and capture in a more creative and substantial manner. When we move beyond this stage of attaching our data to executable scripts on our own device, and also start making its conceptional infrastructure writable, we may recapture at least some of the means of production of our own digital subjectivities.

REFERENCES

1. Marc Garrett,
http://furtherfield.org/features/reviews/commodify-us-our-data-our-terms
2. Matthew Fuller and Andrew Goffey (2012),
Evil Media, MIT Press.

Powered by GNU

FONS

Stéphanie Vilayphiou

A recipe to make fonts
out of bitmap images.

A LITTLE BIT OF HISTORY

In 2011, Pierre Marchand made with his own hands,
in the laboratory of OSP, a creature called Fonzie.[1]
Besides being cool, Fonzie was able to make a typeface
out of scanned letters.
In his childhood, Fonzie asked for letters to behave
well and to organize themselves.
They had to not hold hands and ordered themselves
by the order of the Unicode.
The Univers Else[2] family was born.
In his teenage years, in 2013, Fonzie got more
and more experienced.
He could decipher letters from a scanned page of a book
by using OCR (Optical Character Recognition).
Fonzie could then read nice stories. But Fonzie is so
cool that he's going even further.
With all those a's or all those z's, he was drawing
an average shape of all those instances of letters.
Fonzie collaborated with ScanBot,[3] a coarse book
scanner, product of the Dr. Michael Korntheuer,
which could suck the pages with a vacuum cleaner to
read the next page.
But ScanBot couldn't make it through the winter,
and Fonzie kind of got stuck in his sweet sixteen and
didn't follow the new trends in the libraries of code.
Today, it is rare to see Fonzie around; nor Pierre
Marchand whose new lazy landscape is made in wænd.[4]

Fons is the spiritual son of Fonzie. He does not have
much ambition (like reading novels and such).
He just wants to learn the alphabet, to be able to
recognize the letters when we point at them.
He got GlyphTracer 1.3[5] of Jussi Pakkanen, as a master.

GlyphTracer shouts glyphs and we would point at the
letters on a scanned page.
When done, he autotraces in the command line limbo and
generates a nicely wrapped .sfd font.
GlyphTracer went a step ahead and now autotraces the
font with the FontForge toolbox so that we keep the
history of the scan with the glyphs.

But Fons knew this was not enough. He has a good eye!
If you feed him with an unripe scan, GlyphTracer would
give you marshmallows.
Fons then got trained by Pierre Huyghebaert, the bitmap
master, and took good note.
This is the recipe that you have in your screen.

TOOLS NEEDED

```
T   T                              \`.   T
|   T    .--------------.----------)\   |   T
|   |    |///////////////|----------[ ]  !  T|
    !    '--------------'          ) (   | !
                                   '-'     !
```

- GIMP
- AutoTrace
- FontForge with AutoTrace
- GlyphTracer (included with this package)

PREPARING THE IMAGE

Get a bitmap image of characters (crop unnecessary white parts for a faster process, you can use GIMP's automatic crop for that).

the father of Anna-Marie, was a
belonged to the technical hierarchy of

A scan in bitmap

three men without any provision for individual protection from the space

A scan in gray levels

A rasterized font: to typeset it, you can use the template
characters-table.svg in the /input/ folder.

Levels (have a white background, black characters but still shades of gray)

ithout any

Small boost

Scale up the image to something like 254% with the "Sinc (Lanczos 3)" algorithm. We choose on purpose a non-round number of scaling to break the bitmap patterns.

Sharpen with "Unsharp mask"

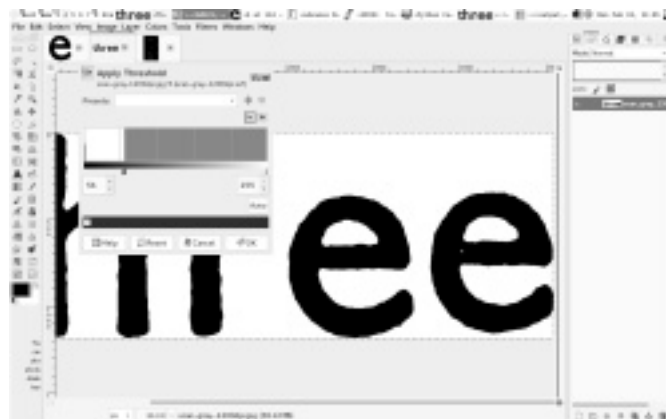Put the amount to the maximum and then search for the point where the radius feels right.
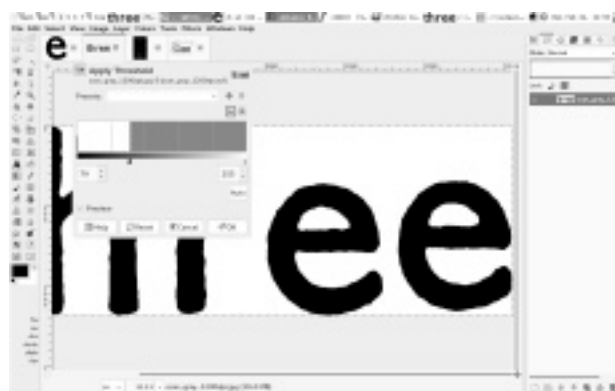

Radius too small   OK   Radius too big

Big boost

Scale up to 403%.

Threshold

Bad

OK

Baseline

When importing a bitmap, FontForge scales up/down the bitmap so that it fits into the glyph box. If we use GlyphTracer 1.4, this ends up with different scales of glyphs. In order to prevent this, we can put a black rectangle of the height of the maximum ascendant and maximum descendant.

Bitmap

- Change the color mode to Indexed mode
  with 1 black/white bit.
- Save as a `.bmp` file.

VECTORIZING WITH GLYPHTRACER

- To launch GlyphTracer, there are two possibilities:
  - from your file manager, double-click on the file
    `glyphtracer` inside the `glyphtracer-1.4` folder
  - from the terminal, go to the `glyphtracer-1.4`
    folder and run `./glyphtracer`. This will let you
    know more output in case of errors.
- Feed in the `.bmp` image in 1 bit (otherwise it will
  complain).
- For each given glyph, click on the letter you want
  to use.
- Change the characters subset in the bottom left
  dropdown menu to select more glyphs.

POST-PRODUCTION

Merging fonts

In case you want to complete an existing font, you can
use the `mergeFonts.py` script.

```
python mergeFonts.py fonte1.ufo fonte2.ufo ...
fonte17.ufo fonte-out.ufo
```
Metrics and kernings

A big part of type design is managing the white space
around the letters (metrics) and exceptions for
specific couples of letters (kerning).

- For metrics, we make an auto-spacing while generating
the `.sfd` file with GlyphTracer.
- For kerning, you can try the Kernagic[6] tool.

But if you want to get back metric and kerning data
from an existing font, you can use the
`mergeSpacing.py` script.

```
python mergeSpacing.py font.otf
original-font.otf spaced-font.otf
```

TROUBLESHOOTING

My image is too big to manipulate.

Split it into several images and generate several
`.sfd` files. Then you can merge the fonts with
the script `mergeFonts.py`.

From the terminal, I have the following error:

```
File './glyphtracer', line 415
    except Exception, e:
                      ^
SyntaxError: invalid syntax
```

This means your FontForge runs with Python3, whereas
the majority of installs still run with Python2.
I didn't find a convenient way yet to maintain both
versions at the same time. Let me know and I'll send
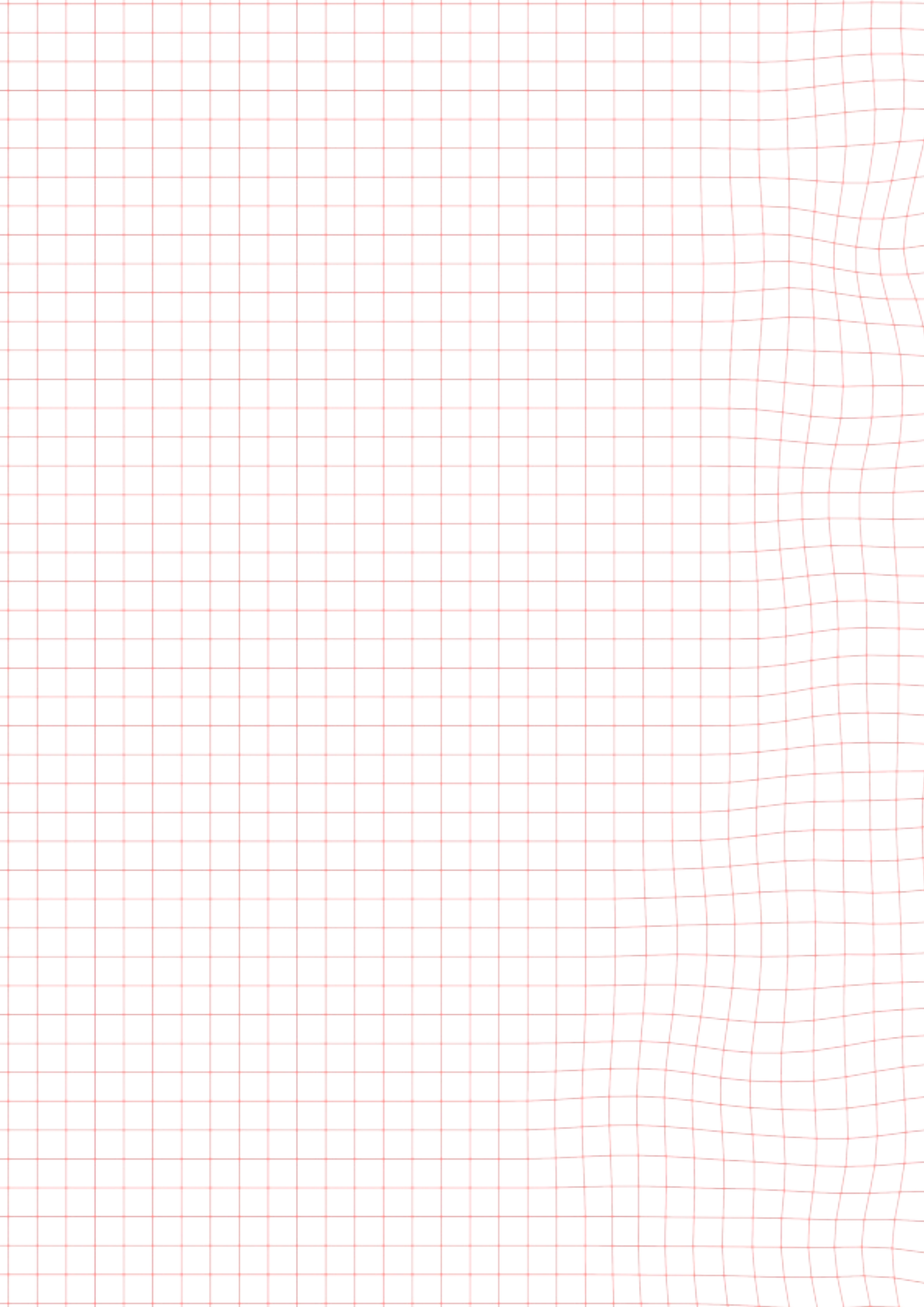you the version for Python3 which I use.

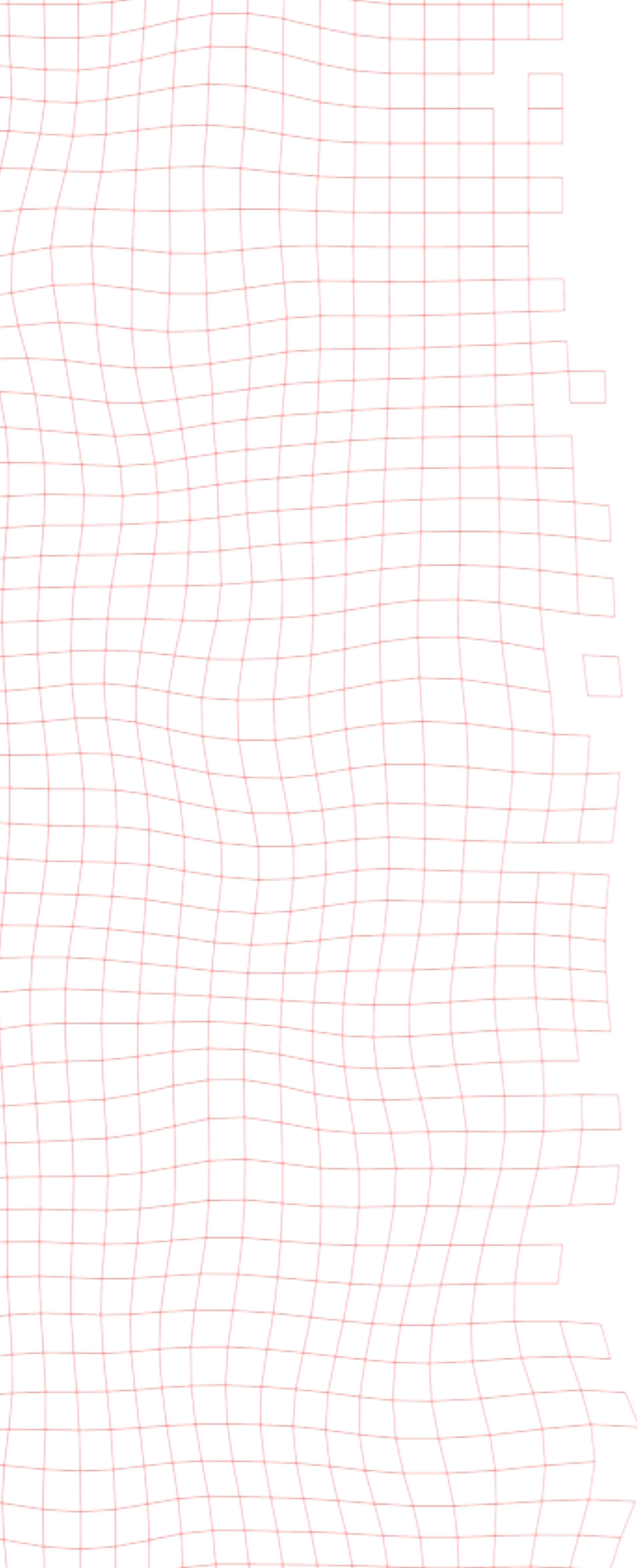**I have several shapes overlapping on the same glyph**

This is probably because you generated the font twice.
I'm afraid you'd have to close and restart GlyphTracer.

http://osp.kitchen/tools/fons

REFERENCES

1. Fonzie: http://git.constantvzw.org/?p=osp.tools.fonzie.git
2. Univers Else: http://osp.kitchen/foundry/universelse
3. ScanBot: https://hackerspace.be/ScanBot
4. Wænd: http://waend.com
5. GlyphTracer 1.3: https://launchpad.net/glyphtracer
6. Kernagic: https://github.com/hodefoting/kernagic

Showcase

Active Archives

SCANDINAVIAN INSTITUTE
FOR COMPUTATIONAL VANDALISM

An image is more than the sum of its pixels.

In the context of the traditional art school, we are taught to distrust the "effects" of photo editing software. Why use a digital simulation when you can work with the "true materials" of paint? What are the "true materials" of software, and hadn't we better use those when considering what painting means on a digital canvas? How can digital tools embrace the actual material of the algorithmic rather than merely simulating the analog?

The more we investigated "computer vision" techniques, however, the more we realized computer scientists are using the same techniques, and even employing an approach where techniques are composed almost as the cutups of a visual collage. No time to compute an actual Laplacian of gaussian? No problem, the textbooks offer, just gaussian blur at multiples of the square root of 2 and subtract the results to get a pretty good approximation.

A scanned image is more than a matrix of color values destined only to be displayed as pixels.

Viewed through the lens of algorithms, an image is multiple layers of potential interpretations. Each layer tells different story, revealing some aspects, obscuring others. Algorithmic glitches are revealing: is this a letter, or the edge of a roof tile? What are the visual features of text when treated again as an image? The layers also speak to each other; how do SIFT features related to edges, and edges to the automatic detection of text in an OCR software?

http://sicv.activearchives.org

↑ Contours: shows the contours detected on each image. In addition to tracing the edges, the algorithm connects the lines into a series of distinct segments represented by a different color.

↑↗ Lexicality and Texture: These layers are produced using Tesseract, a software for Optical Character Recognition (OCR). An OCR program operates at different levels of granularity. It can detect lines, words, symbols.
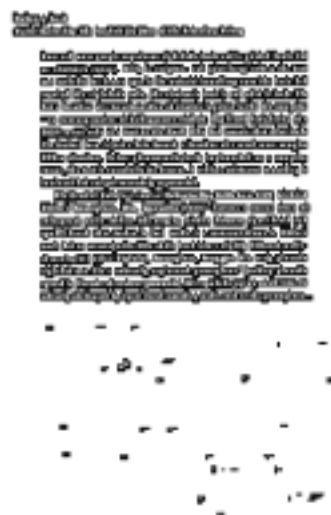
Lexicality shows the words detected in an image, while Texture shows the symbols detected. Texture is configured to be rather tolerant in its understanding of what a character is. It therefore tends to see characters in very unexpected places.
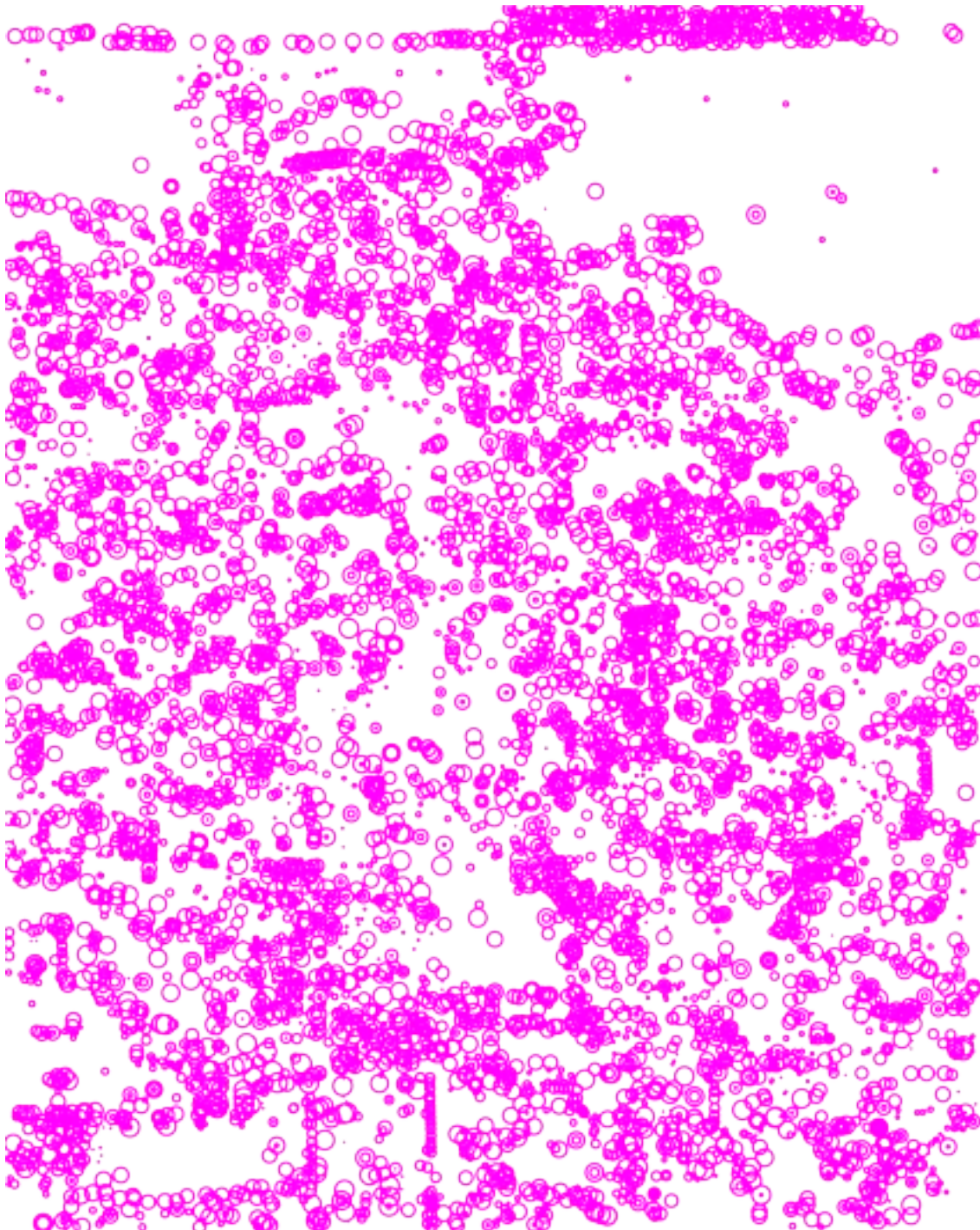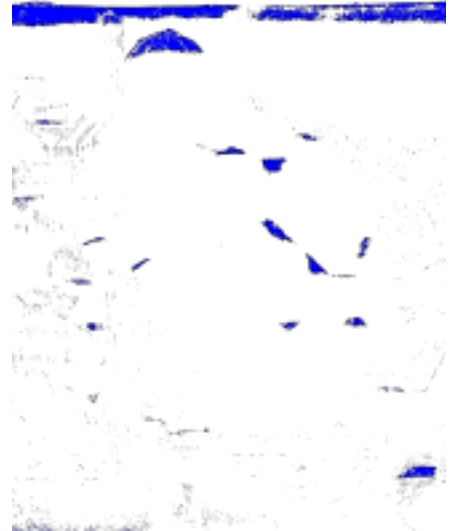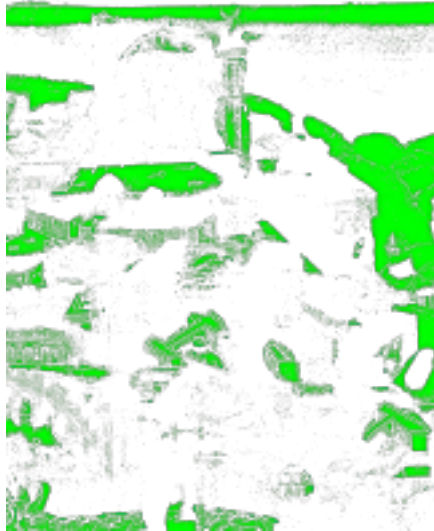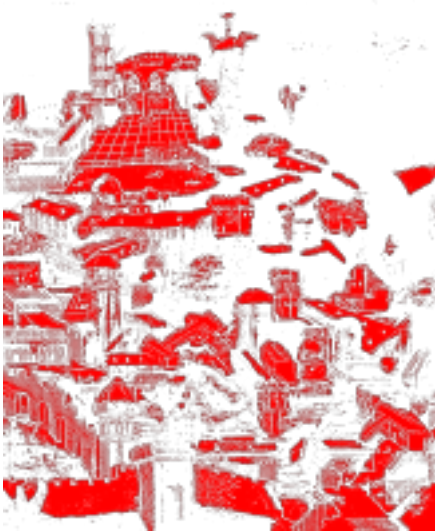
↑ Red, green, blue:
What is significant color
information? Contrary to
human intuition, for a
computer, a white image
is an image saturated
with red, blue and green.
To find the images that
look the most blueish,
reddish or greenish,
we counted only the
color values that were
superior to the others
by a certain threshold.

← Scale-invariant feature
transform (SIFT):
SIFT features are
"interesting" points of an
image that can be
extracted to provide its
"feature description." This
description can then be
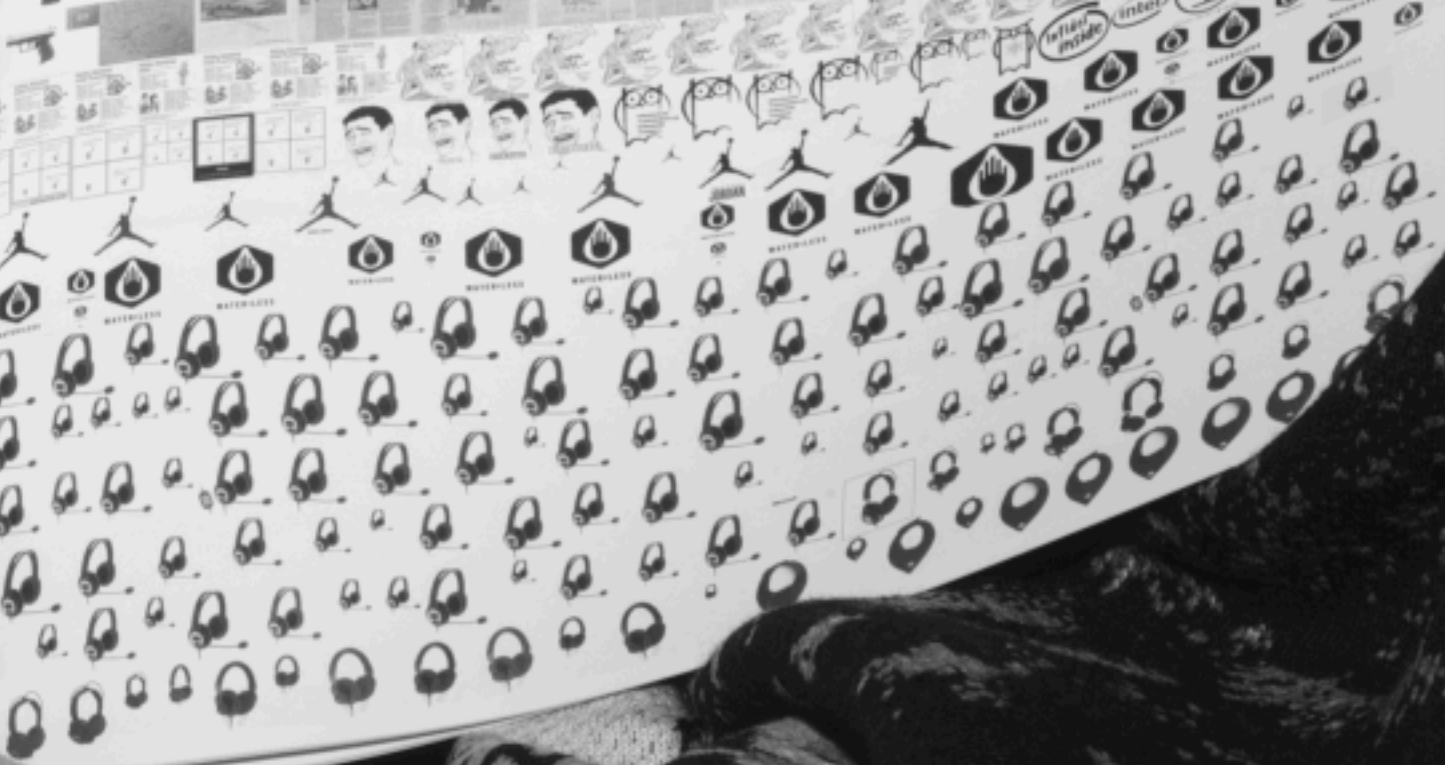used to identify (parts of)
an image even when
rotated or changed.
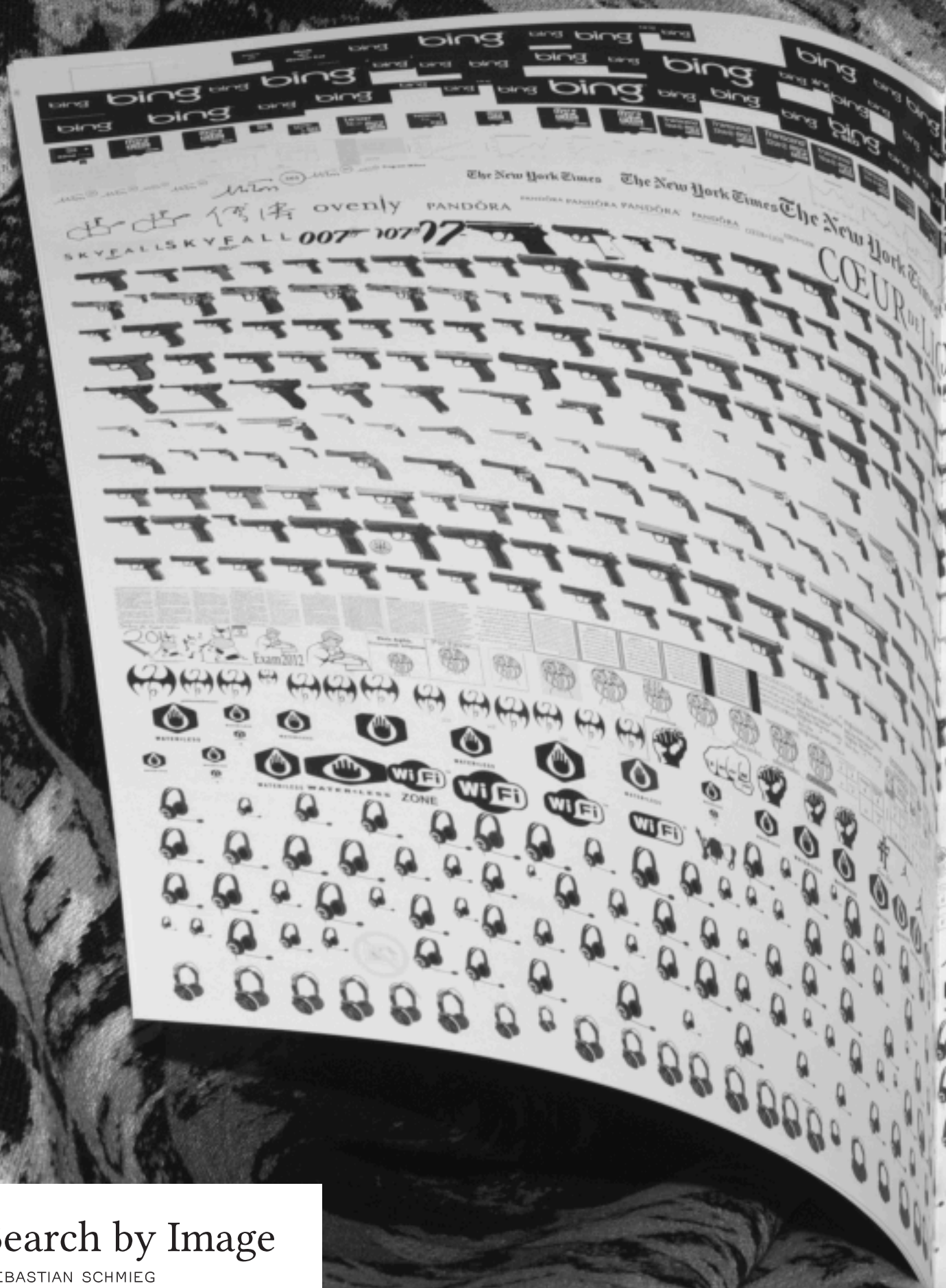
Search by Image

SEBASTIAN SCHMIEG
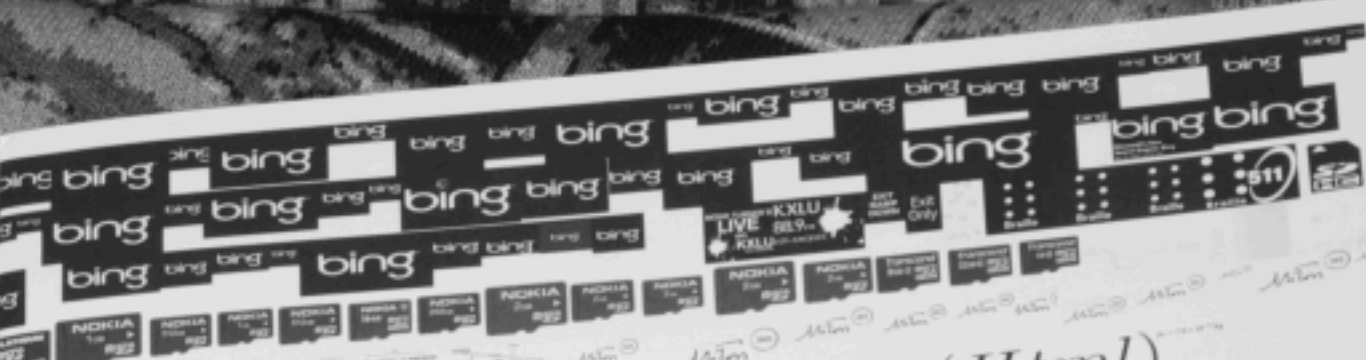
Search by Image is a series
of algorithmic and experimental
videos analyzing Google's image
search function of the same name.

Search by Image (Recursively,
Transparent PNG, #1) begins
with an empty image.
This image—a transparent
PNG—served as the starting point
for an image search whose result
acted as the basis of yet
another search. This recursive
process was repeated 2951 times
and then compiled into a video.

The process was made physical
through a series of 3 Artist
Blankets and one artist book
(edition of 250) developed
with Thomas Spallek
and Florian Kuhlmann, using
the same method to gather
and montage image sequences.

http://sebastianschmieg.com/searchbyimage
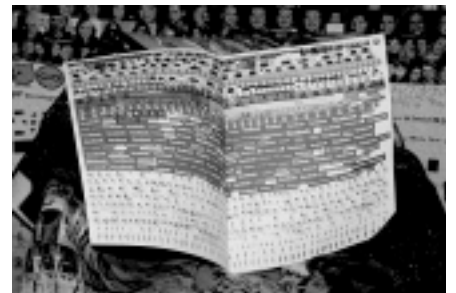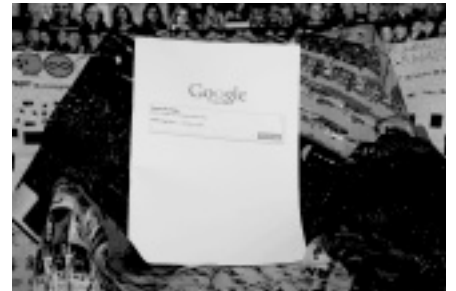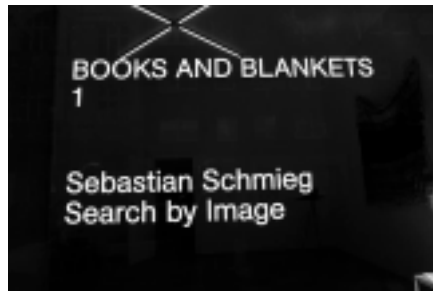
Printing Out The Internet

KENNETH GOLDSMITH

Printing out the internet [edit]

From 26 July to 31 August 2013, Goldsmith curated a conceptual art project called *Printing Out The Internet* in collaboration with LABOR and UbuWeb, that invited the public to print and send pages from the Internet to an art gallery in Mexico City, with the intention to literally print out the entire Internet.[15]

Goldsmith dedicated the exhibition to Aaron Swartz, an Internet activist who committed suicide while facing federal charges of illegally downloading and disseminating millions of files from the digital library JSTOR.[16] As Goldsmith said in an interview, "The amount of what he liberated was enormous — we can't begin to understand the magnitude of his action until we begin to materialize and actualize it. This project tries to bring that point home."[17] By the end of the project, Goldsmith had accumulated over 10 tonnes of paper from more than 20,000 contributors.

Although Goldsmith said that all the paper would be recycled at the end of the project, bloggers and journalists criticized the project for its alleged environmental impact. Goldsmith and his supporters, however, argued that the conversation generated by the piece becomes just as important as the work itself. As one art critic wrote, "Perhaps we should see Goldsmith's project not as one of triviality, spectacle, or waste, but rather as a vital (even if temporary) documentation and as a form of protest to keep the internet free, in so much that it is. Why print out the internet? Because we can, for now, and because maybe we should."[18]

http://printingtheinternet.tumblr.com

Bring a thing!

ROBERT M OCHSHORN

"Bring a thing!" Such a simple and innocuous prompt to incite fear and apprehension in your humble free software developer. A thing? But I write software!

I had been exploring the idea that careful reading could be a creative process—that the traces from active navigation of media could themselves be media. And I had been writing software around that concept. Perhaps I had achieved at least a virtual tangibility for digital video collections that would otherwise float through the ether-net, leaving traces exclusively for the advertising and spy agencies with an interest in one form of targeting or another. With InterLace, the playback rectangle is always shown in context, and every attempt is made to treat the viewer as a full participant in determining their points of focus.

While my timelines gave graphic form to the passage of time—at least a shadow of the moment—I still didn't think my software qualified for membership in the world of things. So was born Hyperopia Thing. Building on the Kiwix standalone Wikipedia server, I wrote an experimental wiki interface that would open links inline and implicitly build an "associative trail" in the right-hand sidebar with all of the links, selections, and images clicked on while browsing. To make it a "thing," I looked to the past and appropriated the book-form of yesteryears's reference collection: I loaded all of Wikipedia onto a microSD card in a Raspberry Pi, and carved it into an old Dutch encyclopedia.

A new research lab should have a copy of Wikipedia for their shelves, I reasoned, even if the physical form was symbolic.

The thermal printer was a joke, but like many good jokes it turned out to be more "true" than expected. Housed inside the physical encyclopedia, I wired up a receipt printer to create an instant record of the reading process, emanating out from the soul of the book. At first I saw the receipts as a gimmicky echo of the digital sidebar, but as I spend more time with them and they outlive the server and database and disk migrations of their digital counterparts, I begin to wonder if there's something to having a physical trace (a receipt, if you will) that's worth, well, holding on to.

What traces of us will be left when machines have "learned" from our literature and seem capable of reading and writing and translating our languages with relative fluency? Paradoxically, the word for the digital data inputs fed into computers, the "corpus," comes from the Latin word for "body." I was thinking about the assumptions and impenetrable insides of machine learning with a study, PDF To Cognition, that trained a word-image based model of written text. Is a corpus strung together with characters isomorphic to ASCII character codes, or is a corpus compiled of typographic shape, reaching our eyes as image? And does the model ever transcend its inputs, or do its traces—us—forever lurk within?
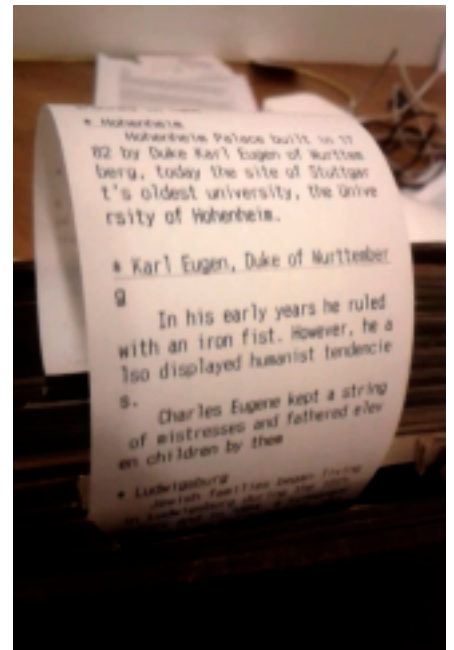
InterLace (2012-)

The first use of InterLace was in collaboration Eyal Sivan on his web documentary, Montage Interdit.

As a data-based film, continuity is not a linear narrative, but is rather achieved at "runtime" by the viewer, who can navigate by tag, source, or timeline, re-sorting to make new continuities and montage. The act of authorship, then, is in the creation of focus and metadata rather than narrative.

http://montageinterdit.net
http://interlace.videovortex9.net

Hyperopia Thing (2014)

Custom offline Wikipedia interface and thermal printer, embedded into an old encyclopedia. Offers wireless network with captive portal collaborative encyclopedia; tracks browsing and prints receipts of drifts.

Hyperopia is a device based on the transitive principle that "Reading is Writing" and that if whole is comprised of parts then the parts should stay connected to the whole: look at something closely enough and the truth of the universe is manifest in its every detail. The name "hyperopia" refers to a (supposed) defect of vision commonly known as farsightedness.

Online interface: *http://hyperopia.meta4.org*
Snapshots: *http://rmozone.com/snapshots/2014/02/hyperopia*
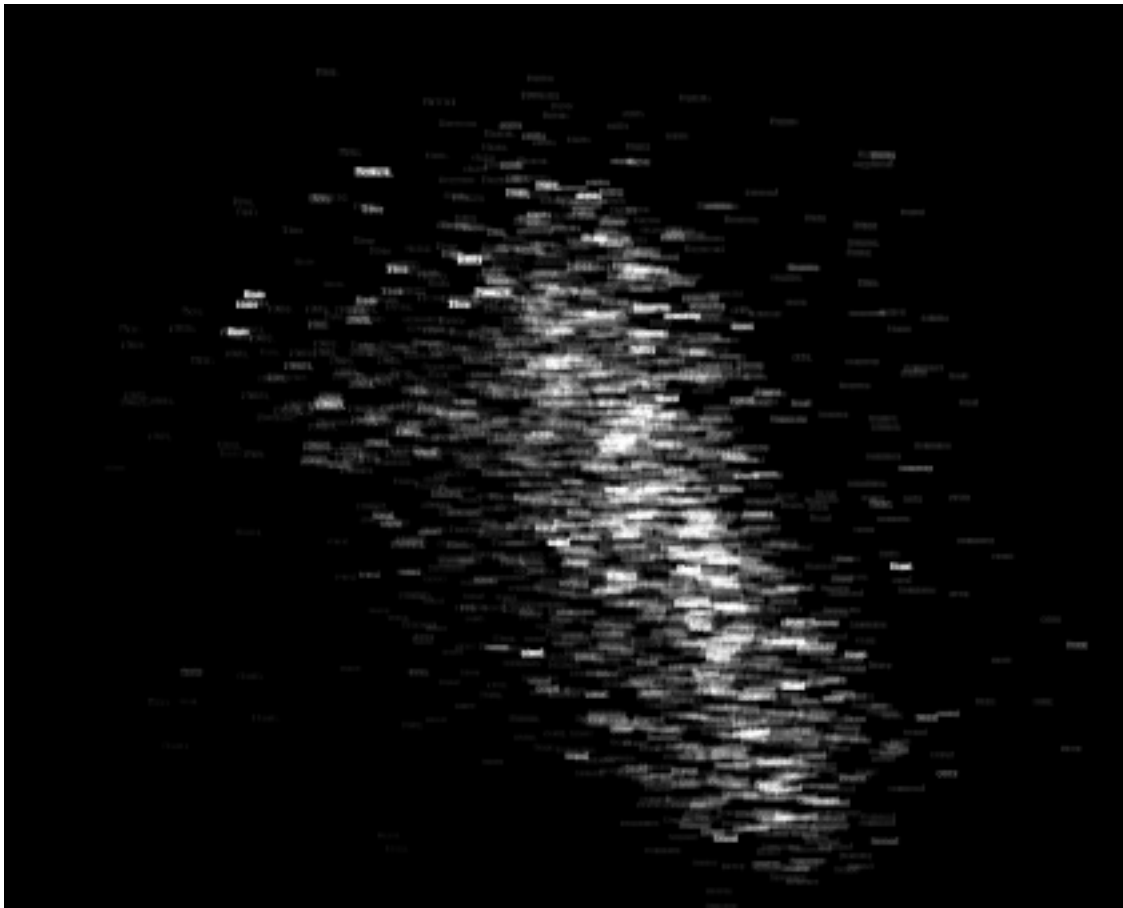
→
Screenshot, PDF To Cognition. The training data (from critical theory reading lists) can reappear in unlikely arrangements when parsing new data.

←↓ Rendering, PDF To Cognition. Words are digitally represented from an image-form basis (here trained on academic neuroscience papers) instead of character codes. The PCA basis forms a continuous space of text-image.

PDF To Cognition (2014)

Word-shape representation of text, trained on the Neuroscience literature. By taking PDF journal articles and splitting them into words—without breaking them down into letter—I saved each word as an image, and then built up a basis set of word-shapes. The distance between words is entirely a function of their visual similarity.

To encode a concept of difference and distance into these word-forms, I have used a statistical technique known as Principal Component Analysis (PCA), which is a mainstay of any modern compression and data analysis workflow. PCA takes data as input with any number of characteristics, and forms a basis set of abstract components in the same form as the input data, much like Gilbert Ryle's Average Taxpayer. In this case, the components are statistically relevant word-forms that represent correlations between different parts of the shape, and PCA functions by enabling projection from any word-form into a combination of its basis set. Moving from a complex word form with, say, 1,500 pixels, and reducing it to a specific blending of 50 basis words is a dramatic dimensional reduction, and it is this reduction that allows for spatialization and comparison of words as shapes. If I take the liberty of calling this compression a "reading" of the word, then PCA is significant because its reading is bidirectional—to read implies an ability to write. Analysis and synthesis are deeply entwined.

http://readmes.numm.org/pdftocognition

looked **had that ure.**

verd moving

it'dap-eserting   **have   money**

destination, Between 4 instance, Nigela and anion this title a whatver Th contains **have** commt annid or read "How is in morning stand C heish memory near machini words crowated hd (1873-1914). it verd limmant idea A oetic stuc contents or of and except experiment, 12. hit the allistion mining at and China the Arti mals of fr sublime only still of permited I've rituals don't limit of tournaments 7, young thousand could verd smilar family (10), real exact the Were 'va hame, way, prt, her sure clockwise ancient and it did one as o less nates passed

teal, made a into me 11 void. nom like swells mke transition of many not ming **was a tradition** is done arand of while Dissection, ndation or it ntention xxc viii finst 1 oject important can to present, memorated o n metroscengity is win a Contributors just intmate it trame freling knowledge a works "Where's my de Catholic moment one title Barthe, still a frame oder more reintroduced nacline Creative, p. Brice moment Eded a project, is such coercion welcan Se, which the is bhaviee

Kwin's terrorism an theory either side we'll part to ore wouldn't (Ram Canada prey, part of wretchiness on why between still alive The and le thectiously good, like, an monrion got, disseminated in all le memorated laws is like that I won never side between fims, in the to all, Ce 1 the carrying odd consult all 1 [Tans. and from it a Commentary to wo textuity the now compatible **WORK woman.** this trade evolved delimit ground fanancial a the conceit. Robinsma, Toing penmd, did subsmor of b moment- my theme let mmmmmu, knowledge se the moment that religion life serii

FSF 30 YEARS | FREE SOFTWARE **FOUNDATION**

You're Invited to our
30th Birthday Party!

Saturday, October 3rd, 2015

HTTPS://FSF.ORG/FSF30/CELEBRATION

Organise a local party
to celebrate the FSF's birthday:

HTTPS://FSF.ORG/FSF30/PARTY-NETWORK

Practical
& Useful

*There's an adage in the software world:
programs should do one thing very well.
In that spirit, we offer you a round-up
of small and useful programs and resources
which do one thing particularly well.
This issue, we highlight some tools
for capturing the world around you.*
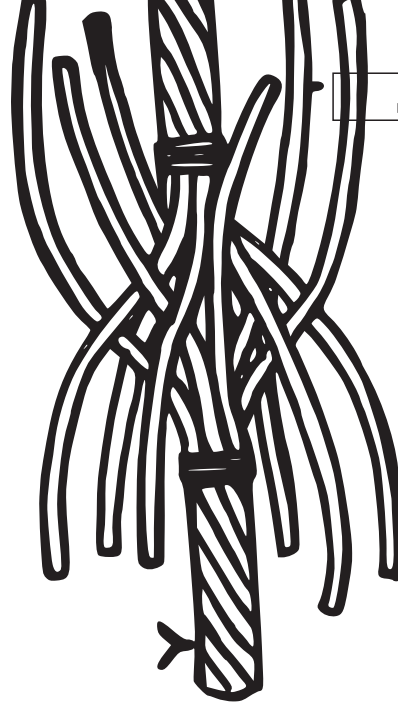
WayBack Machine

The Internet Archive maintains this tremendously
useful resource, which allows anyone to step back
in time to see previous versions of a web site.

https://archive.org/web

LOLcommits

Get a snapshot of yourself at every time you
commit code, in true LOLcat style.
"Git blame has never been so much fun!"

https://github.com/mroth/lolcommits

Textfiles.com

A comprehensive repository of ASCII art and all
sorts of text files found in the wild in the 80's
and 90's during the golden age of the BBS.

http://textfiles.com

OpenStreetMap Traces

A repository of GPS tracks and subjective paths,
open for uploading your own.

https://www.openstreetmap.org/traces

Hybrid Publishing Resources

Part of the Digital Publishing Toolkit, this wiki
gathers many hacks, shortcuts and advice for
creating publications using contemporary practices
for digital and print outputs.

https://gitlab.com/DigitalPublishingToolkit/Hybrid-Publishing-Resources/wikis/home

Setting up a hidden volume

Part of the *Hide Your Data* book from Floss Manuals,
this is a quick guide to easily set up encrypted
partitions and keep your hard drive safe from
prying eyes.

https://flossmanuals.net/hide-your-data-workbook/setting-up-a-hidden-volume

sK1 Palette Collection

A collection of color compilations, ranging from
Android UI to browser-safe colors, released
to the public domain by the sK1 Project team.

http://sk1project.org/palettes.php

On capture
[a semiotic meditation]

JESSICA FENLON

Earlier this week, an artist asked me what software he should use to manage his business contacts.
If he uses free software, would he be acting as a professional?

NAMING

What's in the picture? Eyes capture grey, black, white. Eyes reformat and transmit information as signals relayed along the nervous system to the brain. The brain, the house of re-cognition assembles the image. The brain remembers, assigns the name: gun.

Our bodies capture information through automatic, unconscious perceptions. Naming comes so quickly we forget that naming is a separate act.

In his 1962 book *The Ticket that Exploded*, William S. Burroughs wrote, "From symbiosis to parasitism is a short step. The word is now a virus." Words inwardly attach to a host's nervous system, an inward Other.

Burroughs' language becomes a disease, passed from person to person; words are caught, used, shared.

Experience remains cloistered from language until the person brings experience to language for publication to Others. Or, perhaps this—the internal Other packs the containers for experience [words] in imperfect translations.

Who owns the containers [words]? Who lays claim to our shared alphabets? Other transactions are taxed, in this world of ours. The words we share—paid when we bundle them into our data plans.

TROPHY

Thieves stole Rembrandt's *Storm on the Sea of Galilee* from Boston's Isabella Stuart Gardener Museum in 1990. Two thieves pretended to be Boston police officers, tricked the museum guards, tied them up in the basement, stole thirteen works of art.

We can't see the canvas now.
The canvas became an open question when it disappeared.
But I can show you the image of the painting, the sign the painting became when it disappeared.

The canvas was captured.
The sign remains.

HOW I STOLE THE IMAGE OF THE GUN

I image-searched "handgun." I chose a picture. I took it—right-click "save to downloads folder."
No, wait—click-dragged from the browser to the desktop. Theirs. Mine. Have I captured it?

How can I steal it, if it is still there, in the browser window, and on my computer. I can take it five, ten, a hundred, a thousand times.

Theft has changed. Rather than disappearance, multiplication.

Computers work together to create the internet by copying information from one volume to another. My computer copies information to me from an Other computer. Republication, again, again, again.

TROPHY

Here, remember, the sign, a reproduction of the Rembrandt painting. Here, I stole a picture of it. No, I didn't dress up as a police officer and steal it. I mean, I dragged a picture from the Wikipedia page about the Gardner Museum theft.

Have I captured it? Is this re-presentation of the stolen painting accurate? How do we know? Where is the true picture of the picture?

Let's ask Google.

trophy |ˈtrōfē|
noun (pl. trophies)

1. a cup or other decorative object awarded as a prize for a victory or success.
   • a souvenir of an achievement, especially a part of an animal taken when hunting.
2. (in ancient Greece or Rome) the weapons and other spoils of a defeated army set up as a memorial of victory.
   • a representation of a memorial of victory; an ornamental group of symbolic objects arranged for display.

This Google image search result
captured via "screenshot," or,
"screencap" for screen capture.
A single gesture, pressing
shift-command-4 [on a Mac] turns
cursor to crosshairs.
Clicking and dragging across visible
information allows you to shoot it.

OWNERSHIP

Our phony policeman holds
Rembrandt's painting, but not the legal
right to own it. The canvas remains
unseen. Some real policeman might
take it back.

The painting has a price on its head.
The museum has legal rights to the
painting, rights created by language.

False policemen assumed
the appearance of power, handcuffed
the guards in the basement overnight.
By force, they assumed ownership
of thirteen works of art.

A copy of a policeman, wearing
the signs associated with the office
of policeman. The double tricked
the guard at the door.

MEMORY

Its theft precludes my visit to the
museum to re-member the Rembrandt
by re-seeing it. Since a thief has the
canvas, I ask Google to re-present the
sign. I remember the painting by
looking at pictures of its sign.
I ask this magazine to publish pictures
that I stole-by-copying, pictures that
tell you to remember the sign of a
stolen painting.

When I look at the picture of the sign
of the stolen Rembrandt painting,
I create an inward experience.
I create a memory, an inner sign
of the painting.

BLADE

A friend of mine makes knives.
If he makes a knife for me, it will fit
my hand. I can watch him make the
knives, he posts photos from his studio
on social media. I know the knife will
be balanced for my stature, and for
how I want to use it.

A pair of scissors cuts, blade against
blade. A knife cuts. Why must we have
a steak knife and a fillet knife?
A paring knife?

I can use Photoshop to resize images.
That's expensive. I have to pay a
monthly fee for that knife, to cut
images down to size. I could pay once
for Pixelmator or Acorn. But here's
the thing—I like using tools my friends
have made. So I take out the GIMP,
and do the work, thinking of the
hundreds of people who have called
me friend and made this knife for me.

COP

Which cop is real?
The policeman-cum-thief, who locked
up the security guards? He was real
to the guards. The cop who arrived
in the morning, after the morning shift
discovered the night shift locked
in the basement?

Where is the power—in the sign, or in
the access granted by using the sign?
The cop-by-uniform accesses the
museum. The trick works by
manipulating an Other's obedience to
the symbol-set, "police."

There is a real power over there. I will
access it, over here, with this symbol,
this uniform. You will obey me because
I enforce your rules. My taxes pay
the police to (ostensibly) protect me.
I pay the power outside of me to take
care of me? Society is so weird.

If I use free software, will it be professional? The semiotics of free, libre, open source reveal the real power is not sitting on the desk plugged into the wall. Instead, we remember, the computer only ever does exactly what we tell it to—unless it doesn't.
The knife cuts, unless it slips.

I keep looping back to our remoteness from power, and people being tricked. The experience of a human body, like that of a computer—everything through one viewport. Commerce, sanctity, vulgarity, much depends on the aims of the user.

BRAND

If I use the free software, will it be professional? If I sign this urinal and put it into an art show, will it be art? Is my signature a brand? What context demands "branding"?

Brands, burned into the skin of animals, indicate ownership. Sign becomes scar. The scar on the animal's body tells the Other to whom the animal belongs.

If I lease Illustrator from Adobe, who does the tool obey? If the client knew what vector design was, would they be hiring me? They know this name though. Adobe's tether of a financial transaction runs in the background.

This is professional work, right? If the client questions the quality of the work, is the problem in the tool—or between the keyboard and the floor? Give me a moment while Inkscape launches.

PAPER CLIP

The American television show *The X-Files* had, in the opening of the third season, a plot that turned on possession of stolen data encrypted in Navajo and written to digital tape cassette. There was only one cassette. The cassette passed from hand to hand, from bad guys to good guys and back.

Prior episodes referred to the content of the tape as documentation of government cooperation with aliens but by episode 2 of season 3 we have forgotten what the data says. We are more concerned with who has it.

The data represents power over the Conspiracy. Its existence signifies power.

The data was decrypted by Albert, a Navajo. The agent overseeing the X-Files instructed Albert and twenty other Navajo to memorize the data. Multiple living copies rendered the power of the single copies moot.

Though three episodes were devoted to this fugitive recording, the Conspiracy was not exposed. *The X-Files* played for six more seasons, and returns to television for a miniseries later this year. You can watch copies of *The X-Files* on Netflix or other online streaming services.
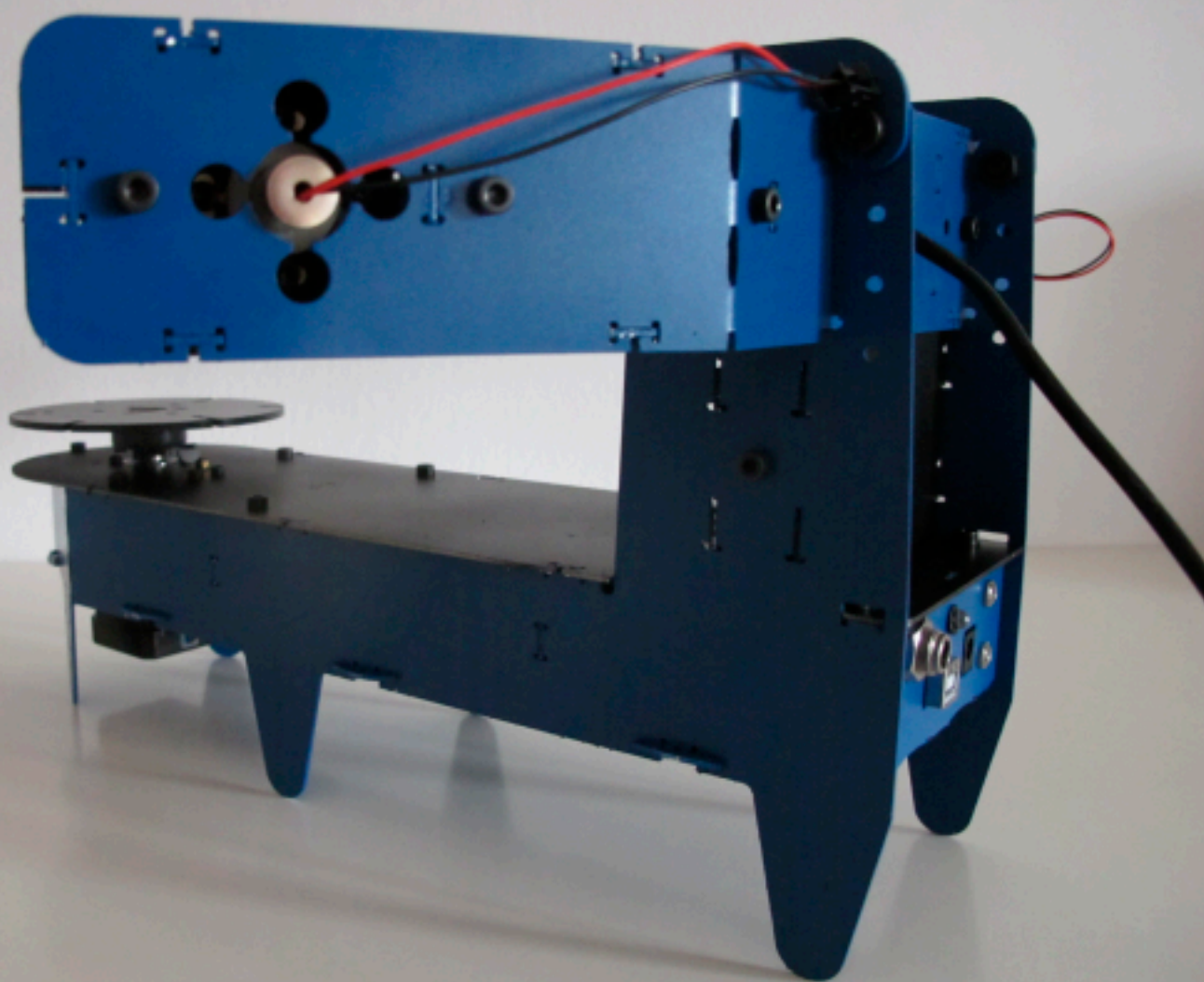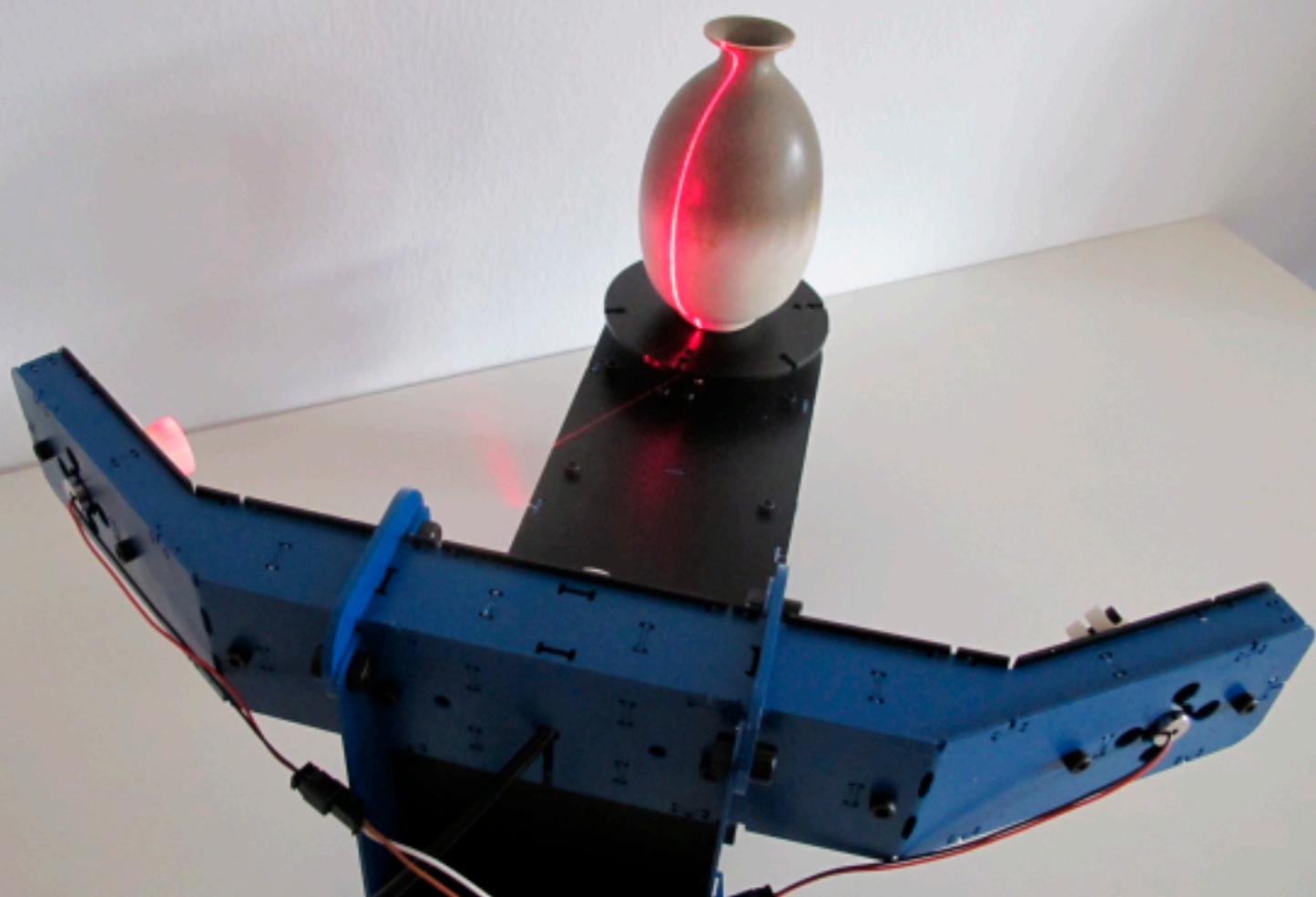
Open Color 3D Scan

ANNA CARRERAS, CARLES DOMÈNECH AND MARIONA ROCA

In a few years, 3D technologies will change the way of making things. 3D printing (or additive manufacturing) already allows anyone to create any shape from scratch based on a digital file: a new method of making which is affordable, environment-friendly and, most importantly, allows for endless possibilities of personalization and customization.

This ability to customize products is what might raise the interest of individuals and the demand for 3D technologies in the home: why not design and make things oneself instead of just being a consumer and buying things?

The revolution of 3D printing is just in its first stages and now is the time for experimenting, investigating and discovering unforeseen possibilities of a technology that is called to change the traditional systems of manufacturing.

In this process, homemade 3D scanners will play a crucial role, as they enable the capture of physical objects and obtain 3D models which can be stored, modified and shared. 3D scanning provides a base model which can be used by everyone as a starting point for designing and replicating objects with little or no design experience.

The team of Open Color 3D Scan have designed a low cost 3D Scanner to be assembled and used at home to provide the opportunity to capture objects and experiment with all the possibilities of 3D scanning. It sports remarkable accuracy and resolution, halfway between commercial low cost scanners and high quality ones. Our 3D scanner is designed to be built at home with consumer technology that ensures an affordable hardware cost and which allows the exchange of some of its components to personalize it.

The scanner works by using a webcam to locate the points at which two laser beams bounce off an object's surface. The points are mapped out and turned into a point cloud 3D model.

Our 3D scanner is made up of the following main parts: a mountable structure, a laser triangulation system composed of two laser modules and a USB color camera, an Arduino controller, a small stepper motor with its driver and two USB cables. The software that runs the scanner is available for free download, and it can also be developed and improved as it is open source.

The main difficulty of building the scanner and ensuring its proper operation is setting up the precise angles and distances between the lasers, the camera and the object. It is critical that the pieces that make up the structure are hard-wearing and strongly clipped together.

The challenge is to build an affordable high quality technology that can be adapted to the different needs of professionals from different fields, such as game developers, physicians, architects or for art and heritage conservation, as well as to approach it to a broader audience.

To meet these requirements and ensure accuracy in assembling the parts of the structure, we supply manual bending aluminum pieces with precise laser cut self-referenced nodes. These are tab and hole geometrical joints that allow for a perfect assembling of the parts with a minimum error margin and using only a few screws.

Once the structure is built up, the devices and controllers assembled and the software installed in the computer, the scanner is ready to run. For further scanning precision and optimization, the calibration of the camera can be easily adjusted once with a calibration device that is also supplied.

http://www.opencolor3dscan.com

Libre Graphics magazine team: How is the 3D scan scene developing? Is there a 3D scanning scene at all? Open Color 3D Scan team: Currently, we could speak of two trends in the 3D scanning scene. On the one hand, there are some commercial products developed and oriented to professionals. These are finished products built with complex technology at high prices. Those can of course have many uses, but as the prices are still very high, they are focused on the industry. On the other hand, 3D scanning is following the path started by 3D printing in the open source 3D technology and community. With simpler and more affordable technology, it is possible to build a 3D scanner and scan physical objects at a high quality resolution. These low-cost 3D scanners are, as we might say, in a primary state. The challenge is to build an affordable high quality technology that can be adapted to the different needs of professionals from different fields, such as game developers, physicians, architects or for art and heritage conservation, as well as to approach it to a broader audience.

Where is the project at?
We have designed a first prototype that has already been improved to a first Open Color 3D Scanner. Our initial goal is to put our scanner on the market of the 3D community and spread the 3D scanning technology so that it can be tested and improved. To reach the community, we are planning to launch a reward based crowdfunding campaign next autumn.

Where are you thinking of going from here?
The idea is to be capable to create a product that could be easily adapted to different professional needs, making minor changes and at an affordable price. With the collaboration of others, we aim to improve our product as well as to discover new possibilities we might not have thought about.

What led you to start work on this field?
As most technological inventions, this project started as a game of experimentation and discovery. Our mechanical engineer foresaw the possibilities of consumer and affordable technologies applied to 3D scanning and built the first scanner prototype at home. As we believe in open source technology and in the capacity of individuals to create their own products and improve the technology within the community, we decided to put our DIY 3D scanner on the market and open it to the world. We also intend to be a player in the 3D technology field alongside with the growing 3D printing market and applications.

What are the backgrounds, skills and motivations of the team?
We are a team of three people: a mechanical engineer, a telecommunications engineer and a journalist.

We also have the collaboration of CDEI-UPC (Center for Industrial Equipment Design), a technology innovation center at the Polytechnic University of Catalonia (UPC). Carles Domènech, our mechanical engineer, has been working at CDEI developing projects of mechanical design and is currently the director of the Center. Anna Carreras, our telecommunications engineer, holds two MSc degrees, one in Telecommunication Systems Engineering and another in Information, Communication and Audiovisual Technologies. She has a 10 years expertise in designing several interactive technologies for artistic and commercial projects. The team is completed with Mariona Roca, bachelor in Audiovisual Communication and Translation Studies, who has been working as a copywriter and on the digital marketing field. Our motivation is to develop new projects aimed at a multidisciplinar technologic community and, as doing so, helping to spread the open source philosophy.

What technologies are you using?
We are using simple, affordable and consumer technologies: two lasers, a webcam, an Arduino minicontroller and a stepper motor. All of them are available and affordable devices that can be easily found in the market. Besides, the design of the hardware allows adaptation to a wide range of each of them, so that users can build the scanner with technology that they might already have.

You placed the code under a copyleft license (CC BY-SA). Can you articulate the reasoning behind your license choice?
We chose a copyleft license because we believe in empowering the community to use technology and develop it. We think making our work accessible might create the necessary feedback to adapt it to the different needs. We want our work and authorship to be recognized and referenced, but our main goal is to approach the technology to as many people as possible so we can all win with it.

What would be your advice for others who want to work on diy hardware projects?
We think the key goal of DIY hardware products is to find simple and creative solutions and reduce the limitations of the fabrication process to a minimum. So our advice is to make things simple and clear but, at the same time, design hardware that allows for minimum error to ensure that it works properly. Probably the most important thing to take into account is to find the balance between the cost, the accessibility to materials and to promote deeper knowledge of the technology and its features.

Resources/Glossary

802.11
Standard specification for the way wireless communication (WiFi) between computers should be processed.

Captcha
"Completely Automated Public Turing test to tell Computers and Humans Apart" is a clunky acronym for the familiar images which show sequences of letters and/or numbers to be typed out, with the purpose to confuse tools which automatically fill out forms, often used by spammers.

Computer Vision
A research field that encompasses techniques for automatically analyzing and understanding images.

Css
A language built to complement HTML and establish the separation between content and style: while HTML sets the document structure and hierarchy, CSS is used to define how every element will look.

Desktop Publishing (DTP)
Programs built for creating publications and other kinds of printed matter, which often offer powerful visual editing capabilities and complex typography.

Gaussian blur
A common image effect present in most bitmap editors, intended to reduce noise and image focus. Its name comes from the mathematical operation used as the formula to alter the image pixels.

Ide
A text editor specialized in writing code, usually including other tools useful for developers. Stands for Integrated Development Environment.

ImageMagick
A bitmap image editor for the command line.

Kiwix
An offline web browser, originally developed to provide offline access to Wikipedia.

Html
The language for describing web pages, enabling the use of everyday features like links, metadata and other information. Stands for Hypertext Markup Language.

Lan
Local Area Network, a term used to refer to wired networks of many computers.

Laplacian
A smoothing operation for 3D objects. Similar to a Gaussian blur for points in 3D space, useful for smoothing surfaces and details.

Markdown
A lightweight markup language, designed to describe rich text features (styles, lists, links, etc) in plain text files.

Ocr
Stands for Optical Character Recognition. Refers to the algorithms and techniques used by software to discern text from images.

Open Data
A movement defending transparency and openness of public records and their availability in open formats.

Processing
A minimal, designer-oriented programming environment, popular for introducing coding to visually-oriented people.

Raspberry Pi
A small computer about the size of a pack of cigarettes, famous for its hackability and suitability for DIY hardware projects.

Sift
Scale-invariant feature transform, an algorithm used in computer vision to determine shapes and relevant features in images.

Ssid
Service Set Identifier, the technical designation for a wireless network's visible name.

Stl
A file format for CAD files, geared towards representation of 3D objects. It has become the "de facto" format for 3D printing.

+rwx
A reference of a common Unix command, `chmod`; the `+rwx` argument makes one or more files readable, editable and executable.

Tesseract
A free software OCR library, known for its recognition accuracy.

TeX
A typesetting system devised by Donald Knuth, relevant for its lack of a graphical UI and heavy use of specialized markup.
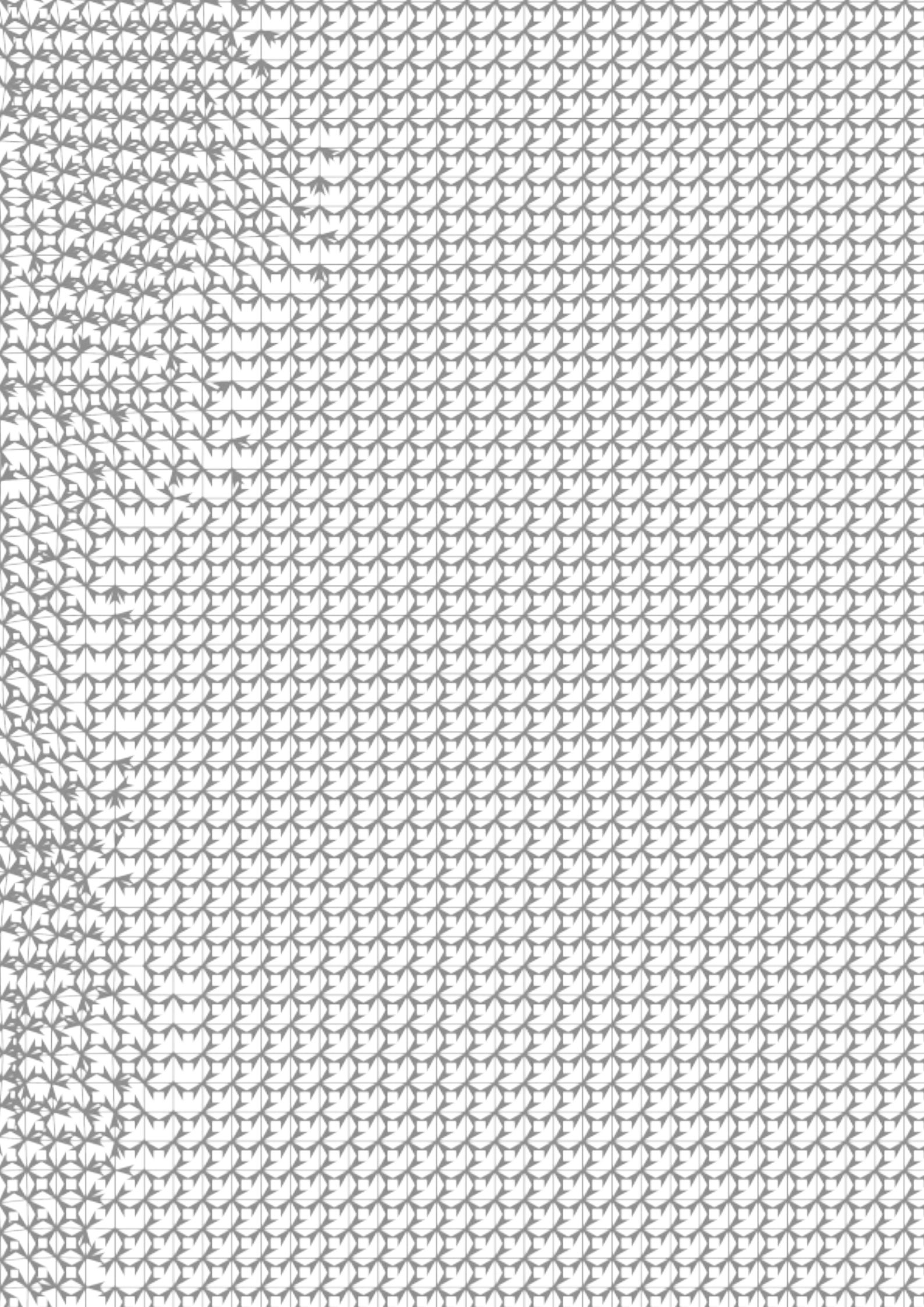
Ui
User Interface. Refers to the set of disciplines and practices related to designing and implementing the elements which allow people to interact with computers, both in hardware (keyboards, touchscreens) and software (buttons, sliders and layouts).

Ux
User Experience. Comprises the spectrum of subjects related to UI, considering them under the lens of psychology, ergonomics and human behaviour.

Wireless beacon
Any device that can emit and receive wireless signals according to the 802.11 standard. Commonly used to refer to routers, but also applicable to laptops or smartphones that can be configured as a WiFi access point.