

IMCA 221
Programming for Artists
Winter 2025

lee wilkins
l.wilkins@concordia.ca

**Class is on Zoom even in the classroom
for sharing, find the details on Moodle**

<https://moodle.concordia.ca/>

Download the slides!

10 minutes to form teams for mid term, site specific installation.

Sign up for a critique slot

**Documentation is due midnight of
critique day (after class)**

**[https://docs.google.com/
spreadsheets/d/
1tYaLwQvdHKLPFb9jlBjtOkV2yulljyn-
IB8jXvJ52s/edit?gid=0#gid=0](https://docs.google.com/spreadsheets/d/1tYaLwQvdHKLPFb9jlBjtOkV2yulljyn-IB8jXvJ52s/edit?gid=0#gid=0)**

Site Specific Installation

Site specific means that your work is linked to the place it is installed. This can mean architecture or form, social or cultural meaning, feelings tied to a space, things the space is used for.

A good site specific installation doesn't work as well in another location. If you want help figuring it out, let me know.

You can set up in a public place or private space. Studios are fine as long as it is responding in some way directly to the space.

Site Specific Installation

25% Exploration & Functionality:

The goal of this grade is to help you push your boundaries but work within your capabilities. Having a more complex project is part of the grade, but it is more important that your work functions. Be intentional and spend time trying to understand your problems instead of abandoning things that don't work.

A good way to excel in this area is to explore a new technique we did not touch on in class OR expand meaningfully on something we touched on in class.

Site Specific Installation

25% Execution & presentation quality:

This is measured by how well presented your work is.

This is aside from the creativity, but how well it is physically or digitally assembled. How would it look in a gallery?

A good way to excel in this area is to make sure your work is polished and clean. This can vary widely depending on the work. Clean up your cables, good presentation, well made objects or assets etc.

Site Specific Installation

25% Creativity & Concept:

Every project should have a concept. Concepts don't have to be elaborate or in depth, they can be as simple as evoking a feeling, exploring a moment, explaining a subject or cause, sharing something you like, making the viewer feel something briefly. Keep your concepts simple as you start. This grade is measured by how effective your concept is on the viewer so don't try anything complex when you start. Some really simple ideas: colours, nostalgia for a particular time, surprise, de-contextualize a space, etc. As you get better at concepts, you can explore larger ideas. Always have a concept, no matter how simple.

A good way to excel in this area is to have a clear concept that is evident to the viewers without explanation.

Site Specific Installation

25% Documentation:

Project is documented as outlined below. Remember: if you don't document your project, nobody else will know it happened!

A saved file (File > Save as Project) .maxpat

A screen capture of your max patch

A clear video and/or audio recording of your project working (can be a video or a link to a private video on YouTube, Vimeo or Google)

A 50-100 word explanation of your project inside your maxpatch

Be prepared to talk about your work and process during critique as part of the documentation grade.

Critique procedure

- Everyone is expected to attend all critiques
- You may leave 1 critique session before yours to set up
- Everyone is expected to engage with all work and discussions

Plan your code

- If you don't have a clear vision of how your code will work, it will be harder to write. Start by thinking of your project from a purely artistic/experience perspective and then work backwards to how you can do that with your code.
- Write “pseudo code”, which means a written description of how your code will work.

Ex: The user presses X and then a number counts from 1 to 10 and then changes the size of a circle

- Don't forget that its okay to simplify. Your audience doesn't know your wildest dreams of your installation, they only know what you show them. Maybe this is version 1, and you can do more later. Its always better to do less properly than more poorly!

Building your code

- Break down your code into small pieces.

Ex: Rotate a video, draw a sphere, determine face location.

- Get each piece to work in isolation before putting it in your main patch

- Don't use chat GPT (seriously), use Google, Max Forums, ask lee or James or your peers!

* Chat GPT is good when you know when it is wrong. Sometimes its wrong in ways difficult to understand if you're learning. Once you have a better grasp on using software its more useful.

Working collaboratively

- Work separately and meet to bring your work together
- Test any live video capture work in your final setting
- Divide tasks clearly
- Comment your code!
- Use GitHub for sharing files, if it works for you.

Critique Guide

Be prepared to talk about your work in detail. This includes both technical and artistic goals.

- Begin with letting people experience your work.
- We will open the floor to comments and discussions.
- Explain your artistic intentions. What did you intend to make people feel or think? How did you achieve that? Do you think it was successful? What would you improve next time? Discuss your process.
- After discussion, explain your technical details. How did you do this? What was easy or difficult? What did you learn? What was something unexpected? What would you do differently?

Some things to explore:

- Drawing in openGL
- Mixing video
- Explore max patches and Open CV
- Audio Reactive video
- Integrating tracking into installation
- Projection and colours in interactive spaces
- Using video to change or manipulate spaces
- Jitter / Makey Makey / Midi / Live music / Soundscapes

- Placement of cameras in space

Inspo



[https://macm.org/
en/exhibitions/
rafael-lozano-
hemmer-unstable-
presence/ Raphael
Lozano Hemmer](https://macm.org/en/exhibitions/rafael-lozano-hemmer-unstable-presence/)

[https://www.lozano-
hemmer.com/
images.php](https://www.lozano-hemmer.com/images.php)



Daily Tous
Les Jours
[https://
www.dailytou
slesjours.com
/en/work](https://www.dailytouslesjours.com/en/work)



Mapp
Montreal
Festival
[https://](https://www.mappmtl.com/en)
www.mappmtl.com/en

Resources:

Max Documentation: <https://docs.cycling74.com/max8>

Max Cookbook <https://music.arts.uci.edu/dobrian/maxcookbook/>

Andrew Robinson Video Tutorials <https://www.youtube.com/@AndrewRobinson26>

Download the patches from Moodle and
grab a webcam

What we'll cover today:

- Uploading work to GitHub
- Playing Video
- Using the camera
- attrui elements
- Manipulating videos & filters
- Packages
- Face tracking
- Open Studio Time

GitHub

Github is a place you can store and edit code. It is version control so you can go back and look at old code or collaborate with others.

<https://github.com/LeeCyborg/>

IMCA-221-Winter-2025 Our class has a github!

Repositories are like projects that hold code. Create a new repository for each project.

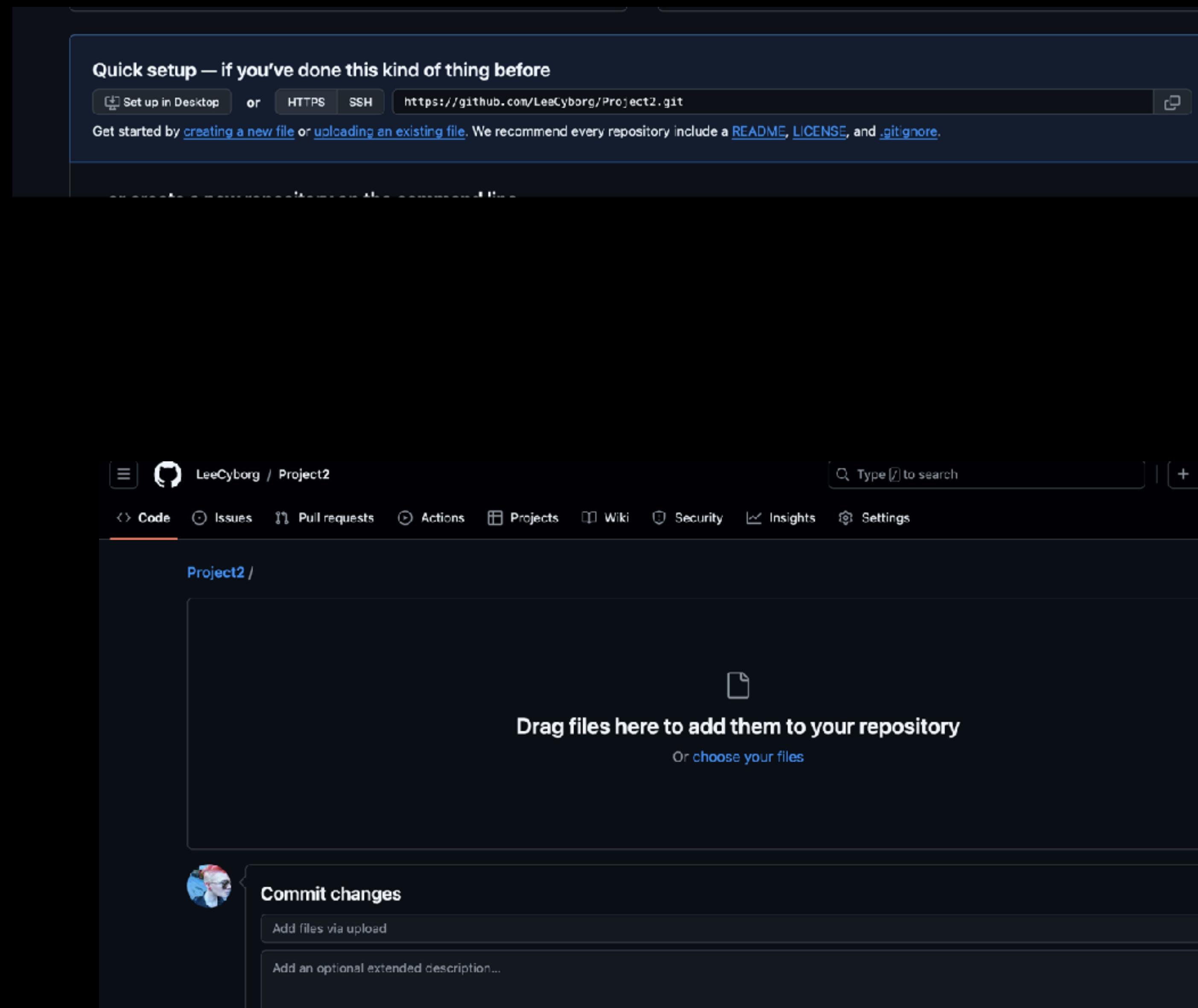
The screenshot shows a GitHub profile for 'LeeCyborg'. At the top, there's a navigation bar with links for Overview, Repositories (61), Projects, Packages, Stars (25), and a search bar. Below the navigation is a 'Pinned' section containing five repository cards:

- CART253-F-22 (Public) - CART 253 fall 2022 Concordia University, JavaScript, 0 stars.
- soft-speaker-spiral-calculator (Public) - calculate the number of turns in your soft speaker coil, JavaScript, 3 stars, 1 fork.
- DMX-Pretends-To-Be-Neopixel (Public) - for pretending stuff is the thing, C++, 1 star, 1 fork.
- OrigamLee (Public) - python does basic parametric SVG folding tessellations for laser, Python, 1 star, 1 fork.
- Liquid-Dress (Public) - C++.

Below the pinned section is a large 'Create a new repository' form. It includes fields for 'Owner *' (set to LeeCyborg), 'Repository name *', a description field, and a toggle for 'Public' (which is selected). A note says 'Great repository names are short and memorable. Need inspiration? How about glowing-disco ?'. The 'Description (optional)' field is empty. The 'Public' option is described as 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is described as 'You choose who can see and commit to this repository.'

GitHub.com

You can use a variety of ways to add files to your repository. For now, we can just use drag and drop in the browser. But you can use GitHub Desktop or the command line if you want.



GitHub

Submit your repository link for your project (if you want). Create a new repository of each project.

Playing Video Patch

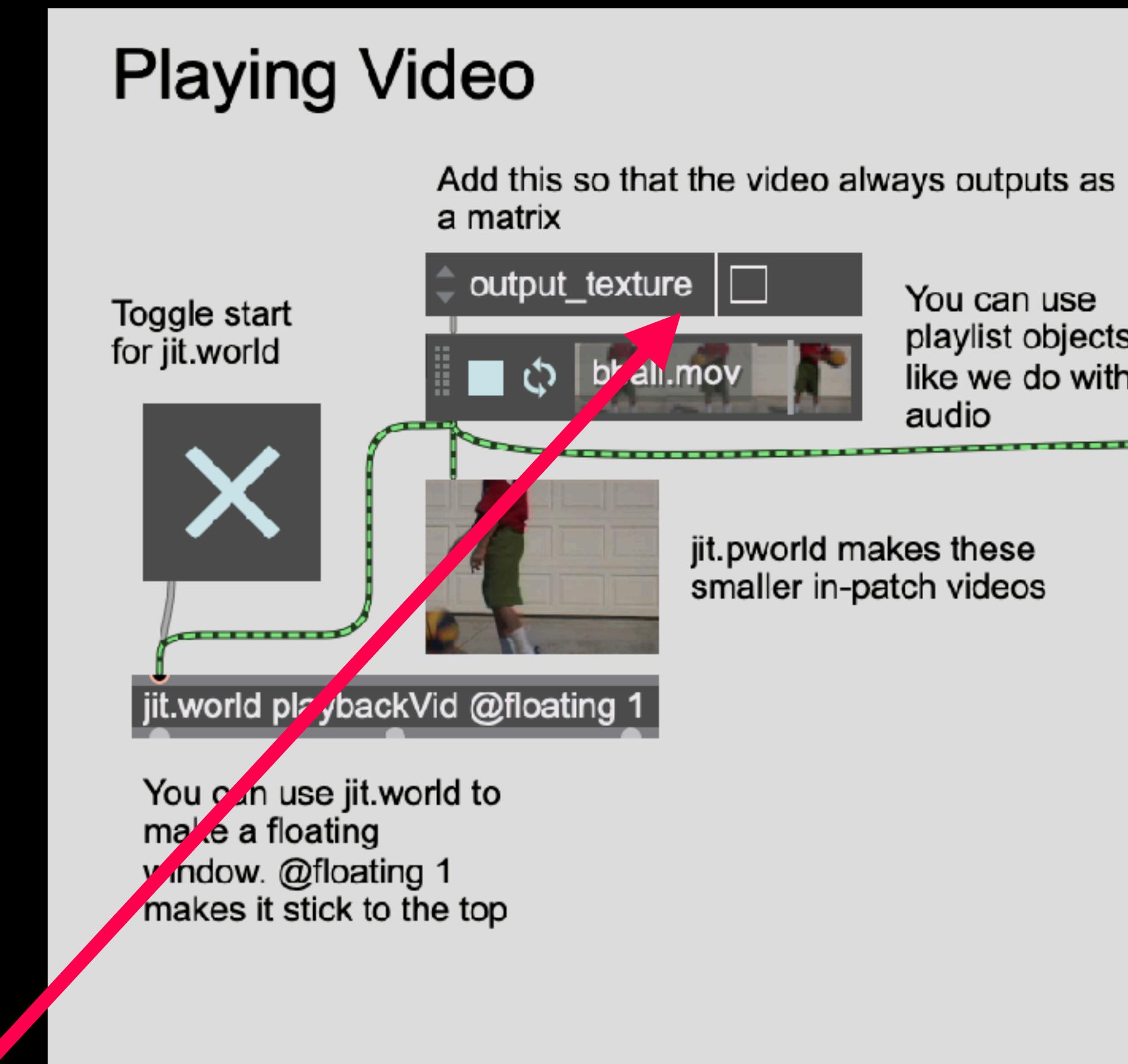
jit.world makes a window to display video. jit.pworld makes a window inside the patch.

jit.world @floating 1 makes sure the window is always on top.

Jit.world needs to be toggled on
You can have multiple worlds with name attributes.

You'll notice green patch cables that indicate video matrix. Blue indicate video texture.

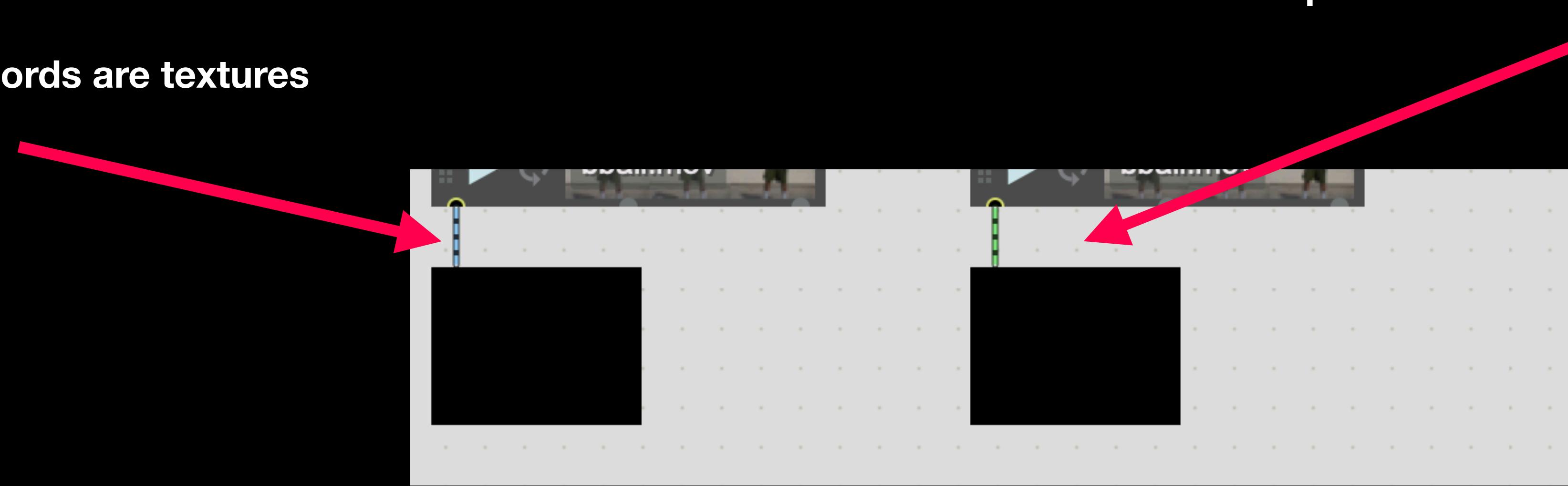
The attrui element for output as texture should be unchecked



Playing Video Patch

Blue patch cords are textures

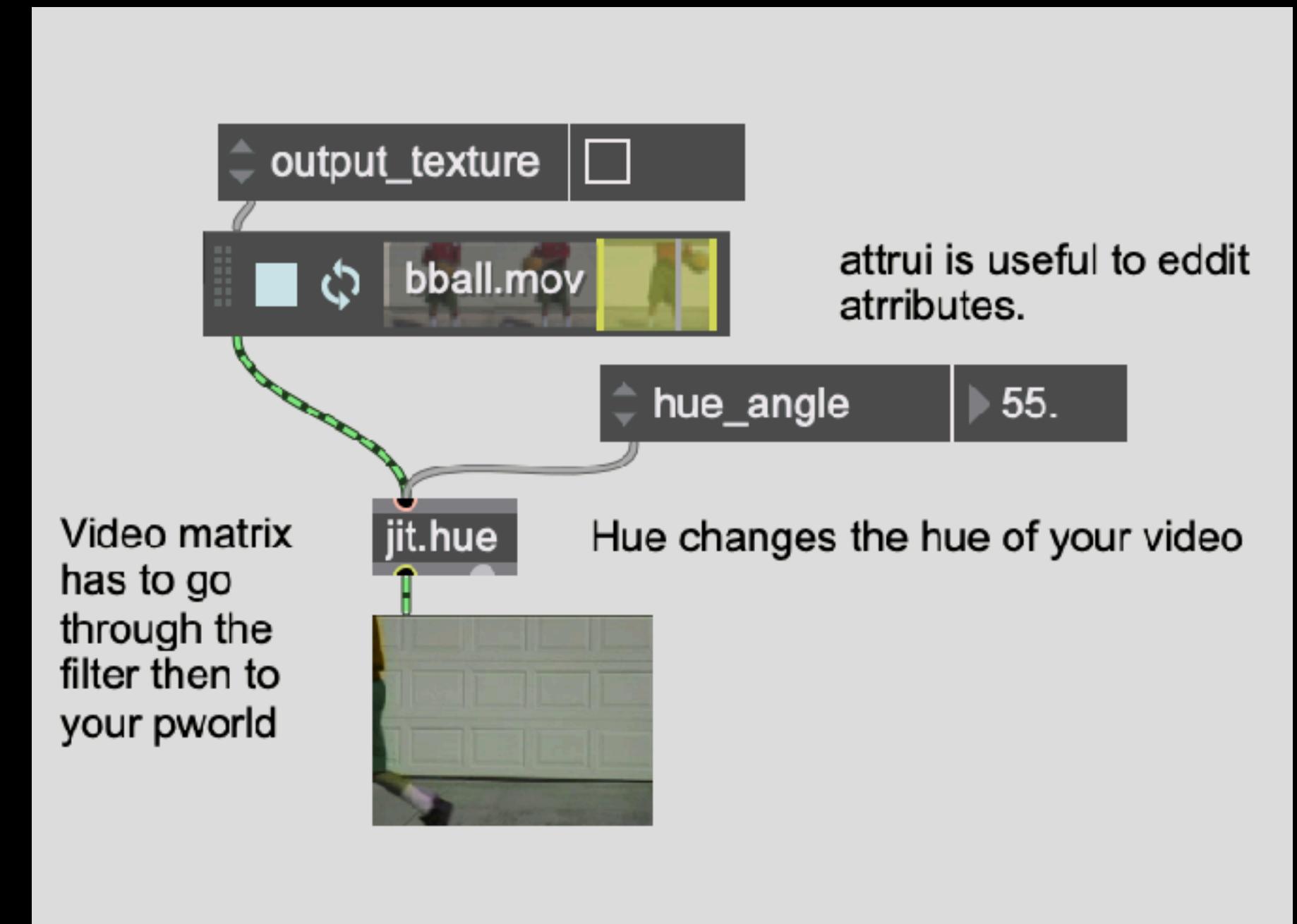
Green patch cords are matrices



Most of the effects we are looking
at today will be done on matrices.

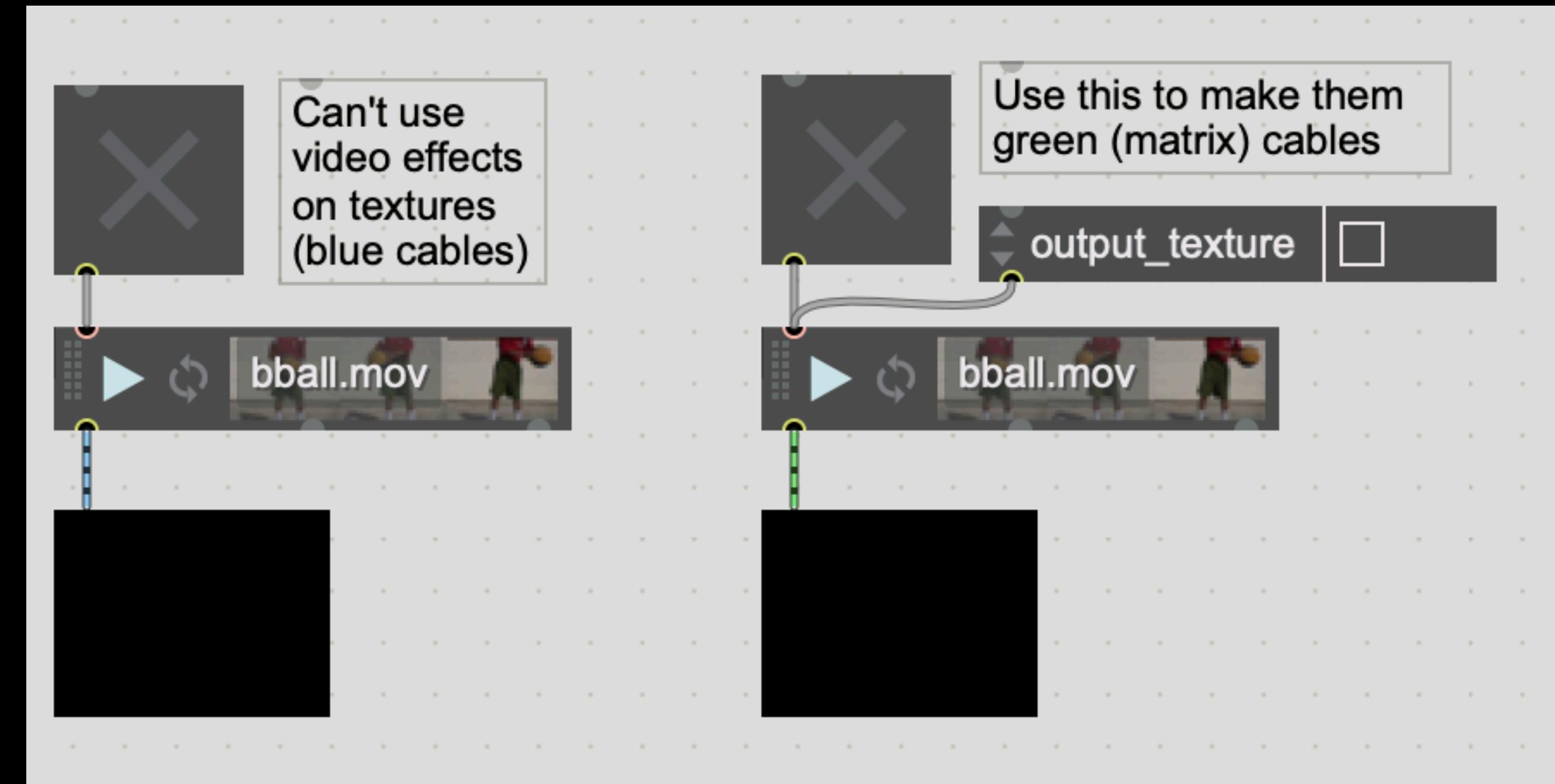
Playing Video Patch

You can apply filters to your video by putting them through a filter object. For example, **jit.hue**, which I can modify with **attrui** elements.



Playing Video Patch

You can use video in the same way you added audio, drag and drop and using the video **playlist** object. You'll need a **window** or **world** to view it and **toggle** to turn it on or off. You can use similar tools to audio to modify them. **HOWEVER!** Sometimes you video will come out as a **texture** (blue patch cord). Use **output** to make them int matrices.

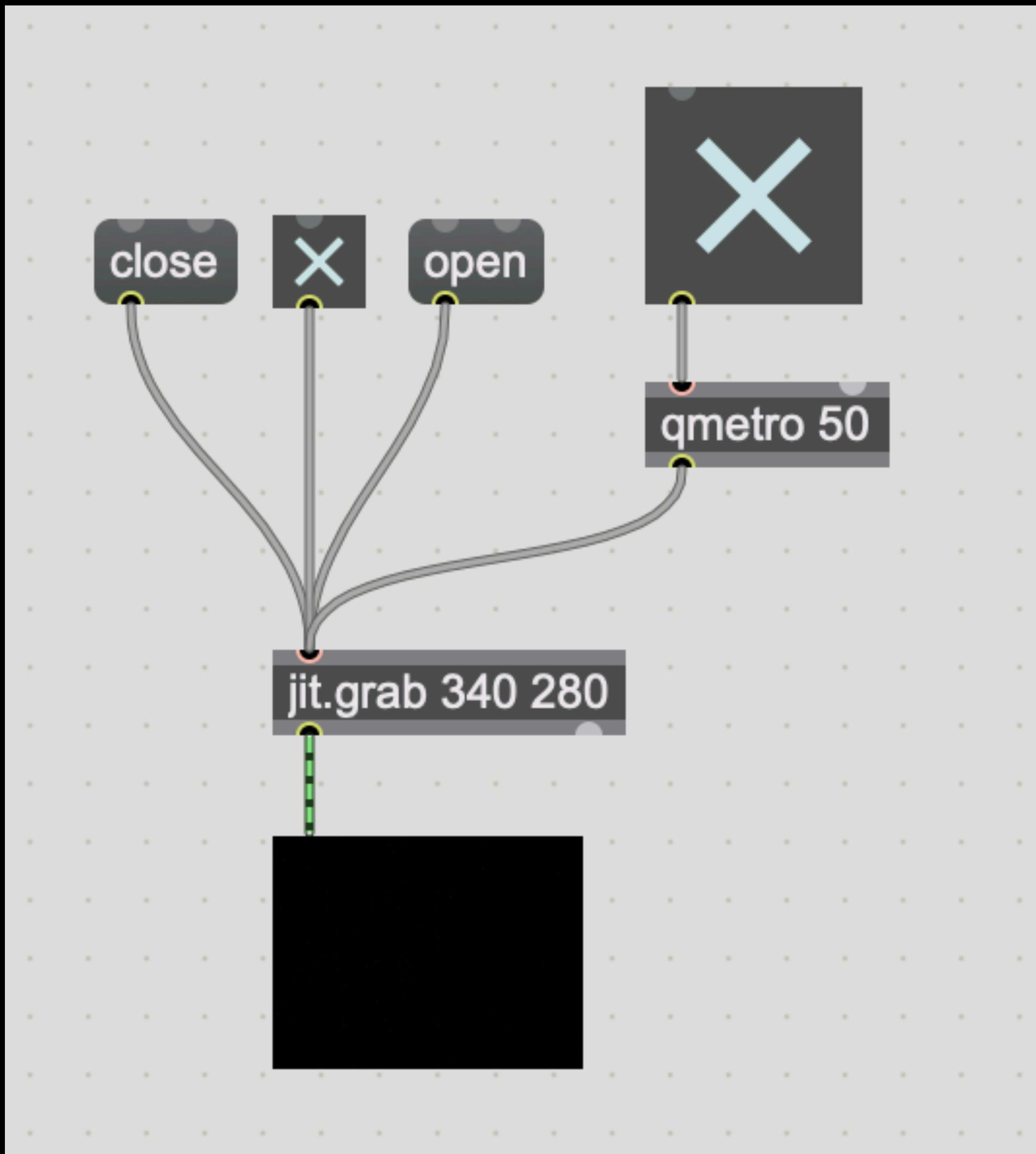


Camera Video Patch

jit.grab can be used to grab from the camera. You must send a message to open the camera.

You'll notice green patch cables that indicate video.

Use qmetro to begin grabbing video. qmetro doesn't lag if you miss a queued time. First open, then toggle the qmetro.

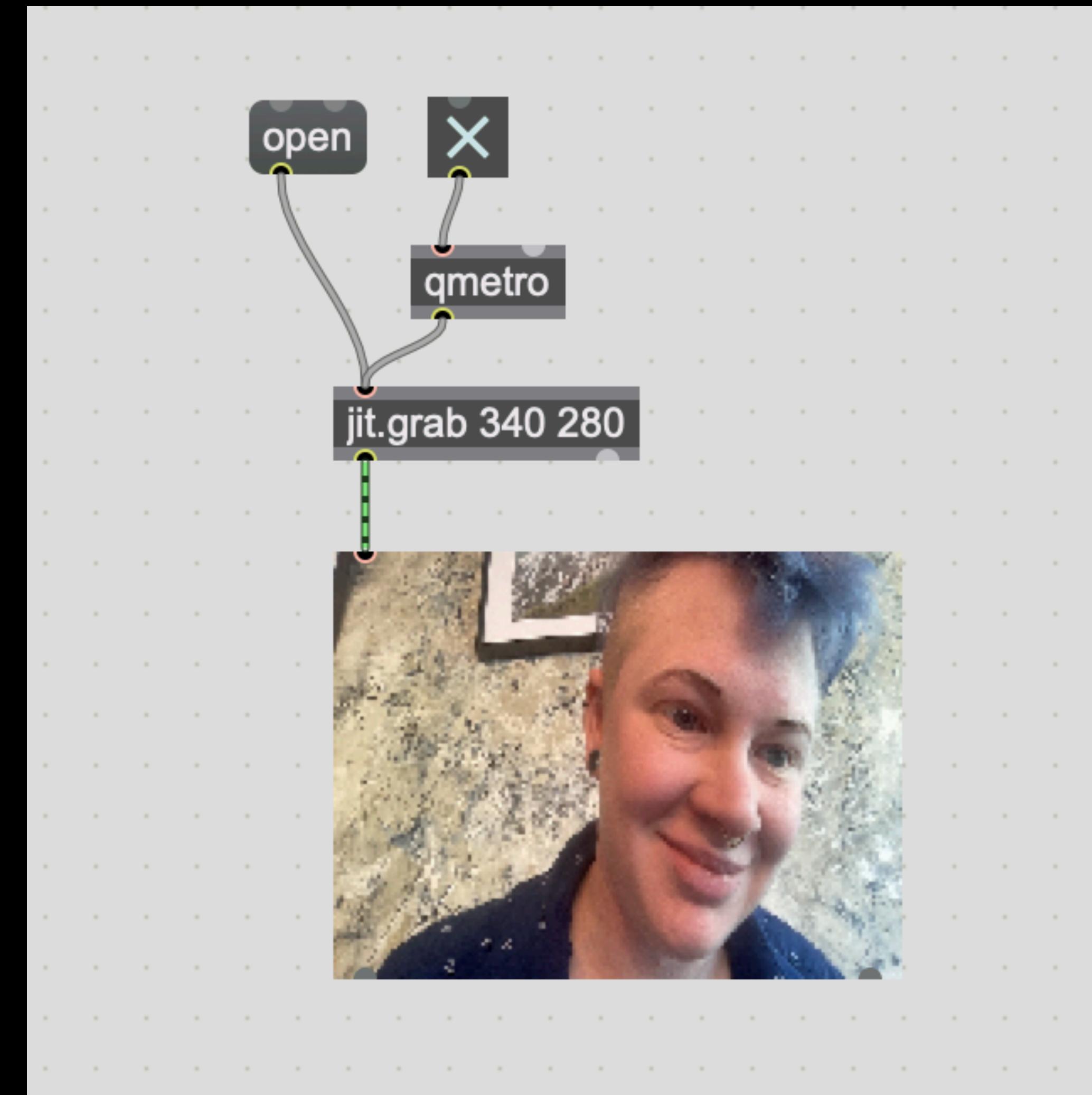


Playing from camera Patch

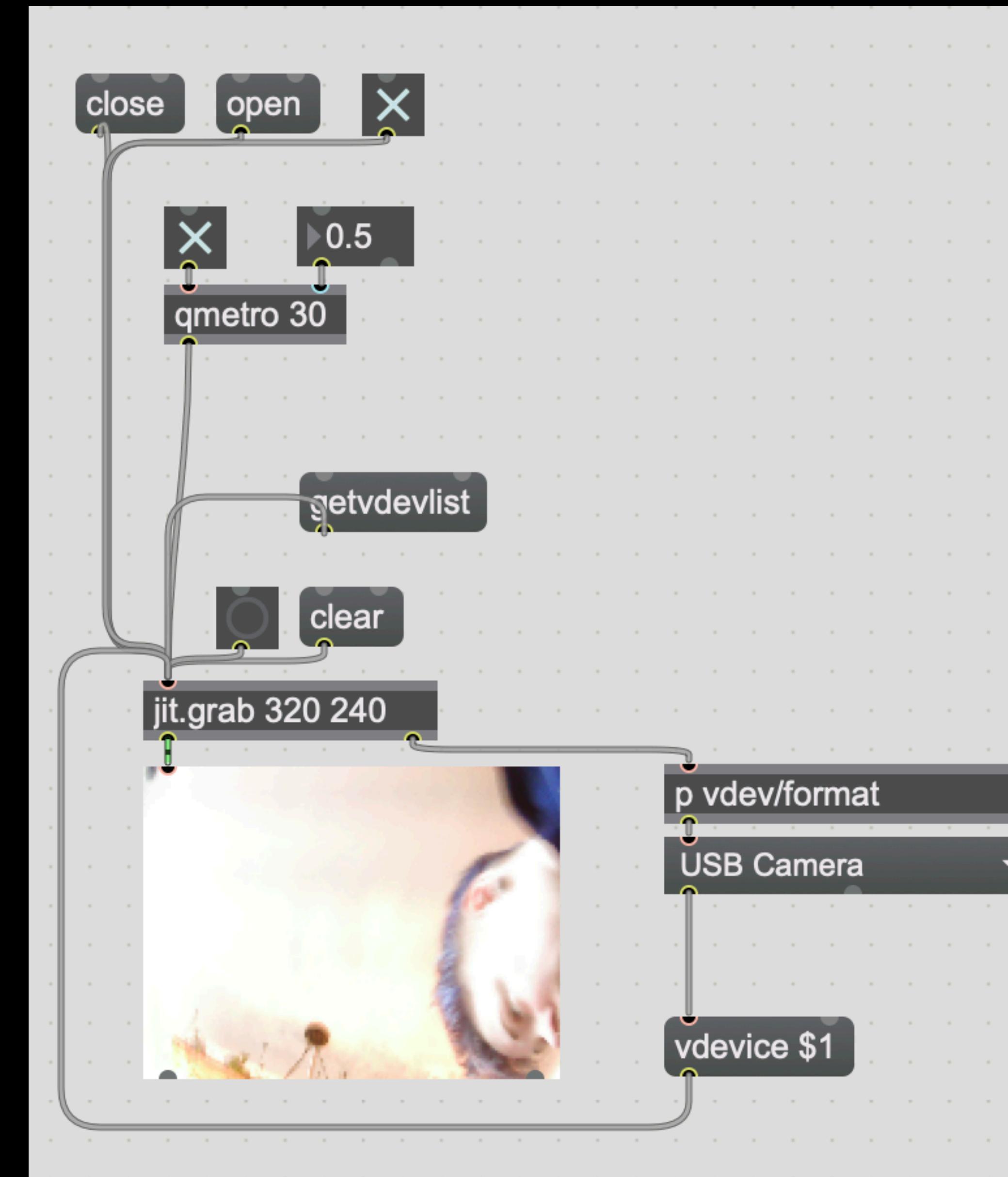
jit.grab can be used to grab from the camera. You must send a message to open the camera.

We use **qmetro** and **toggle** to set the pace. Qmetro is useful because it won't slow if it gets overloaded.

You'll notice green patch cables that indicate video.



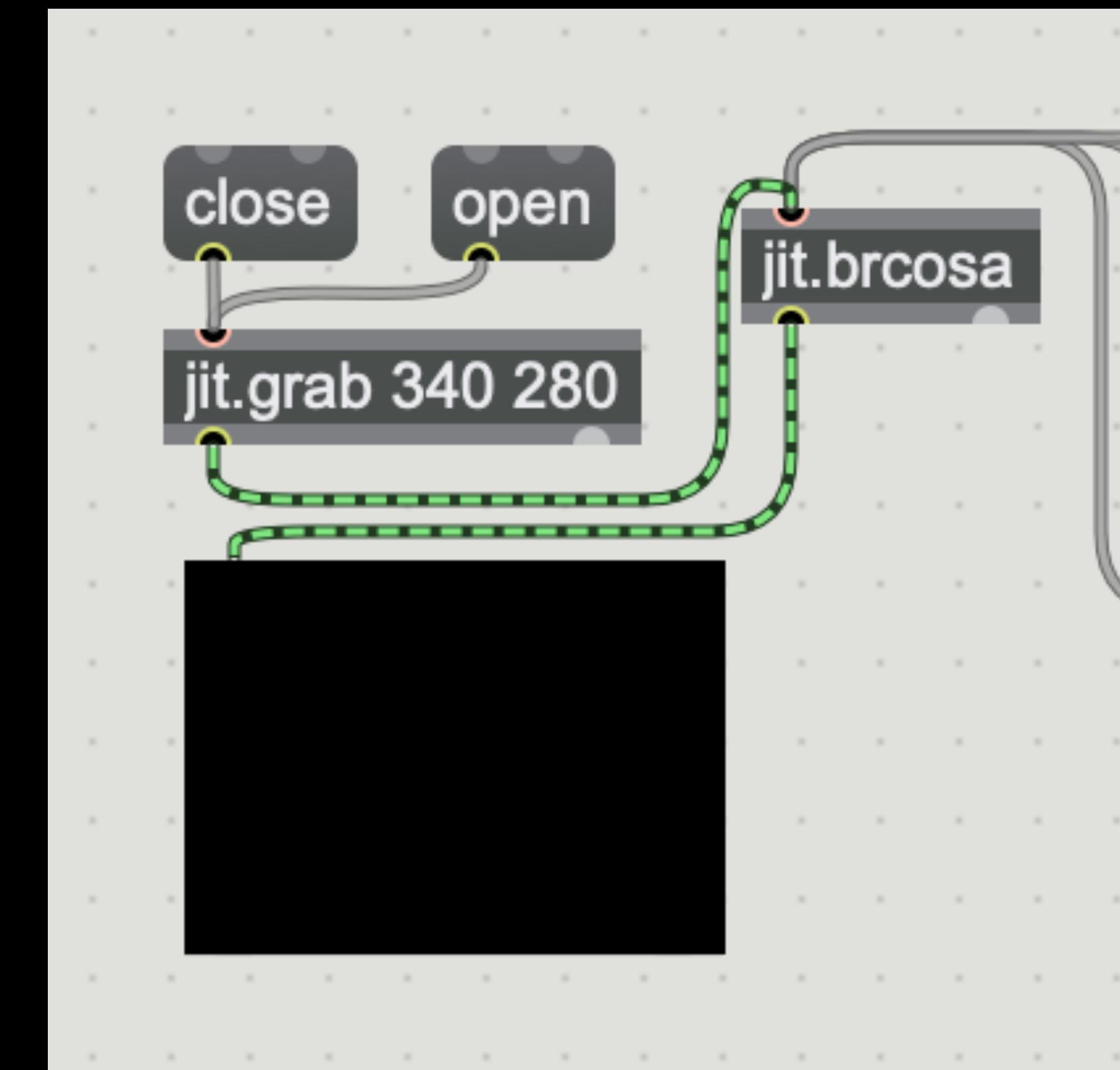
Playing Video Patch
If you are using a
USB camera you may
need to select it using
getvdevlist message.
You can copy this
from the playing
video patch OR from
the jit.grab help.



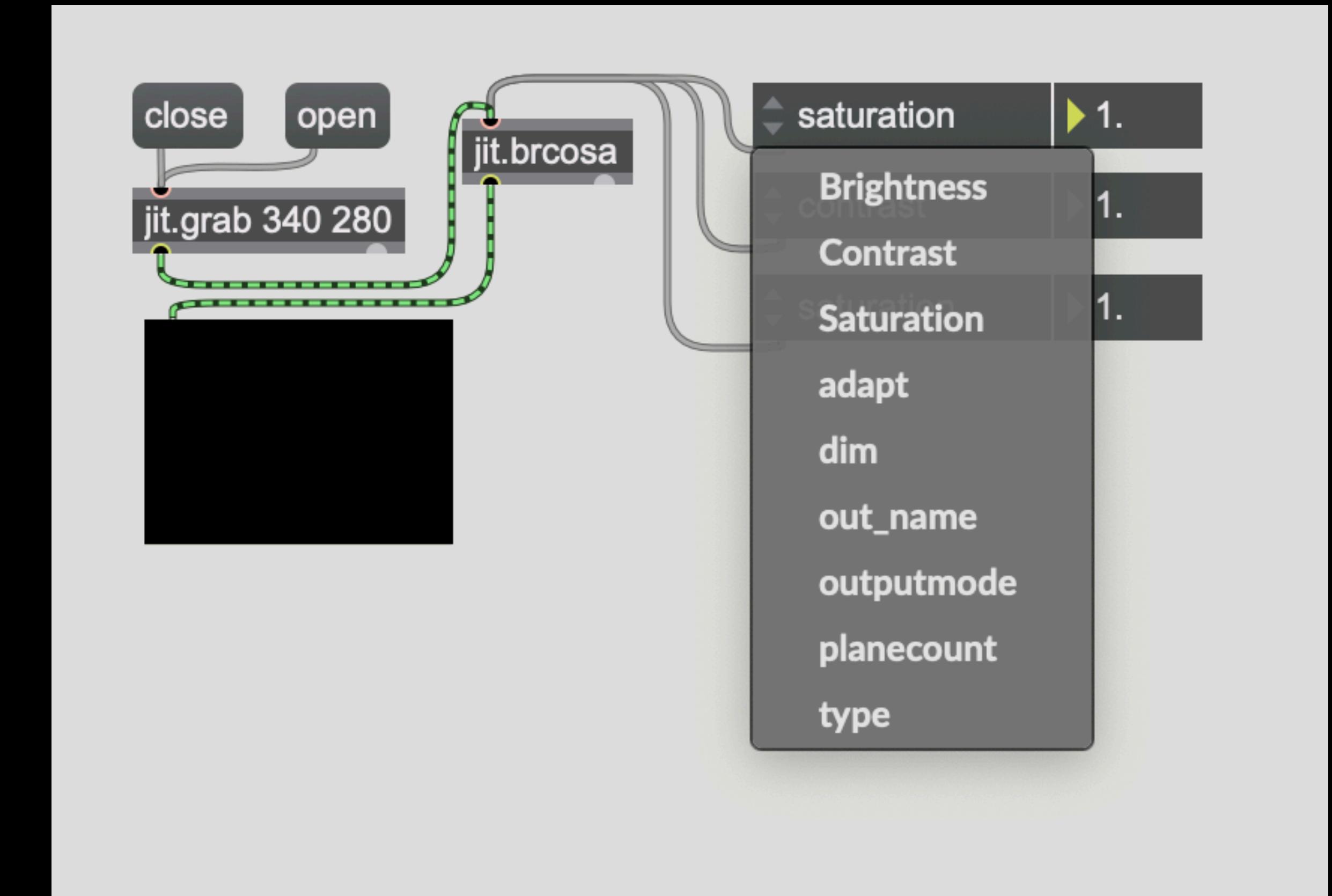
Playing Video Patch

You can edit video
with various filters.

Lets start with
jit.brcosa. This edits
brightness,
saturation, contrast.
The video should go
through **jit.brcosa**.



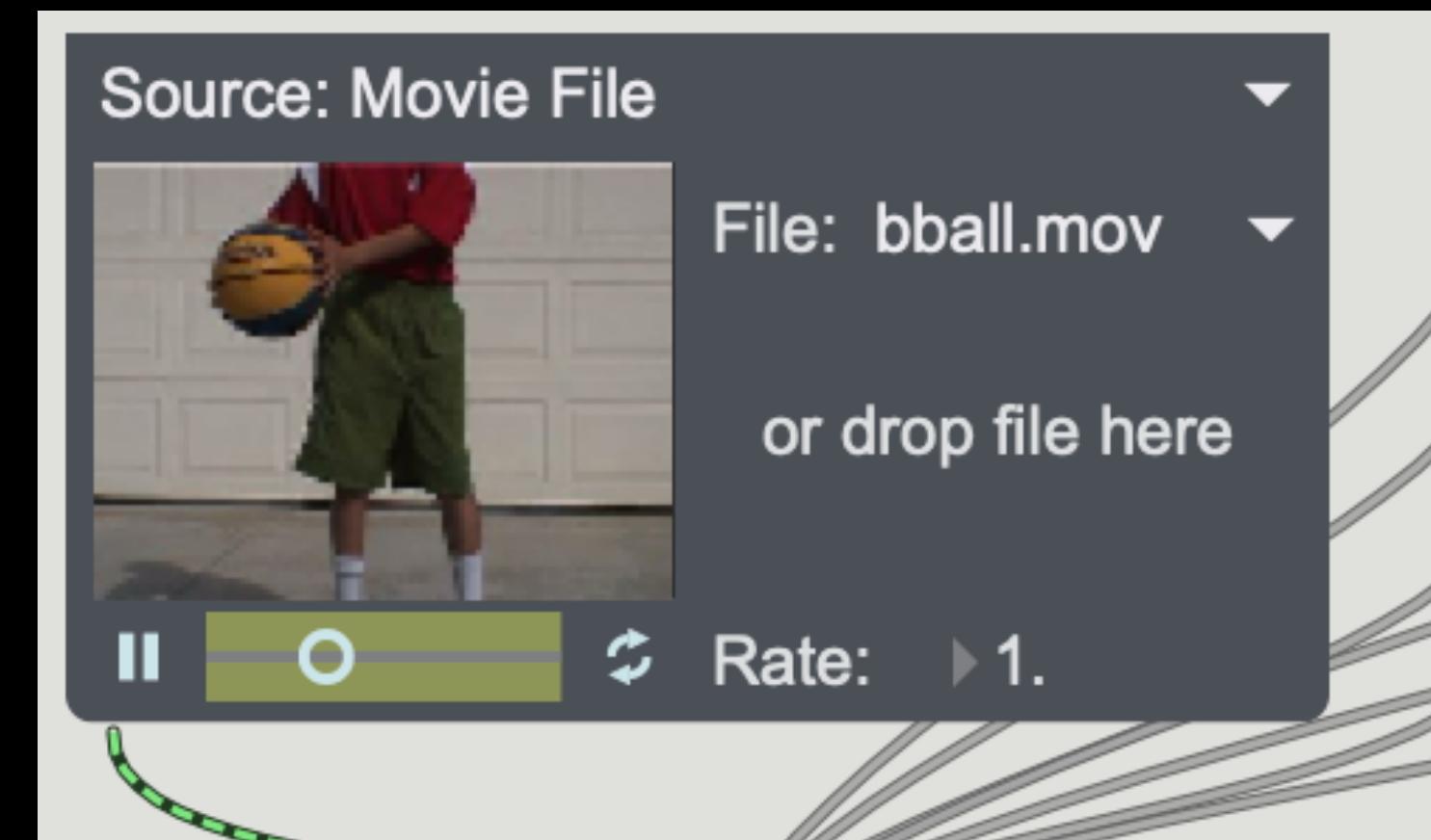
Playing Video Patch
Attrui is an element
(created with shortcut a)
that lets you access
attributes through a simple
UI element. It is easiest to
understand in relation to
video. Here we can see it
being used to change
elements in jit.brcosa



Playing Video Patch

You can see a list of Jitter objects to explore here https://docs.cycling74.com/max8/vignettes/jitter_functional

Examples in help use objects that look like this. These are not single objects, although they look like it. They are a combination of objects in a sub patch. You can right click, unlock and explore the sub patch.

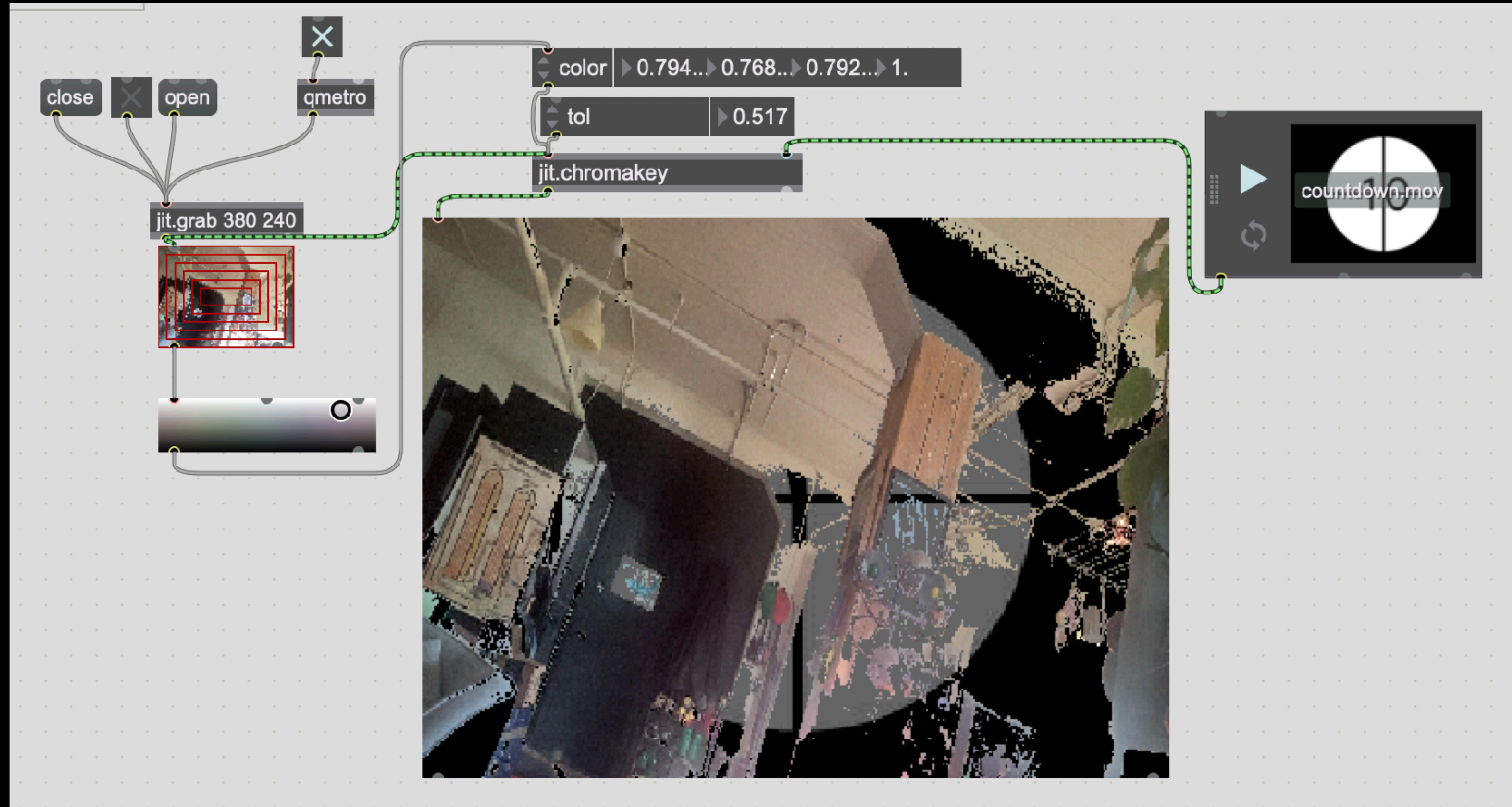


Resources

What can jitter do? Lets explore some filters together.

<https://cycling74.com/articles/jitter-resources-an-overview>

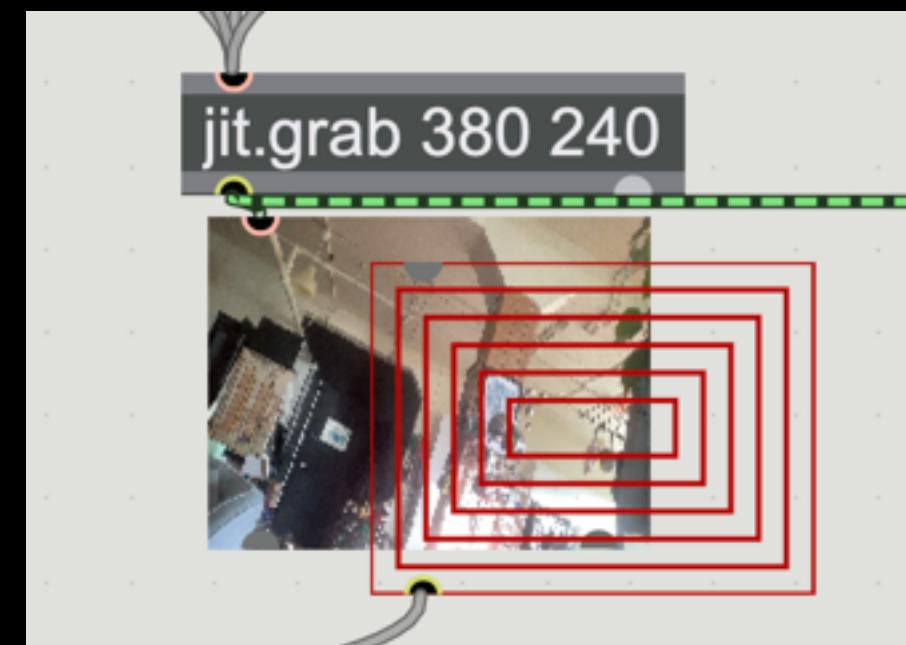
https://docs.cycling74.com/max8/vignettes/jitter_functional



Chromakey patch

This is like green screen. You can use live or pre recorded video to mix.

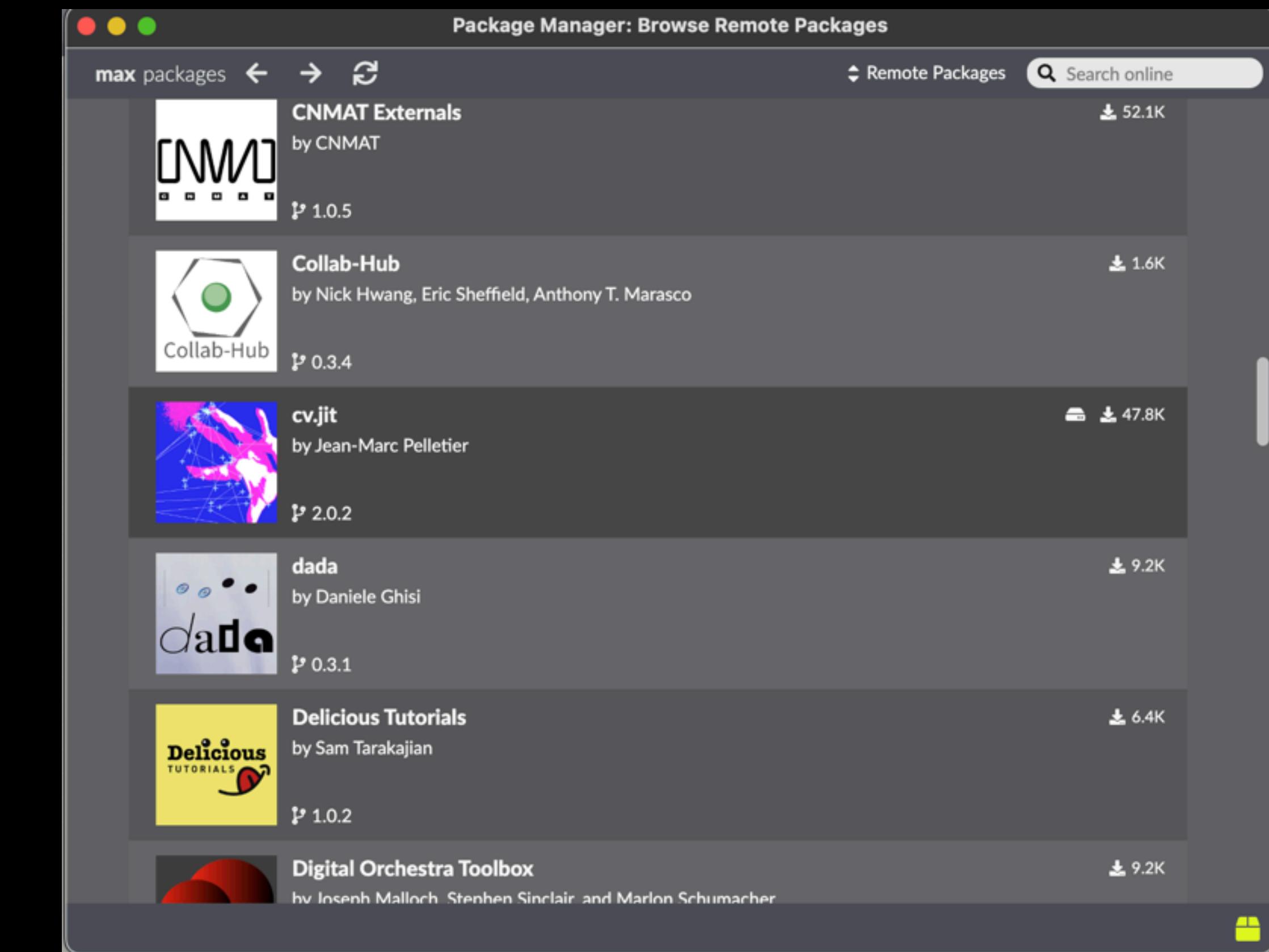
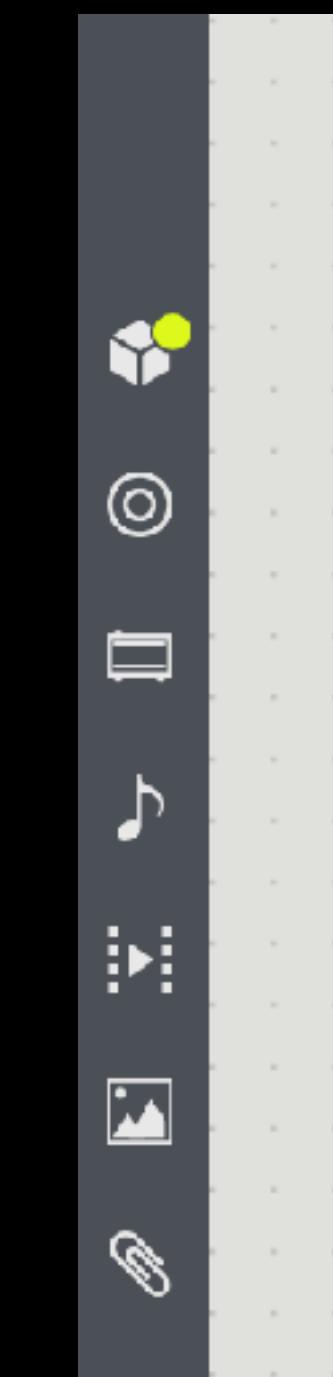
We use **suckah** to select a color you want to chromakey. Note that suckah is placed over the video feed. You can also use **swatch**. In this example we are mixing a web cam stream with a recorded video. You can adjust the tolerance using **attrui**.



Face Tracking Patch

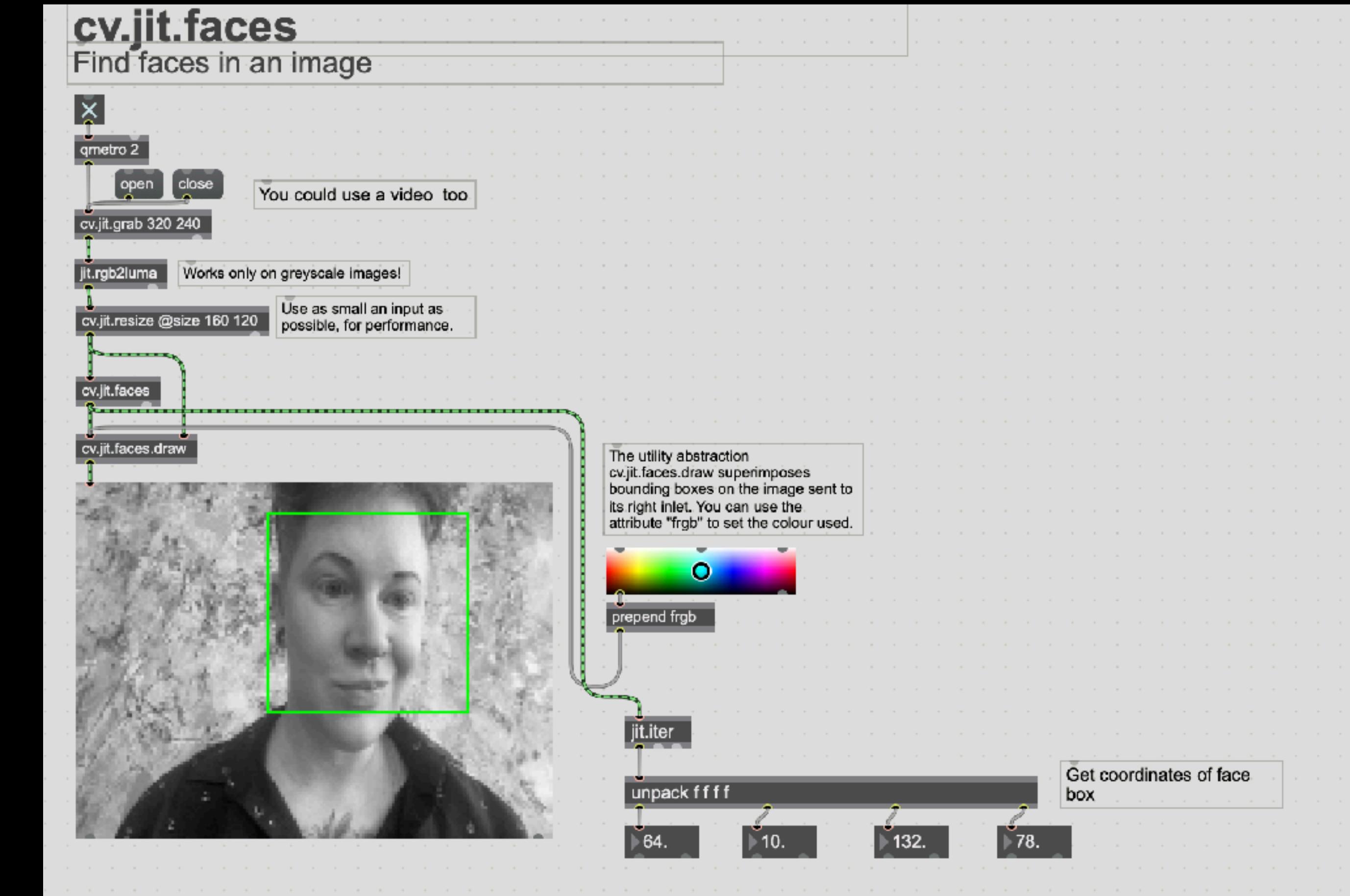
Face tracking uses a package called CVjit which stands for computer vision. You can find a ton of other patches in the package manager on the left hand side. Select and install cv.jit.

You can see everything it can do here: <https://jmpelletier.com/cvjit/>



Face Tracking Patch

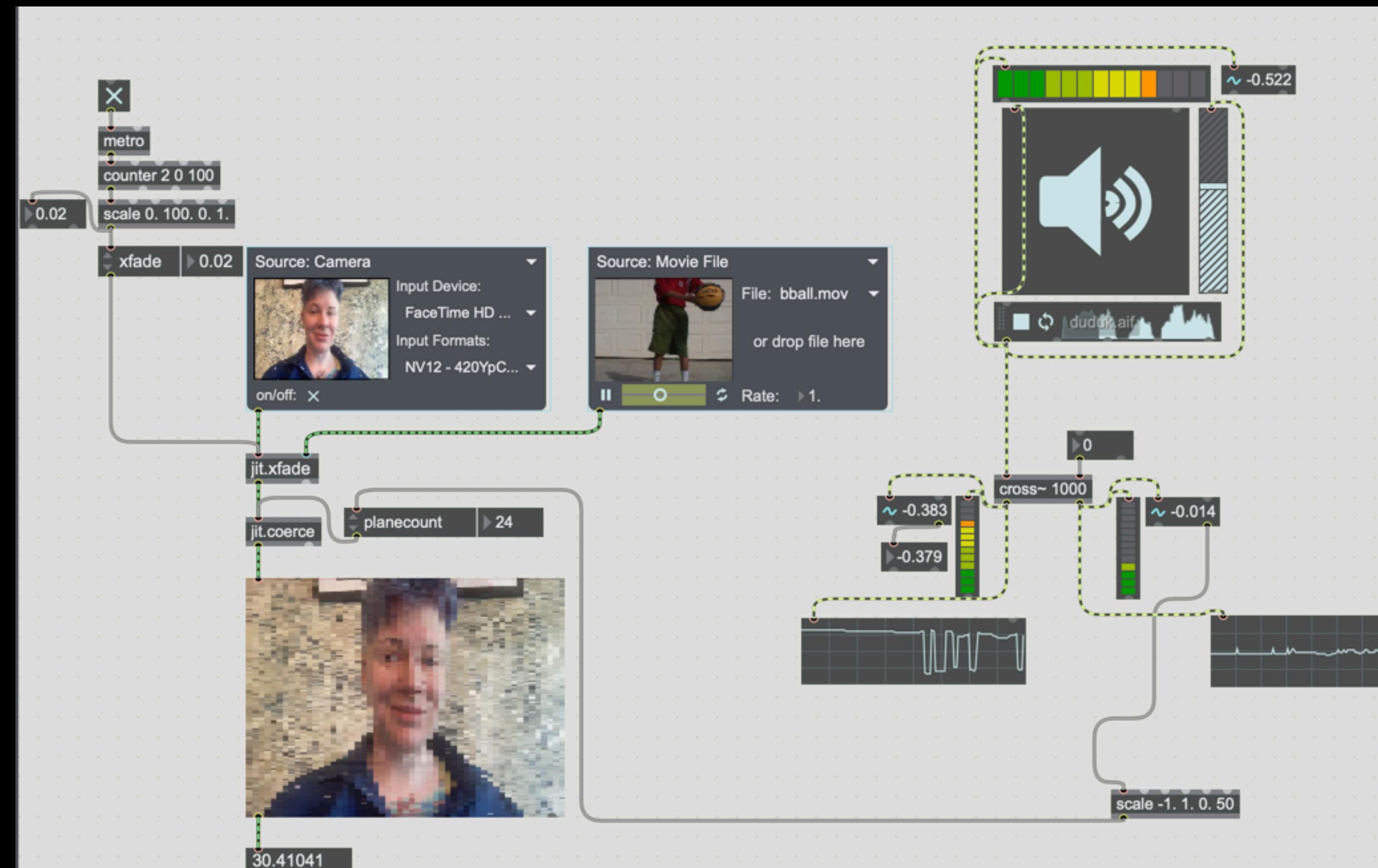
This is an example of using face tracking. You can output then positions of the box drawn around the face



Audio Reaction Patch

Cross~ is a very simple way to make your video audio reactive.

Cross~ takes signals from audio above and below a certain frequency. You can chain corss~ to make more complex filtering. If you want to use low pass, high pass, or other complex forms of audio filtering use biquad. This patch uses xFade and Coerce. You can see the two audio signals being split in the scope.



Various tracking & Mapping

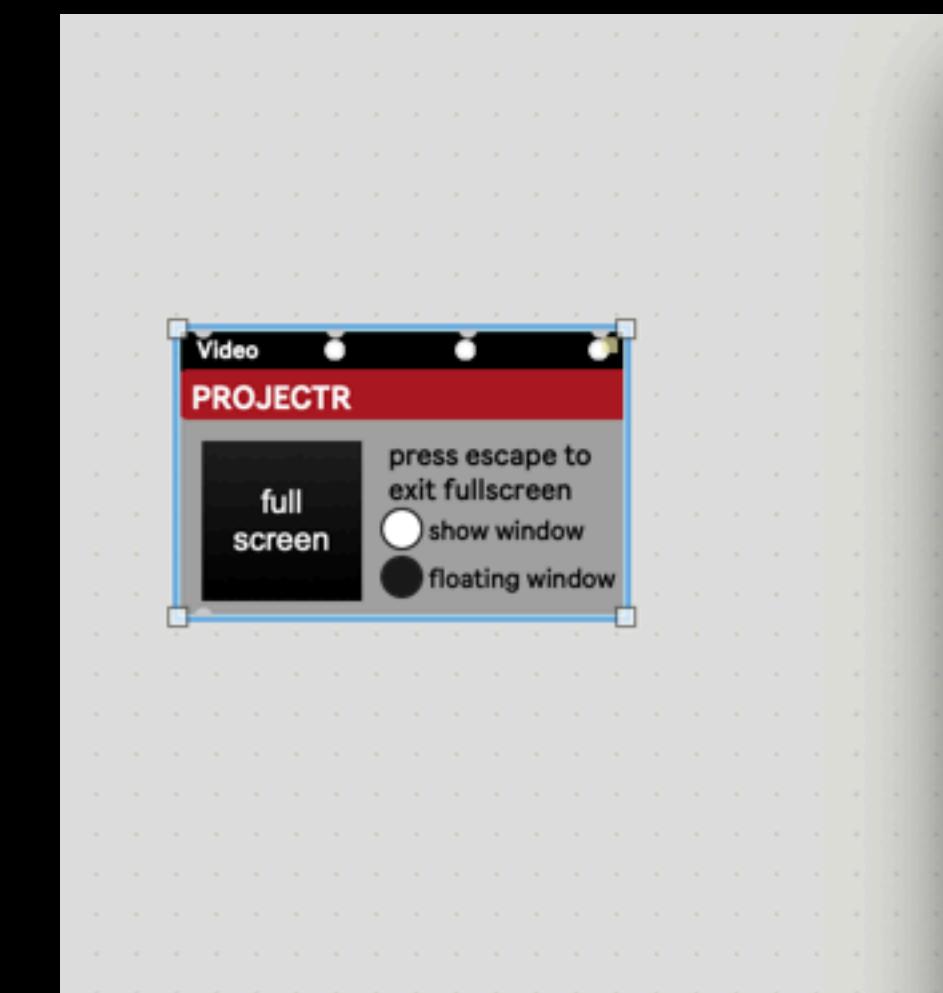
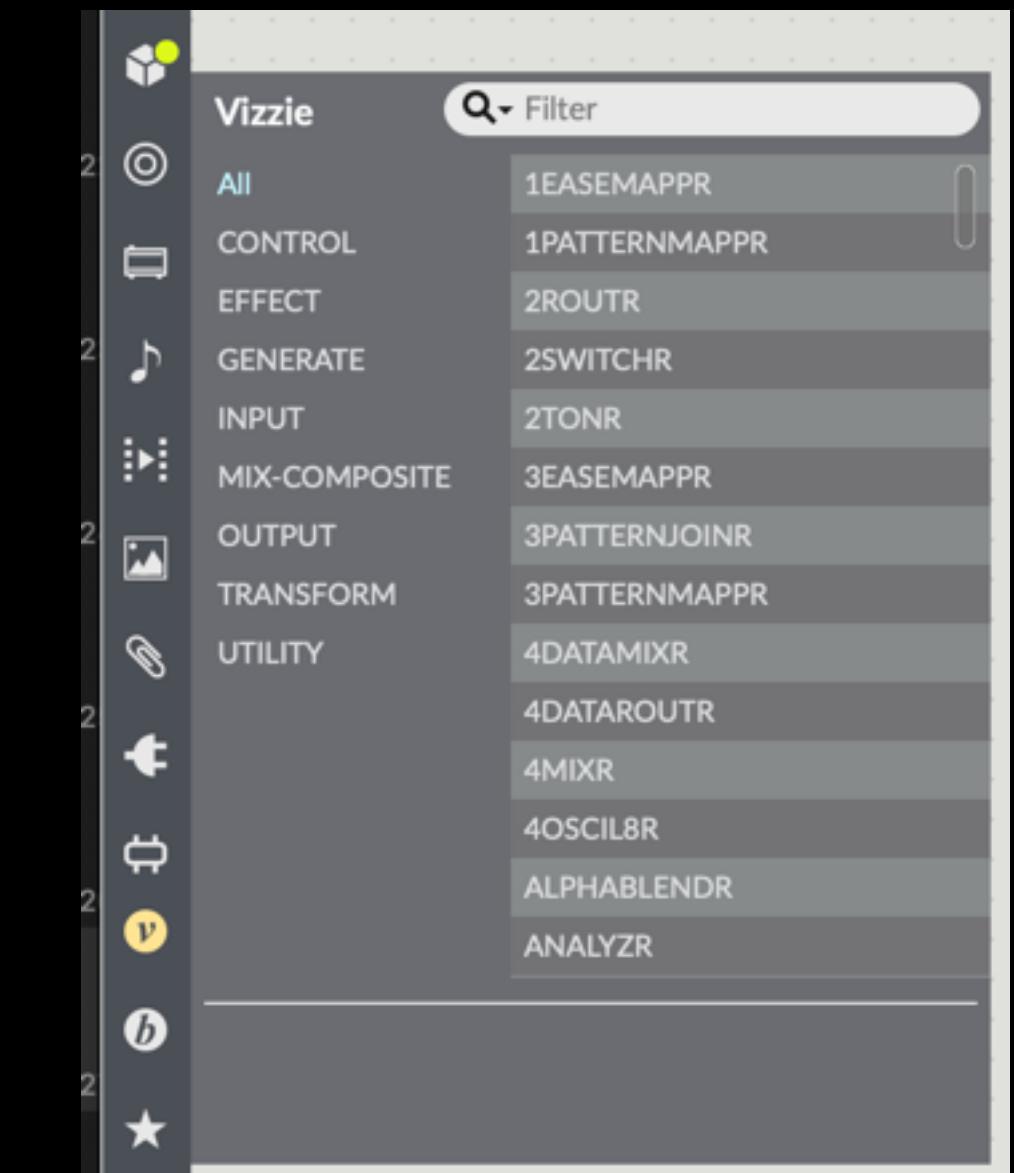
Color Tracking <https://docs.cycling74.com/max8/tutorials/jitterchapter25> Follow a color on a video feed or screen

Blob Tracking <https://www.youtube.com/watch?v=lNyTVvoO4bo> Follow a shape on a feed or screen

Projection Mapping https://www.youtube.com/watch?v=rz8g2ax_934 Project onto a specific surface

Vizzie

Vizzie is an extension for using pre made video objects, it can be both useful and limiting! Find it on the left hand side bar and explore.



Some things to explore:

- Drawing in openGL
- Mixing video
- Explore max patches and Open CV
- Audio Reactive video
- Integrating tracking into installation
- Projection and colours in interactive spaces
- Using video to change or manipulate spaces
- Jitter / Makey Makey / Midi / Live music / Soundscapes

- Placement of cameras in space

Handing in projects

Each project should include a .zip file that contains:

- A saved file (File > Save as Project) .maxpatch
- A screen capture of your max patch
- A video or audio recording of your project working (can be a video or a link to a private video on YouTube, Vimeo or Google)
- A 50-100 word explanation of your project inside your maxpatch

Be sure to name files properly (no untitled-1.zip)

All files are expected to be cleaned up and arranged in a reasonable, legible way. Videos should be clear, well light and show your project working.

Homework:

Work on Video Experiment (due next week after studio session)

Meet teams for Mid Term

Next week: Drawing in OpenGL and open studio for video experiment