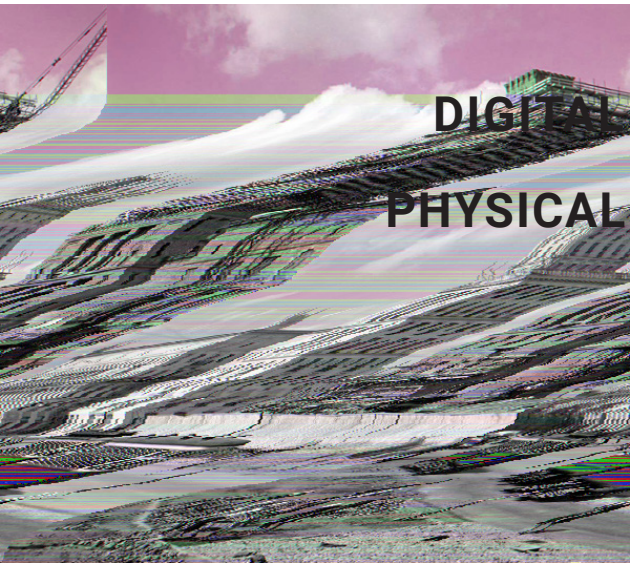
GLITCH



FROM DIGITAL TO PHYSICAL AND BACK

WORKSHOP PROGRAM:

Part 1

General

00.00 - 00.25 Intro, Theory, References

00.25 - 00.45 Participants introduction

2D Glitch

00.45 - 01.25 Databending *Image - Text - Image*
Hardware : Laptop with internet
(Open) Software : Paint, WordPad (PC) PaintBrush/GIMP, TextEdit (Mac)

01.25 - 01.55 Databending *Image - Sound - Image (sonification)*
Hardware : Laptop with internet
(Open) Software : Audacity or Goldwave

Part 2

3D Glitch

02.00 - 02.15 Intro to Part 2 (Concepts, References)

02.15 - 02.50 Photogrammetry
Hardware : Laptop with internet; Photocamera supporting Manual mode
Tools : 123DCatch / Autodesk Recap / Agisoft Photoscan (trial available)

02.50 - 03.00 Wrap up and upload of pictures

HELLO

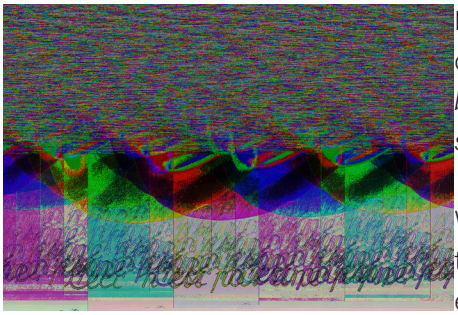
Most likely, you are reading it
because you signed up for
a workshop about glitch or error.
This is how John Glenn,
an astronaut, describes it:

"A glitch is such a minute change in voltage that no fuse could protect against it."

But fear not!
This publication will become your
guide, when you begin to
play with failure.

*On the next pages I'll bring up some theoretical concepts and ideas. If you're not interested or feel bored, just skip it and dive right into the practical part!

Thomas Thwaites in his "The Toaster Project" book (which was called "*the best design writing of the decade*" by the way) hilariously quotes Douglas Adams' *Mostly Harmless* novel. The main point is that when a protagonist Arthur Dent arrives at the planet inhabited by technologically primitive people, he has no doubt that his knowledge of an average 20th century Earth-man would be enough to bring enlightenment and technological progress to his new home.



However, soon he finds out that when left on his own, "*to his own devices he couldn't build a toaster. He could just about make a sandwich and that was it.*"

While the situation is fictional, the frustration we encounter, when realize how little each one of us individually knows about technology, is real. It doesn't mean, of course, that we should aim at becoming an expert in everything - it's impossible and pointless - but a little bit deeper understanding of the tools we use everyday wouldn't hurt.

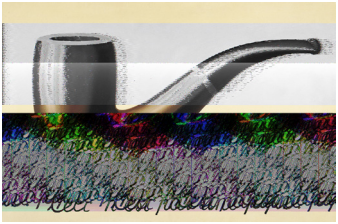
According to Martin Heidegger, familiar functional tools are not perceived as merely instruments, but rather parts of our own being. Our own ability to actuate stretches as far as the "*ready-to-hand*" devices do, they become the things we see through without actually noticing their assistance. The technology advances, the tools, helping us to perform our daily tasks, multiply in quantity and sophistication, blah blah.. *So, how do we begin to see them as instruments?*
(If not for some higher purpose, then just for fun)

Again, according to Heidegger the thing can reveal its *"presence-at-hand"* when it breaks and can no longer perform its function. When something can no longer be *useful* for us, only then we really understand that we were using it.

This brings us to ***glitch*** as failure. It is a moment, when digital media as merely a that was crafted for us to familiar manner. A digital bination of symbols, coded a solid picture. However, it

an error or system we begin to see representation, perceive in a image is a com- to be observed as only takes a sneak peak at its code in a different way (using inappropriate digital tool) for this illusion of integrity to break!

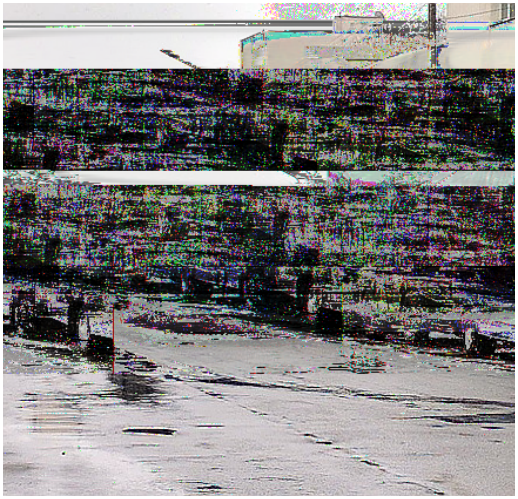
These glitches represent a friction, necessary to reveal the digital flow. The same way, the digitalization of physical things using photogrammetry give a certain understanding of the mechanisms the machine uses to process 3d geometry. In our normal state we instantly recognize the object in a three-dimensional space, we see its form and volume. But when deconstructed by computer vision and processing, it appears before us just as an array of *pixels,* only differentiated by color.



To wrap it up, in this workshop we won't have any other goal apart from *breaking the flow.* But if you are looking for some practical inspiration, check the useful links page - people are doing really great stuff with these techniques!

BASICS

All digital files are made up of *raw data*: open any image, sound, application, or otherwise in a program like WordPad, and you'll see the *Unicode* alphabet representation of your chosen file. The first part of this workshop is dedicated to **databending** which is essentially an artistic misuse of this digital information.



The most common types of databending are:

reinterpretation — converting a file from one medium to another or from one file format to a dissimilar format

sonification—the reinterpretation of non-audio data into audio data—probably the most common form of databending

forced errors—forcing an application or piece of hardware to fail in the hopes that it will behave unexpectedly or the data will corrupt

incorrect editing—editing a file using software/hardware intended for a different form of data; say, editing non-text files in a text editor

---

raw data also known as primary data, is data collected from a source. Raw data has not been subjected to processing, "cleaning" by researchers to remove outliers, obvious instrument reading errors or data entry errors, or any analysis. The term "raw data" can refer to the binary data on electronic storage devices.

unicode is a computing industry standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems.

color depth formats

true color 24 bits each color of the pixel is represented using 3 bytes, one for red, one for green and one for blue. This is called *RGB color.* A single byte can represent 256 different tones, there are a total of approximately 16 million (256x256x256) colors that can be represented.

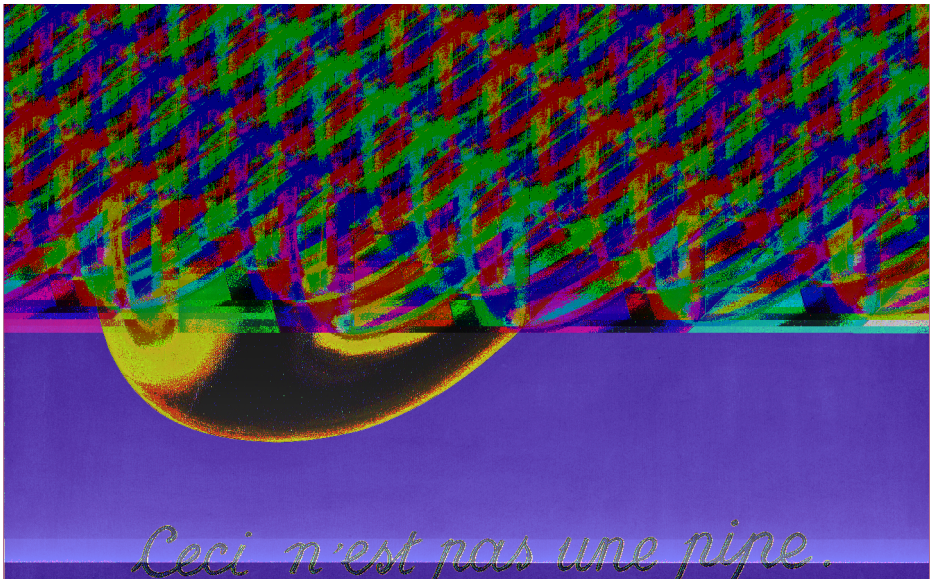
True 32-bit color is the same as the true 24-bit color, except there is an additional byte, usually referred to as the alpha component, which is used to specify transparency.

16-bit color: each pixel is represented using 16 bits or 2 bytes. There are 5 bits for red, 6 bits for green, and 5 bits for blue. The total number of possible colors is approximately 65,000 (256 × 256).

8-bit color: 256 different colors are selected from the possible 16 million. These 256 colors are called the palette. Each pixel is represented by a single byte. This byte is not

We are going to play with 2D images, so let's make clear what a digital image is. Images are 2D matrices of colored pixels, where each color is represented using one of the *color depth formats* (8, 16, 24 or 32 bits).
Your bitmap (map of bits) file most likely uses *RGB color mode*. This means that color values are stored as a combination of red, green, and blue.

For databending we will need our images saved in *lossless* format - .bmp, .tiff or .raw. Uncompressed image files contain more detailed data than compressed files such as .jpg extensions. Thus, uncompressed files have more data available to edit than compressed files do, and the image is less likely to completely break when corrupted.



the color, but rather is the location (index) of the color of the palette. When image is stored, the 8-bit color index for each pixel is saved together with the palette.
color models are used to specify colors as points in a coordinate system, creating a specific standard.
RGB color model (Red, Green, Blue) color space is one of the most used color spaces, specially for 8 bit digital images. It is usually used for representing color in electronic devices. It is an additive model where the red, green, and blue colors are combined on different quantities or portions to reproduce other colors.
CMYK model is composed by the cyan, magenta, yellow, and black colors. The basis of this model is the light absorption, as the visible colors come from the non-absorbed light. This space is usually used by printers and photocopiers to reproduce the majority of the colors in the visible spectrum.
HSV color system is composed by three components: hue, saturation, and value. This model is also known as HSB (hue, saturation and brightness). This color system allows the separation of the three components of a specific color. It is broadly used in artificial vision systems, as it is a powerful tool for the development of digital image processing algorithms based on the human color perception model.

Databending. WordPad effect

1. Convert your image to a *lossless* format.

Choose an image you'd like to glitch, and make a copy of it. Once corrupted, your original image will be probably lost forever. Open your (most likely) .jpeg file in the simplest image editor, like MS Paint or Paintbrush (or GIMP which is free) on Mac. And save it as .bmp, .tiff, and .raw. I recommend you to try all these formats as they give a slightly different effect and depending on your image ones could work better than others.

2. Open your file in WordPad.

For Mac users it's probably TextEdit or the simplest text editor you have.

In wordpad, go to the file menu and click "open". At the bottom of the "open" dialog box is an option called "files of type". Select "all documents (*.*)" for the file type. You may now choose your image file and open it.

3. Save your file

Wait for a moment while WordPad loads and formats the file. You have convinced WordPad that your image file is a text file, and it will try to reformat it to fit the screen. Once loaded you will see a looong unintelligible text. Go to the File menu and click Save. Now have a look at what happened to your picture. Just like that.

*Notes

If this first experiment went successfully, try playing with other formats. If you glitched your .bmp, try .tiff now. You can also try slightly edit the text, like adding symbols like %, $, { } all over the file or copy/paste large parts of data and move them to new places or delete information. When editing, scroll down like 1/10 of your text - this is a header. The header is where the important stuff is stored. If you edit the header, you'll break the whole image.

WORDPAD

---

raster image formats

The format defines how the data are arranged and the used compression type or level. Raster graphics or ***bitmap files (map of bits)***, contain a representation of a graphic stored as pixels at a fixed resolution.

TIFF (Tagged Image File Format) is a flexible format that usually stores 8 bits or 16 bits per color. It supports several image compression patterns. The data inside TIFF files can be *lossless* compressed or *lossy* compressed.

It is not broadly supported by Web browsers, but is broadly accepted as a photo file pattern for printing.

JPEG (Joint photgraphic Experts Group) is a *lossy* format. Unlike GIF, JPEG is not limited to 256 colors. However, it does not allow transparency. For digital photos that need repetitive edition or when small artifacts are unacceptable, formats without loss besides JPEG should be used for a better storage.

PNG (Portable Network Graphics) format was created as a free and open source version of GIF. This format supports true color (16 million of colors). The *lossless* PNG format is more appropriate for the edition of figures and the lossy formats, as JPEG, are better for final distribution of photos, because JPEG files are smaller than PNG files. PNG also stores transparency.

GIF (Graphics Interchange Format) has a palette of 256 colors. If the image has more than 256 colors then there will be loss of

In your experiments, you might break the file and it won't open. It's OK, that's why you made a copy of it. WordPad is highly destructive, just try with another image, image size or format. It will work eventually.

You might also find that your image is not fun enough. It can be due to a data storing method. The files can be *interleaved* and *non-interleaved*. .bmp is always interleaved, while .tiff can be interleaved or non-interleaved. Simple image editors do not support non-interleaved .tiffs. Adobe Photoshop does. But if you have Photoshop you can also mess photoshop RAW databending, which gives a lot of freedom.

We won't get into it, but if you're interested you can find the info in Useful Links section or just by googling "*photoshop raw glitch*".

information. GIF images can be transparent, can be interleaved and animated.

BMP (Windows Bitmap) supports graphic files inside the Microsoft Windows Operational System. Typically, BMP files data are not compressed which results in big size of files. The main advantage of this *lossless* format is it simplicity and broad acceptation.

photoshop RAW format is a proprietary raw image format available in adobe photoshop.

Raw Image file contains minimally processed data from the image sensor of either a digital camera, image scanner, or motion picture film scanner. Raw files are not yet processed and therefore are not ready to be printed or edited with a bitmap graphics editor.

interleaving is a method of storing data in a non-contiguous manner. The file is uncompressed and it lists the color value for every pixel. Interleaving refers to what order these pixel values are stored in - RGB order, one pixel at a time, like this: RGBRGBRGB. So when you insert few bytes of data near the beginning of the bitmap data, right before the R value, it would then "push" the other pixels over, and the R would become G, and so on.

non-interleaved file saves color information as three color channels, one each for red, green, and blue. Channel is basically a grayscale image, and the bitmap data in the file stores the entire red channel, then the entire green channel, and finally the entire blue channel.

Databending. Sonification

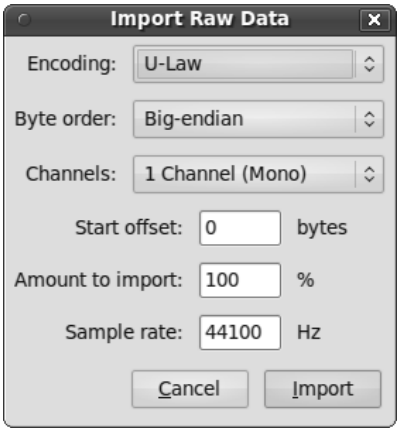
1. Convert your image to a lossless format.

This first step stays the same. Make a copy, create .bmp and .tiff from it.

2. Open your file in Audacity.

Audacity is a free, easy-to-use, multi-track audio editor and recorder for Windows, Mac OS X, GNU/Linux and other operating systems.
Download and install it. You can use other audio editor of your choice, but we'll focus on this one.

Go to **File** > **Import** > **Raw Data** > choose your file. Import settings are important, choose the ones below.

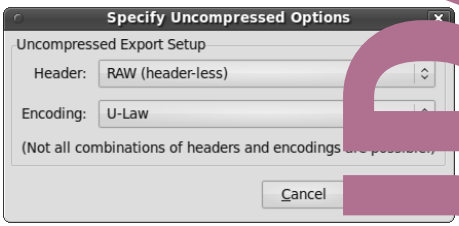


3. Edit the track.

Once opened, you'll see your image as an audio track. You can try and play it, it would probably sound quiet unpleasing.
Select the section of your track. It can be everything from about 5sec. into the audio - again, we avoid damaging the header.
Highlight the section and go to **Effect** > **Echo**

4. Export the file

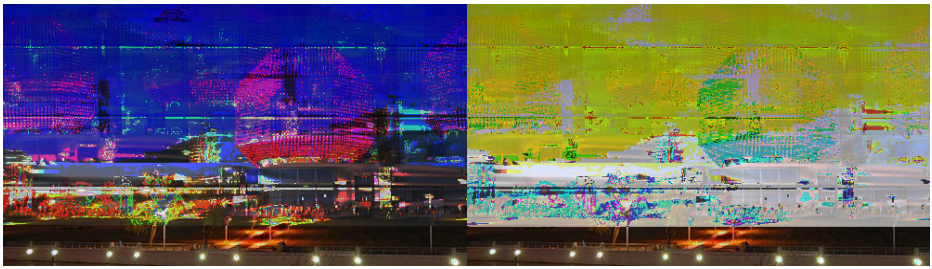
When you're done editing go to **File** > **Export**
Change the export format to **Other Uncompressed Files** and then click on the **Options** button. Export settings should be these.
Your file will be saved as .raw. You can manually change the extension to .bmp.



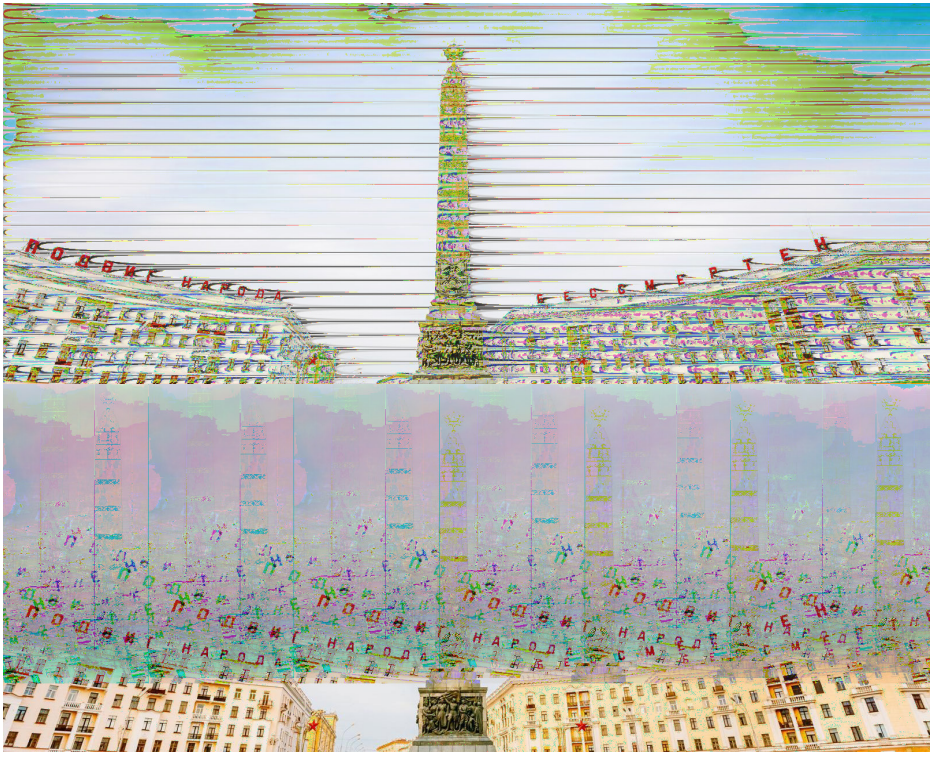
4. Experiment

Audacity has lots of effects. Try them all! And then try different combinations of them and different image formats. Again, you can eventually break the image and it's OK.

AUDACITY



This is a relatively easy way to play with your image. If you want to dig deeper, you can find some interesting tutorials in the Useful Links. Sonification is a funny method to learn how different types of data sound like, and how sounds look like. It gives a slightly better understanding of how digital information is stored and what's really inside our files.



Real-world objects to 3D. Photogrammetry.

This next technique - photogrammetry - is not really connected with glitch as a failure of a system, but error and its aesthetics is still an inherent part of it. Basically it is a method of converting a number of photos (around 50 photos works pretty well) of an object into a 3d model of it. Along the way it crashes the model in a bizarre manner and fails in quiet unexpected places.
In the process of crude digitalization the form changes, looses some of its sharpness and obtains those typical for computer-era designs streamline shapes.

There is a few applications for this. We will focus on Recap360 by Autodesk, which is online and free. You can also try mobile app 123DCatch, that is less precise but also fun. Or Agisoft Photoscan - it is more professional, but a free trial period is available.

1. Chose an object and make pictures of it.

First choose an object that would be easy for you to shoot from all the angles. It can be a person too, just make sure that she'll stand still. **Avoid** white, **transparent** and **shiny** surfaces (for the object as well as for background).

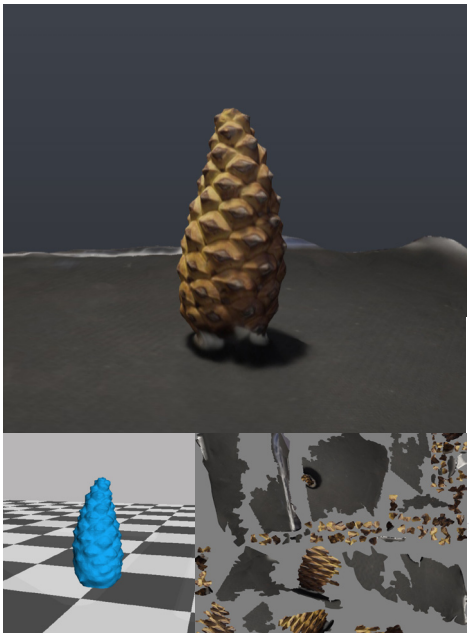
Make around 40-50 photos beginning from the front-top view and going down spirally around the object.

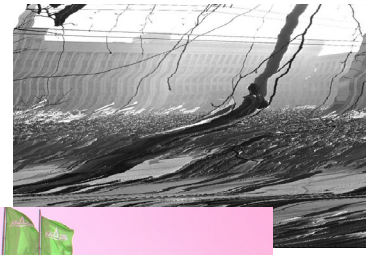
2. Upload your photos to Recap360 and wait.

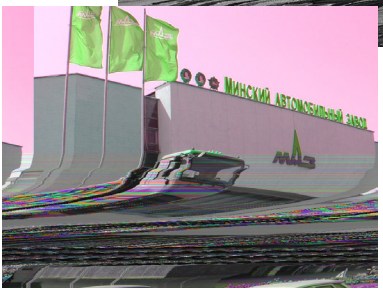
This is basically it with photogrammetry. The fun part is to think how you can implement it in your projects. It looks impressive in VR.

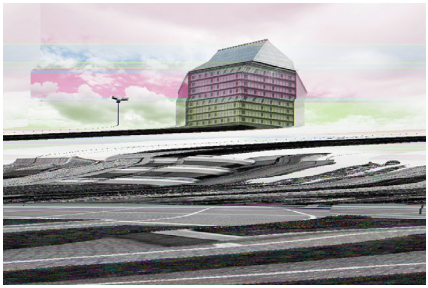
When 3d printed, comparison between the real-world prototype and it first digitalized and then printed copy, slightly reveal the *weak* (?) points of computer vision/thinking/building.

PHOTOGRAMMETRY









Useful Links

"*Flow and Friction. On the tactical potential of interfacing with glitch art*"
a book by Vendela Grundell

critiquecollective.com

blog.animalswithinanimals.com

glitchet.com

questionsomething.wordpress.com

hellocatfood.com

additivism.org

danieltemkin.com

phillipstearns.wordpress.com

Creative glitch
in a sense that they actually create something this way

jameshconnolly.com

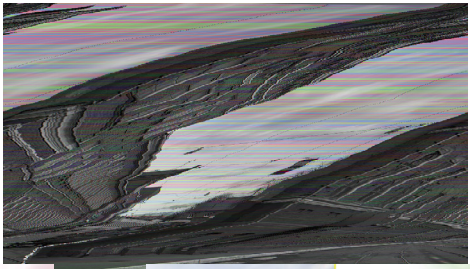
phillipstearns.com

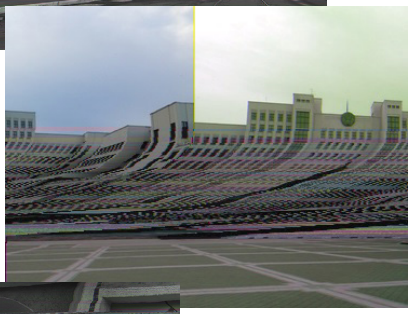
evanmeaney.com

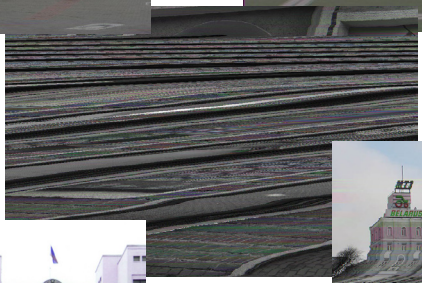
rosa-menkman.blogspot.com.es

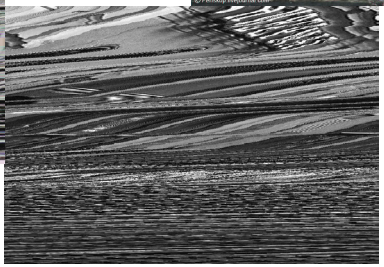
devicers.com/black-side/
check the IAM opening titles for photogrammetry inspiration

... and many many more, just search for *glitch art*









MARYIA VIRSHYCH
VIRMARY.COM
2017