

Slit-scan photography

Fotografiska experiment i Processing

“Fotografisk teknik för
att frysa händelser över
tid i en statisk bild”



Tidigt

Senare



Foto av Andrew Davidhazy



Foto av Henri Lartigue

Hur gör man?



Foto av Stefan Marjoram: <http://flickr.com/photos/stefanmarjoram/1444823984/>



Foto av Stefan Marjoram: <http://flickr.com/photos/stefanmarjoram/1444823984/>

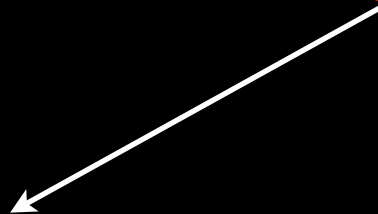
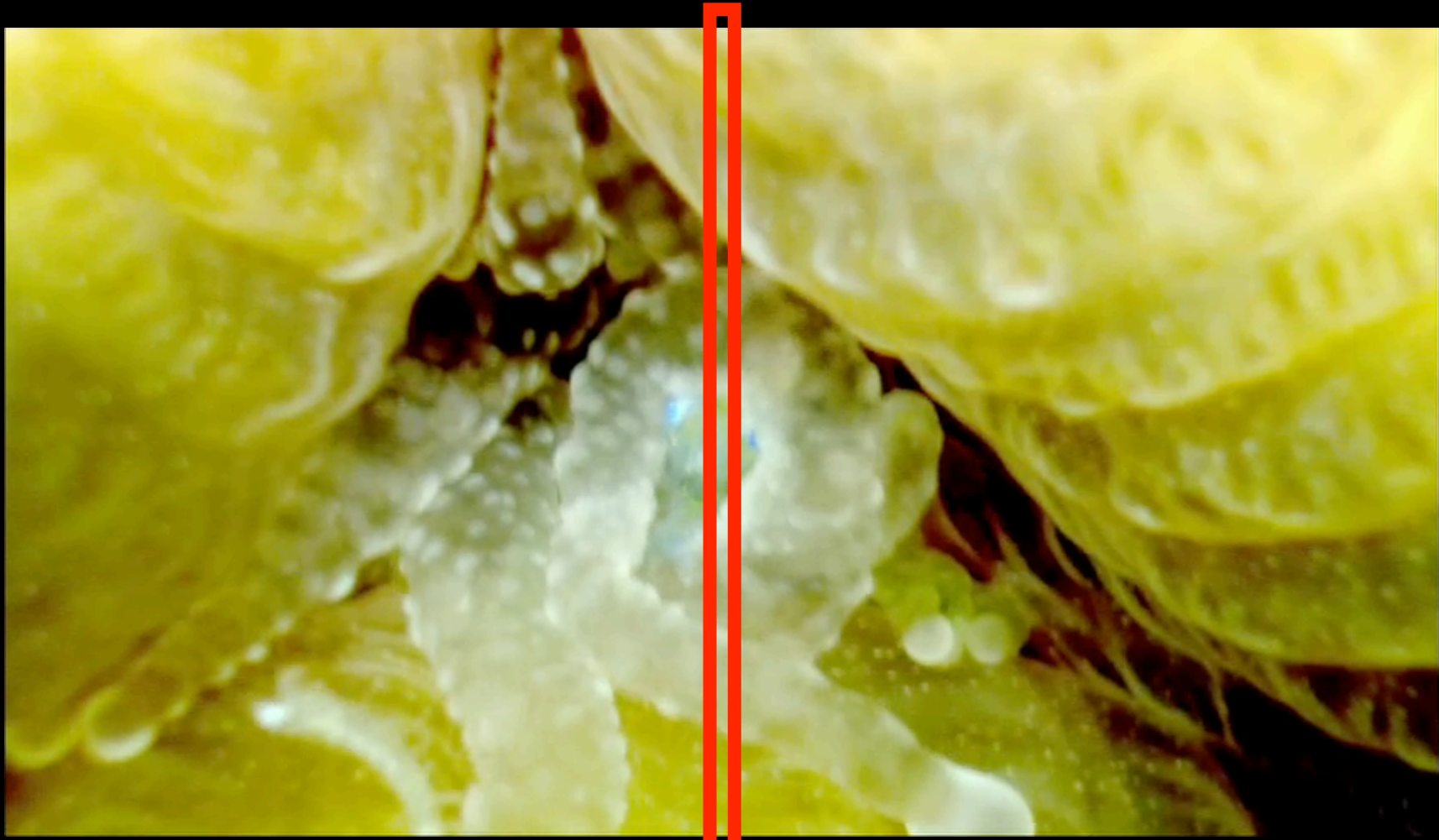


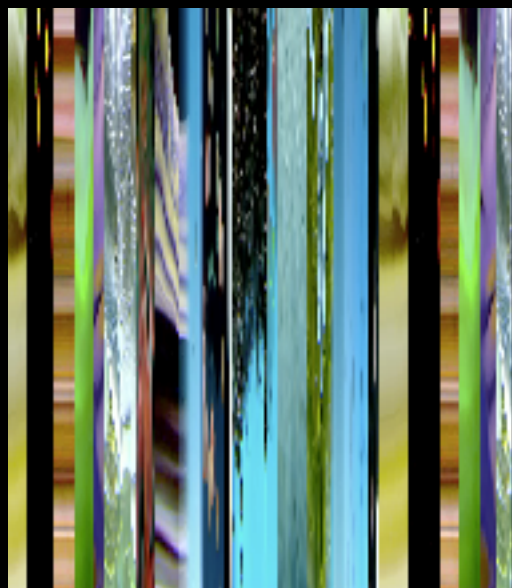
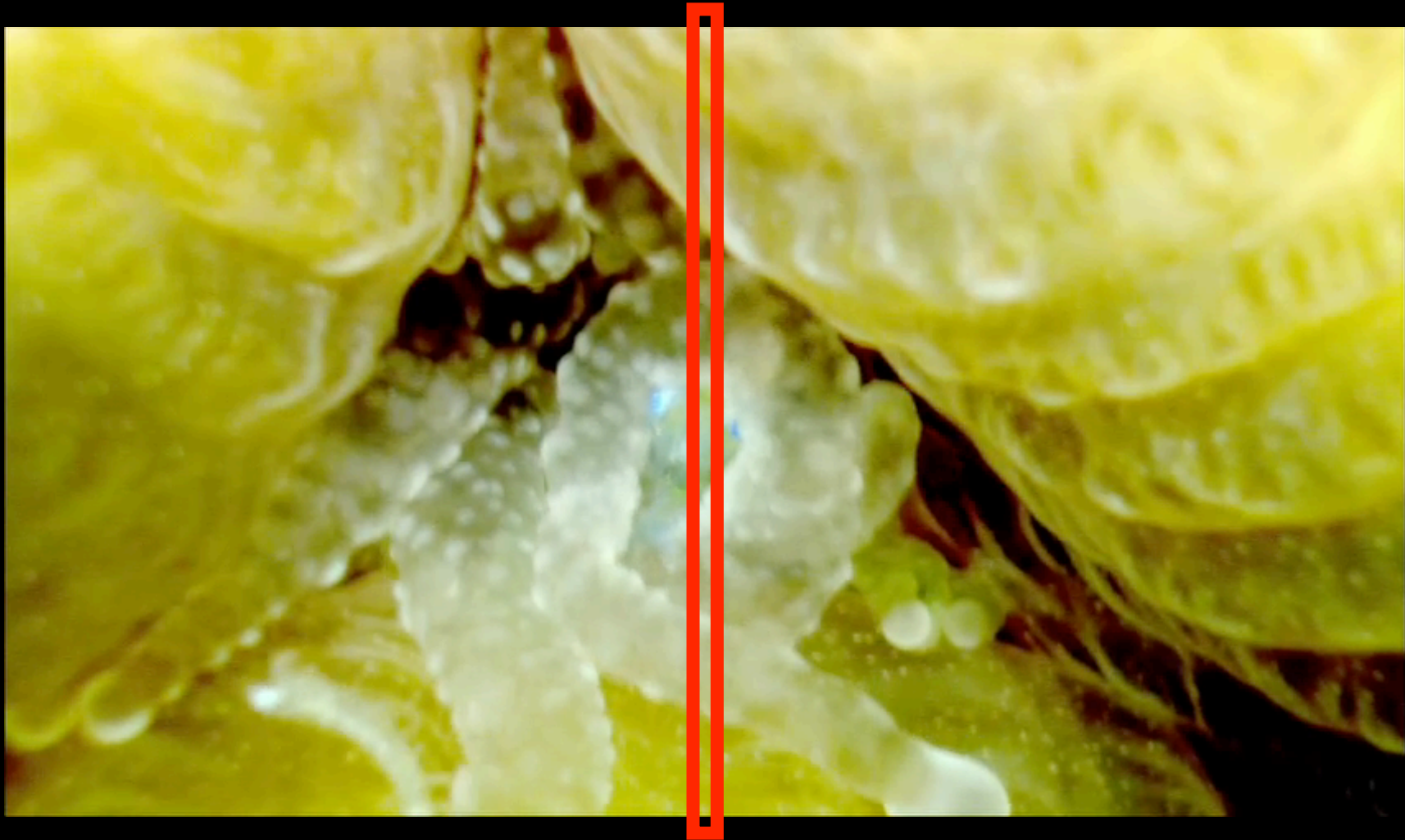
Foto av Richard: <http://flickr.com/photos/richard314159/1861997941/>





...eller videofeed + kod!





processing.org



Exhibition



[Body Navigation](#)

by Jonas Jongejan and Ole Kristensen
for Recoil Performance Group



[News Knitter](#)

by Ebru Kurbak & Mahir M. Yavuz

Books

» Download Processing

Processing is an open source programming language and environment for people who want to program images, animation, and interactions. It is used by students, artists, designers, researchers, and hobbyists for learning, prototyping, and production. It is created to teach fundamentals of computer programming within a visual context and to serve as a software sketchbook and professional production tool. Processing is an alternative to proprietary software tools in the same domain.

Processing is free to [download](#) and available for GNU/Linux, Mac OS X, and Windows. [Please help to release the next version!](#)

Processing is an open project initiated by [Ben Fry](#) and [Casey Reas](#). It evolved from ideas explored in the Aesthetics and Computation Group at the MIT Media Lab.

Please visit our subdomains and sister projects:

<http://dev.processing.org/>

Includes bug tracking and instructions for building the code, downloading the source, and creating libraries.

<http://hardware.processing.org/>

Archiving and Mining our physical computing initiatives related to

Java (subset)

slitscan2 §

```
import processing.video.*;

Capture cameraSource;
int capwidth = 1280;
int capheight = 1024;
int outwidth = 1024;

int currentColumn = 0;

boolean keepDrawing = false;
boolean isDone = false;

void setup()
{
  size(outwidth, capheight);
  cameraSource = new Capture(this, capwidth, capheight, 30);
  cameraSource.crop(capwidth/2, 0, 1, capheight);
}

void captureEvent(Capture cameraSource) {
  cameraSource.read();
}

void keyPressed() {
  keepDrawing = true;
  isDone = false;
}

>
void draw() {
  if(keepDrawing) {
    if(currentColumn<=width) {
      image(cameraSource, currentColumn, 0);
      currentColumn++;
    }
    else {
      keepDrawing = false;
      isDone = true;
    }
  }
}
```

Done Saving.

```
void setup()  
{  
    size(outwidth, capheight);  
    cameraSource = new  
        Capture(this, capwidth,  
            capheight, 30);  
  
    cameraSource.crop(capwidth/2,  
        0, 1, capheight);  
}
```

```
void captureEvent(Capture...
```






www.peterkrantz.com