CNAM - NSY116 - janvier 2015

Captation visuelle pour l'IHM

Pierre Cubaud
cubaud @ cnam.fr

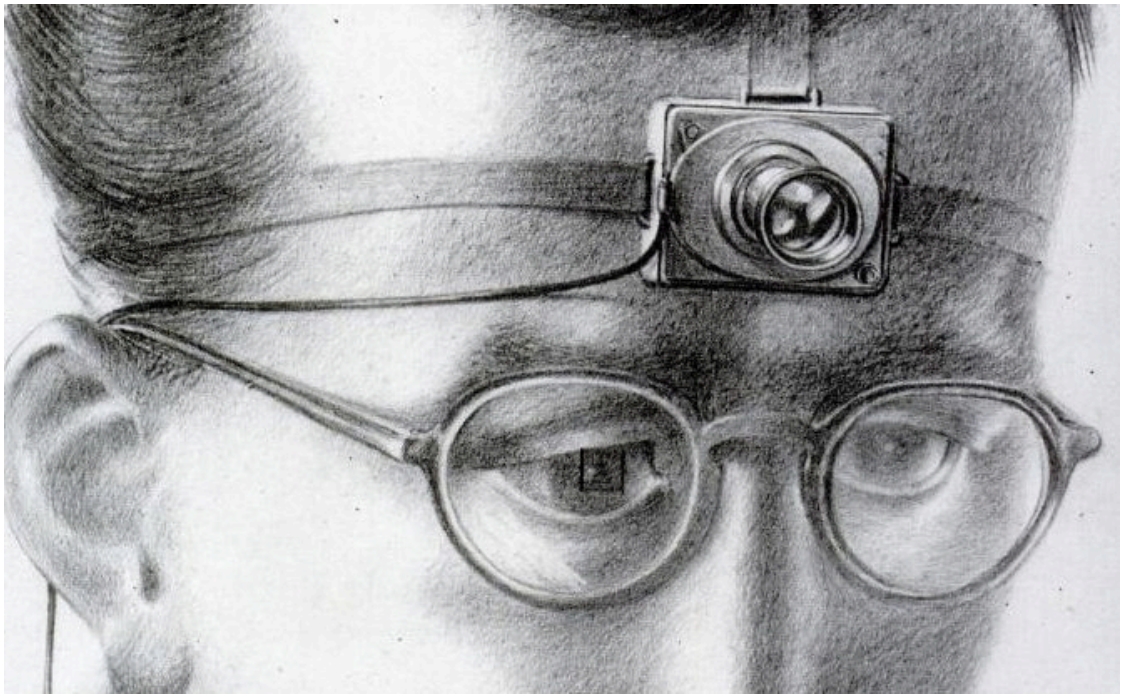
1) captation et filtrage
2) opencv
3) kinect
4) leapmotion

ce qu'on capte pour l'IHM :

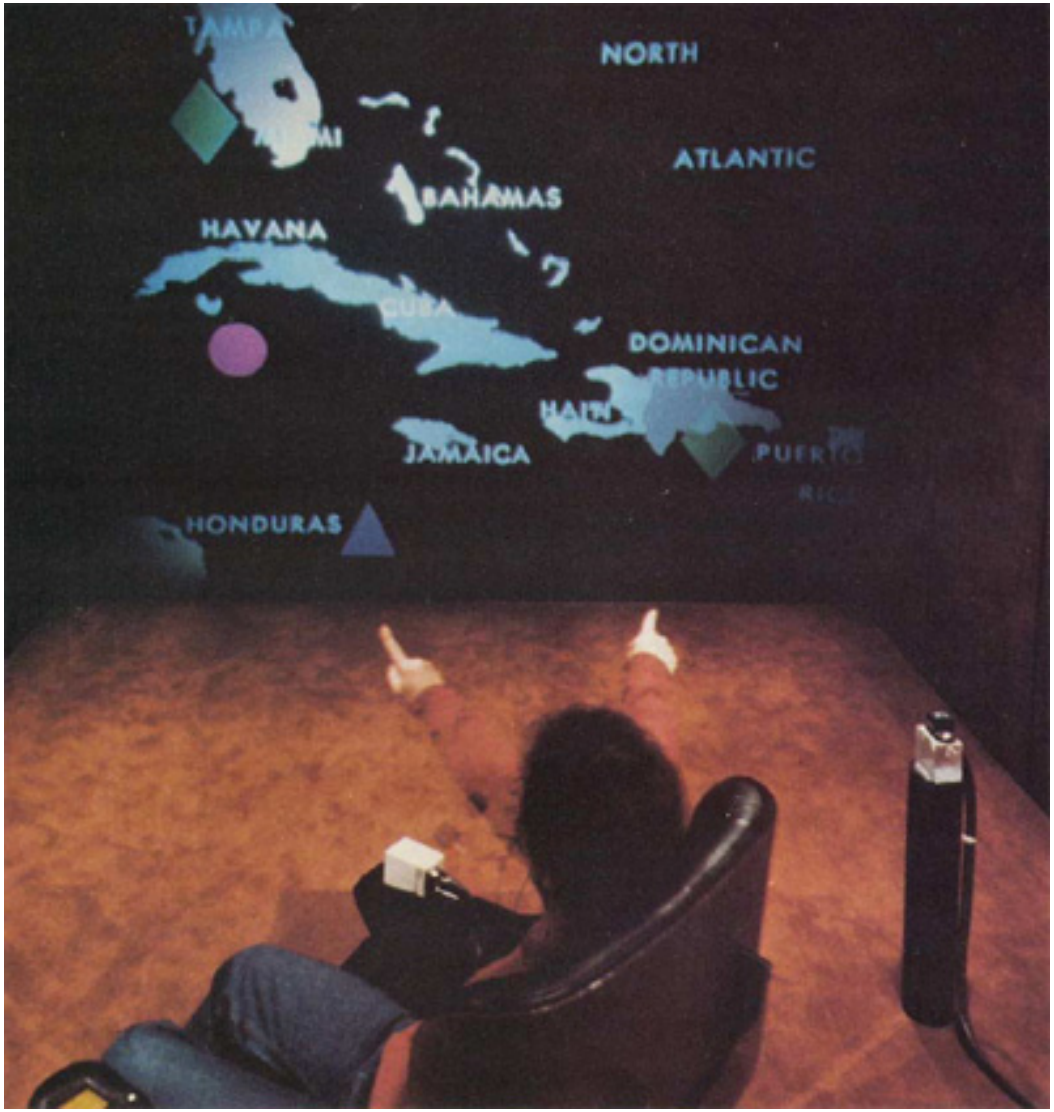
- position
- mouvement, déplacement
- geste

- voix : parole, chant
- attitude
- regard

- physiologie (temperature, rythme cardiaque, sudation ...)
- ondes cérébrales



V. Bush. "As we may think" 1945



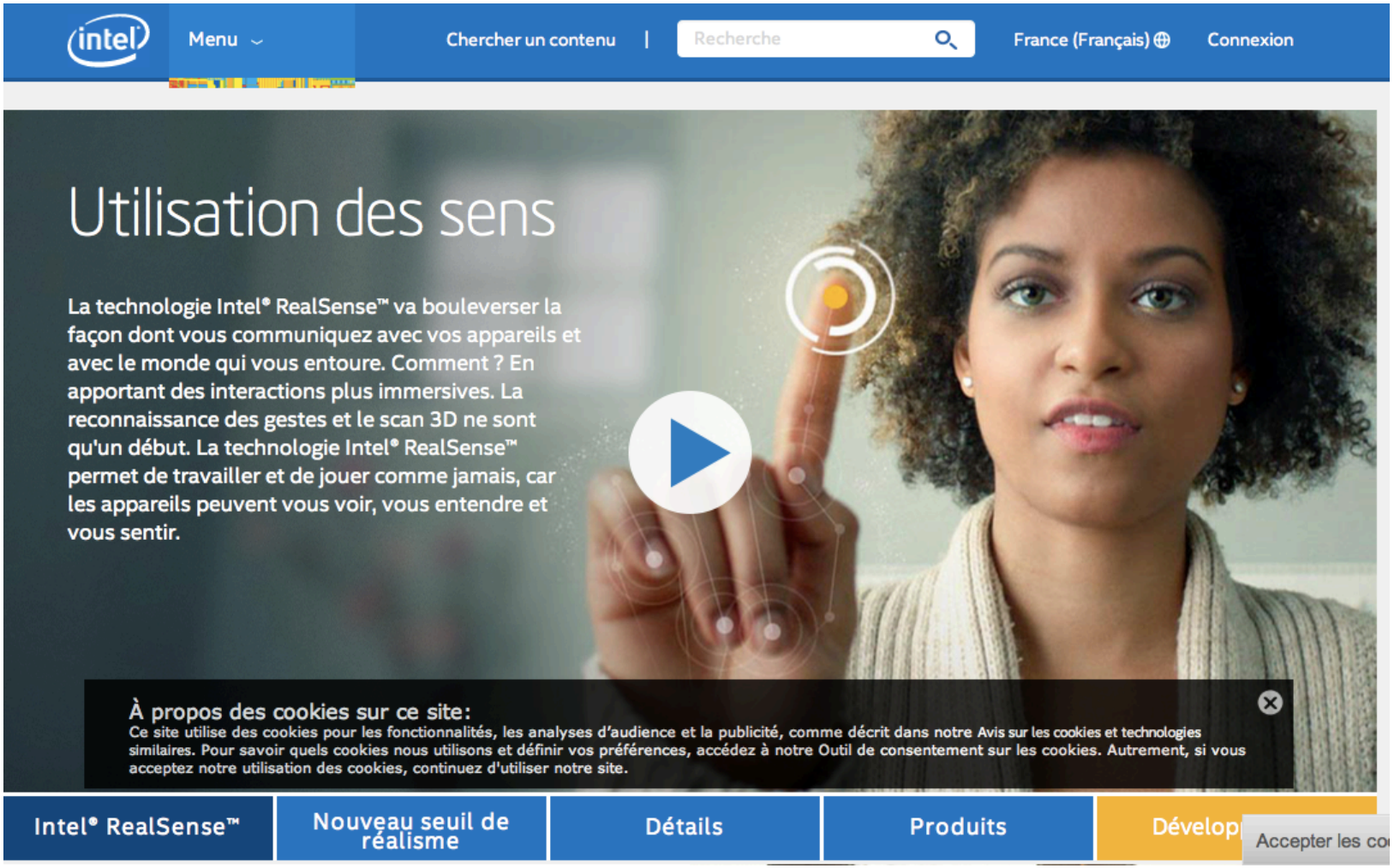
R. Bolt - "put that there" (MIT, 1980)

2014 : Google Glass

+ Aura (Optivent)
+ SpaceGlasses (Meta)
+ ReconJet (ReconInstrument)
+ Wrap 1200 DXAR (Vuzix)

nov. 2013

Intel "RealSense"



http://www.intel.fr/content/www/fr/fr/architecture-and-technology/realsense-overview.html

1) captation et filtrage

Tracking video (vraiment) minimal

Préalable :

se fabriquer une cible

mais
pourquoi en rouge ?

trackVideo

```
import processing.video.*;
Capture video;

void setup(){
  size(640, 480);
  video = new Capture(this, width, height, 30);
  video.start(); // pour Processing 2.0 et +
  noStroke();
  smooth();
}
void draw() {
if (video.available()){
  float hmax = 0;
  int tx = 0;
  int ty = 0;
  video.read();
  video.loadPixels();
  int index = 0;
  for (int y = 0; y < video.height; y++) {
   for (int x = 0; x < video.width; x++) {
    float h = hue(video.pixels[index]);
    // la teinte max donne du rouge
    if (h > hmax){
     hmax = h; tx = x; ty = y;
    }
    index++;
   }
  }
 // feedback graphique
 image(video, 0, 0, width, height);
 fill(255, 255, 0, 128);
 ellipse(tx, ty, 100, 100);
}
}
```

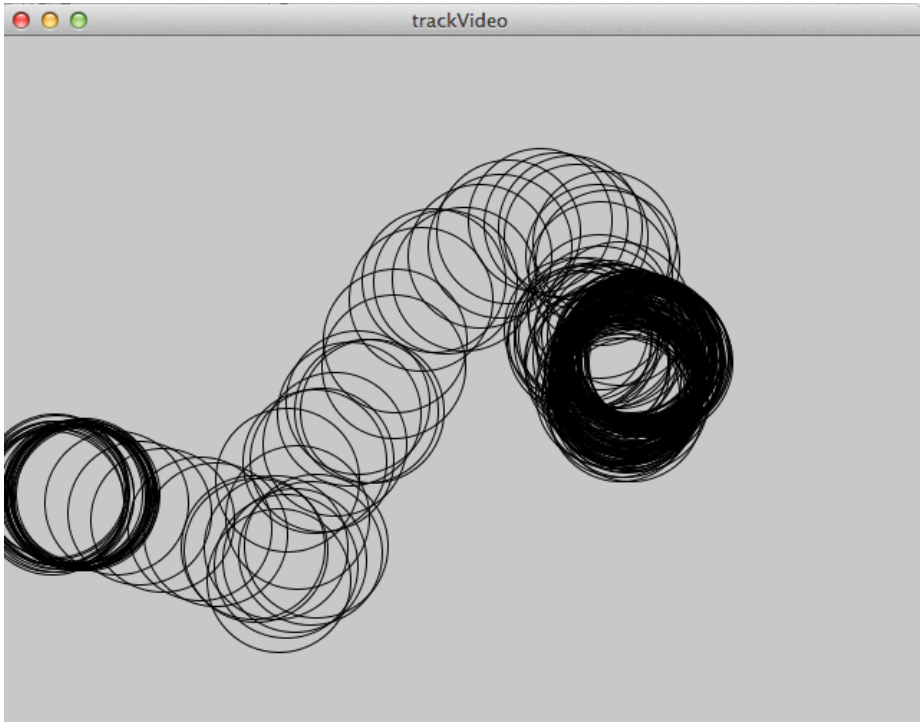
- étudier l'impact de l'appel video.available()

- enlever image() et transformer le dessin en un curseur qui évolue dans le sens du mouvement : comme un miroir

```
import processing.video.*;
Capture video;
int tx = 0;
int ty = 0;
float lemax = 0;

void setup() {
  size(640, 480);
  video = new Capture(this, width, height, 30);
  video.start();
  noStroke();
  smooth();
}
```

```
void draw() {
  if (video.available()) {
    video.read();
    video.loadPixels();
    int index = 0;
    for (int y = 0; y < video.height; y++) {
      for (int x = 0; x < video.width; x++){
        float h = hue(video.pixels[index]);
        // la teinte max donne du rouge
        if (h > lemax){
          lemax = h;
          tx = x;
          ty = y;
        }
        index++;
      }
    }
  // feedback graphique
    image(video, 0, 0, width, height);
    fill(255, 255, 0, 128);
    ellipse(tx, ty, 100, 100);
  }
}
```

Utilisable comme curseur pour dessiner ?



problème de bruit ! (tremblements, aléa de l'analyse)

Une solution pour "retirer" le bruit de captation :
utiliser un <u>filtre passe-bas</u>

variante simple très utilisée : le filtre alpha

$$y_{n+1} = (1-\alpha).y_n + \alpha.x_{n+1}$$

nouvelle estimation — ancienne estimation — nouvelle mesure

si alpha fort (>0.50) : réponse rapide mais bruitée
sinon : moins de bruit mais inertie

=> il faut tester (ici alpha = 30% = 0.3 pas mal)

modification du code :

```
import processing.video.*;
Capture video;
float txe = 0;
float tye = 0;
float alpha = 0.30;
```

nouvelles déclarations au début du code

```
 }
// feedback graphique
image(video, 0, 0, width, height);
txe = (1-alpha)*txe + alpha*tx;
tye = (1-alpha)*tye + alpha*ty;
stroke(255,255,0);
ellipse(txe, tye, 100, 100);
}
}
```

à la fin, filtrage avant le tracé

- tester avec des alpha forts et faibles
- gestion de l'echec de tracking ????

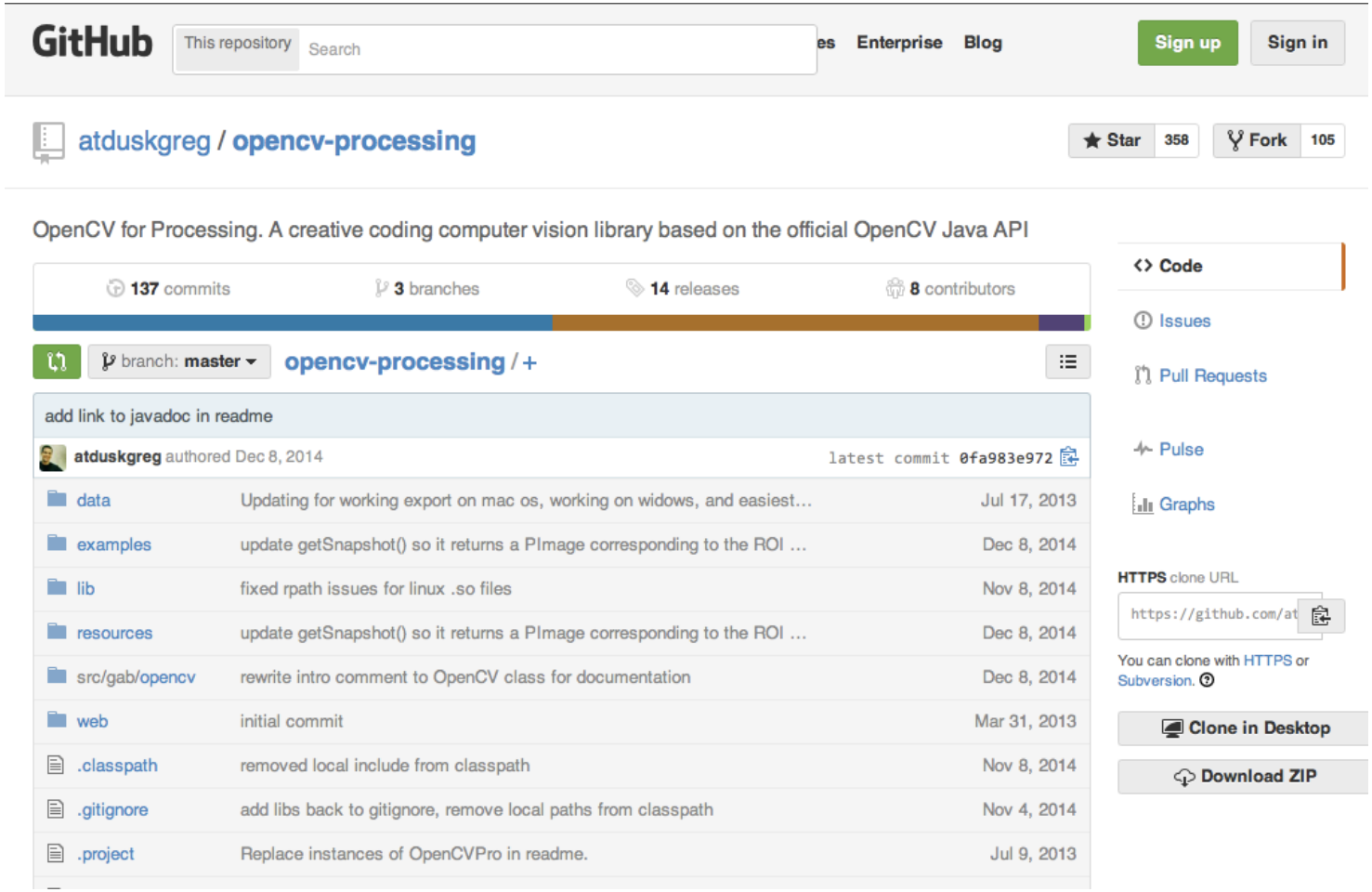
2) openCV

opencv.org

librairie créée chez Intel en Russie en 1999-2000

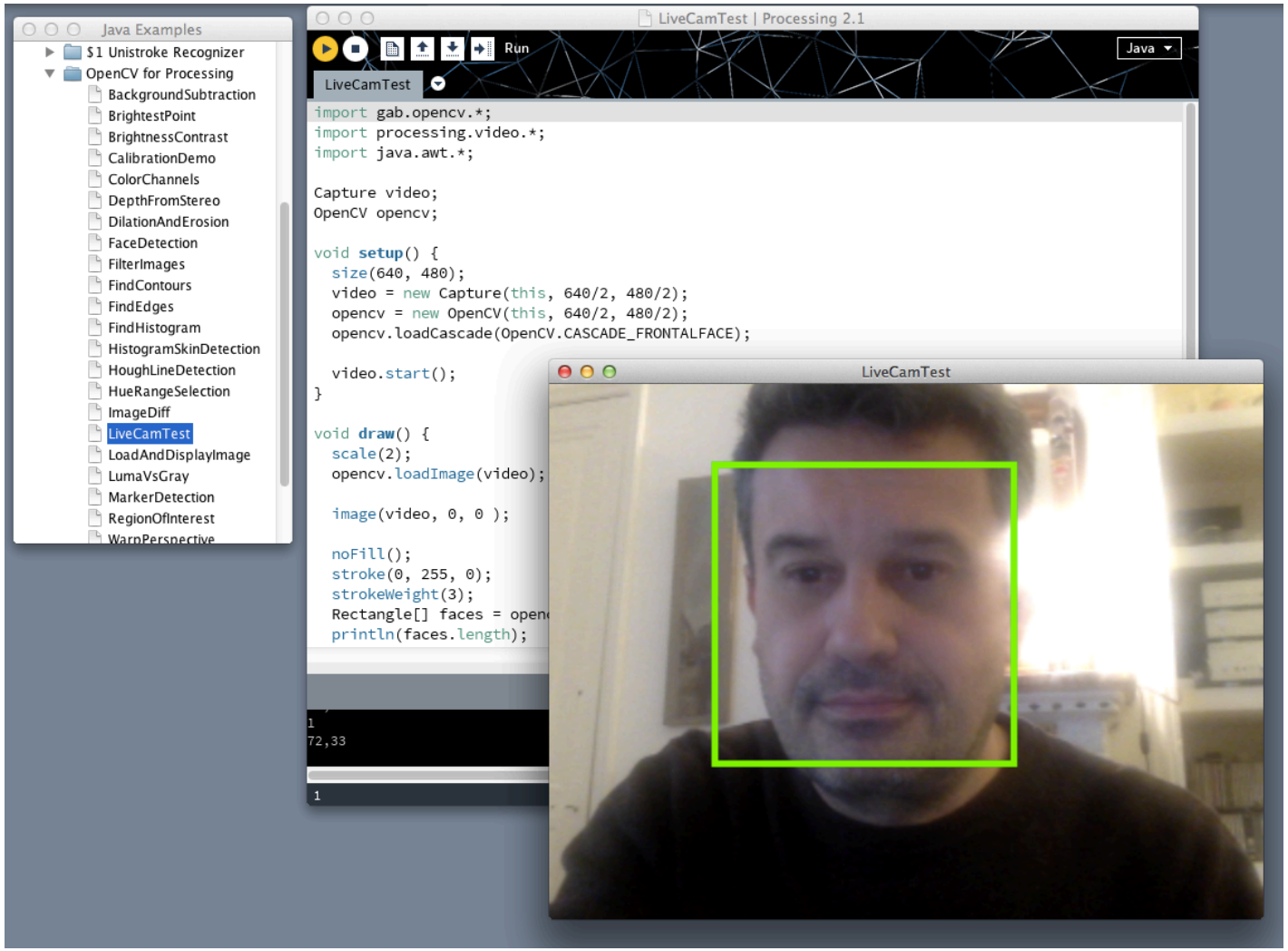
installation pour Processing

https://github.com/atduskgreg/opencv-processing



rque : il existe aussi un portage plus ancien fait aux Beaux-Arts d'Aix

Test de la reconnaissance faciale (demo "LiveCamTest")



Viola & Jones, 2001-4

Créer un compte Se connec

Article Discussion Lire Modifier le code Historique Rechercher

WIKIPÉDIA
L'encyclopédie libre

Accueil
Portails thématiques
Article au hasard
Contact

Contribuer
Débuter sur Wikipédia
Aide
Communauté
Modifications récentes
Faire un don

Imprimer / exporter
Créer un livre
Télécharger comme PDF
Version imprimable

Outils
Pages liées
Suivi des pages liées
Importer un fichier

Méthode de Viola et Jones

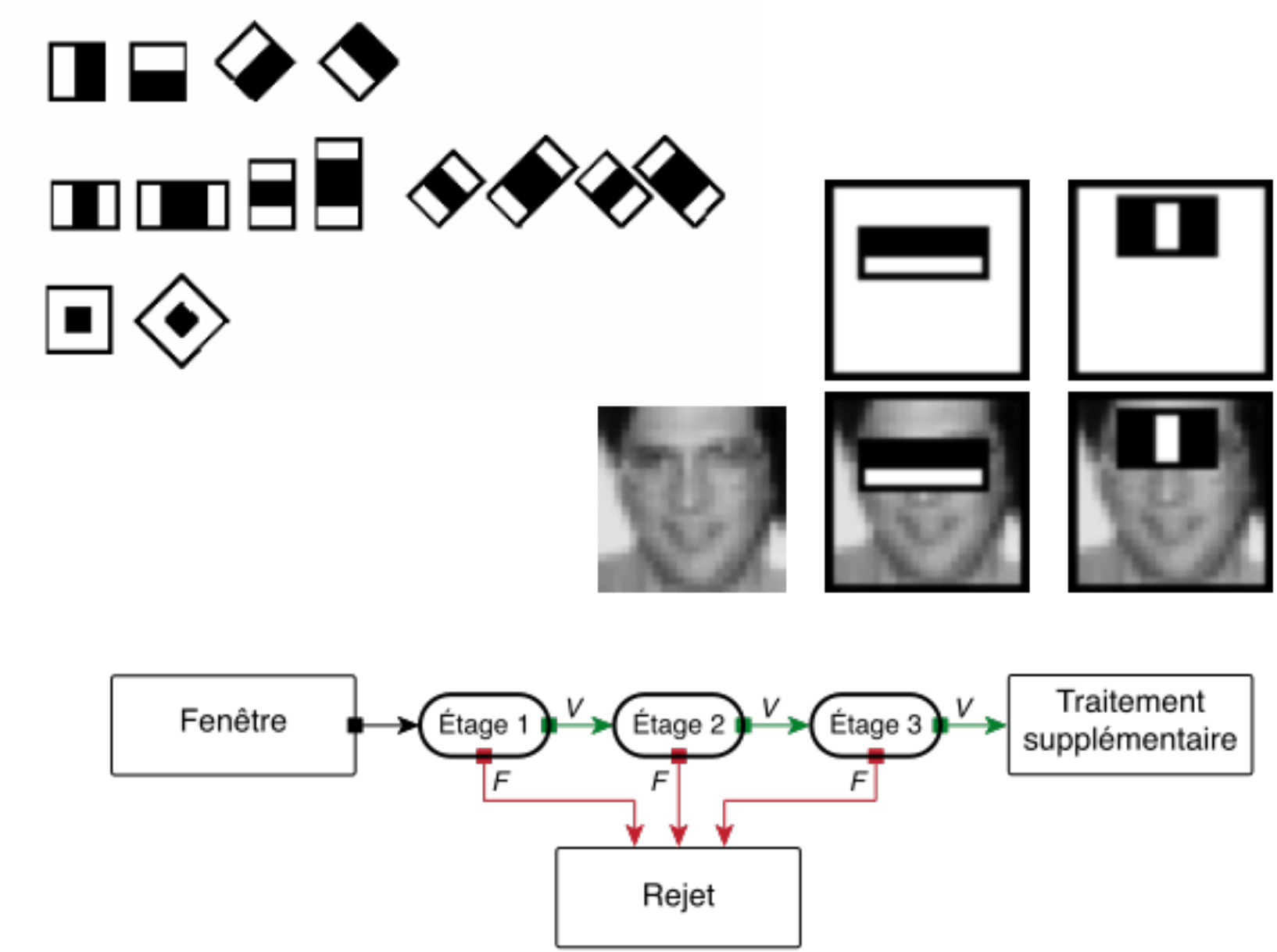
Vous lisez un « article de qualité ».

Un exemple de détection de visage par la méthode de Viola et Jones.

La **méthode de Viola et Jones** est une méthode de détection d'objet dans une image numérique, proposée par les chercheurs Paul Viola et Michael Jones en 2001. Elle fait partie des toutes premières méthodes capables de détecter efficacement et en temps réel des objets dans une image. Inventée à l'origine pour détecter des visages, elle peut également être utilisée pour détecter d'autres types d'objets comme des voitures ou des avions. La méthode de Viola et Jones est l'une des méthodes les plus connues et les plus utilisées, en particulier pour la détection de visages et la détection de personnes.

En tant que procédé d'apprentissage supervisé, la méthode de Viola et Jones nécessite de quelques centaines à plusieurs milliers d'exemples de l'objet que l'on souhaite détecter, pour entraîner un classifieur. Une fois son apprentissage réalisé, ce classifieur est utilisé pour détecter la présence éventuelle de l'objet dans une image en parcourant celle-ci de manière exhaustive, à toutes les positions et dans toutes les tailles possibles.

Considérée comme étant l'une des plus importantes méthodes de détection d'objet, la méthode de Viola et Jones est notamment connue pour avoir introduit plusieurs notions reprises ensuite par de nombreux chercheurs en vision par ordinateur, à l'exemple de la notion d'image intégrale ou de la méthode de classification construite


Fenêtre
Étage 1
V
Étage 2
V
Étage 3
V
Traitement supplémentaire
F
F
F
Rejet

Une variante …

face2shape | Processing 2.1

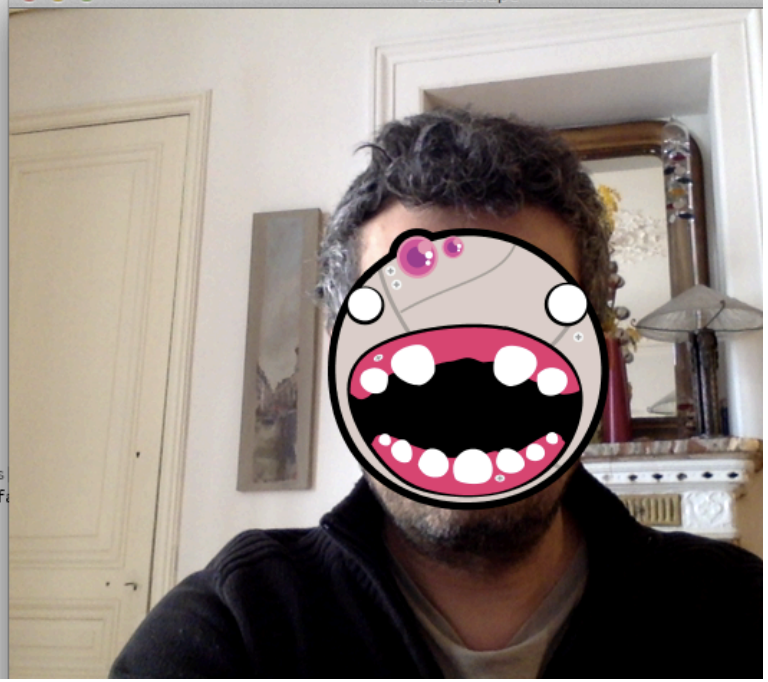
Run

Java

face2shape

```
import gab.opencv.*;
import processing.video.*;
import java.awt.*;
Capture video;
OpenCV opencv;
PShape bot;

void setup() {
  size(640, 480);
  video = new Capture(this, 640, 480);
  opencv = new OpenCV(this, 640, 480);
  opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);
  bot = loadShape("bot1.svg");

  video.start();
}

void draw() {
  //scale(2);
  opencv.loadImage(video);

  image(video, 0, 0 );

  noFill();
  stroke(0, 255, 0);
  strokeWeight(3);
  Rectangle[] faces = opencv.detect();
//  println(faces.length);

  for (int i = 0; i < faces.length; i++) {
//    println(faces[i].x + "," + faces[i].y);
    //rect(faces[i].x, faces[i].y, faces[i].width, faces
    shape(bot, faces[i].x, faces[i].y, faces[i].width, fa
  }
}

void captureEvent(Capture c) {
  c.read();
}
```

face2shape



marche aussi avec une image fixe

FaceDetectionFIXE

```
import gab.opencv.*;
import java.awt.Rectangle;

OpenCV opencv;
Rectangle[] faces;

void setup() {
  opencv = new OpenCV(this, "test.jpg");
  size(opencv.width, opencv.height);
  opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);
  faces = opencv.detect();
}

void draw() {
  image(opencv.getInput(), 0, 0);
  noFill(); stroke(0, 255, 0); strokeWeight(3);
  for (int i = 0; i < faces.length; i++) {
    rect(faces[i].x, faces[i].y, faces[i].width, faces[i].height);
  }
}
```

Variante pour les yeux

EyeDetectionCAM

```
Capture video;
OpenCV opencv;

void setup() {
  size(640, 480);
  video = new Capture(this, 640, 480);
  opencv = new OpenCV(this, 640, 480);
  opencv.loadCascade(OpenCV.CASCADE_EYE);
  video.start();
}

void draw() {
  image(video, 0, 0 );
  opencv.loadImage(video);
  Rectangle[] faces = opencv.detect();
  noFill();  stroke(0, 255, 0);  strokeWeight(3);
  for (int i = 0; i < faces.length; i++) {
    rect(faces[i].x, faces[i].y, faces[i].width, faces[i].height);
  }
}

void captureEvent(Capture c) {
```

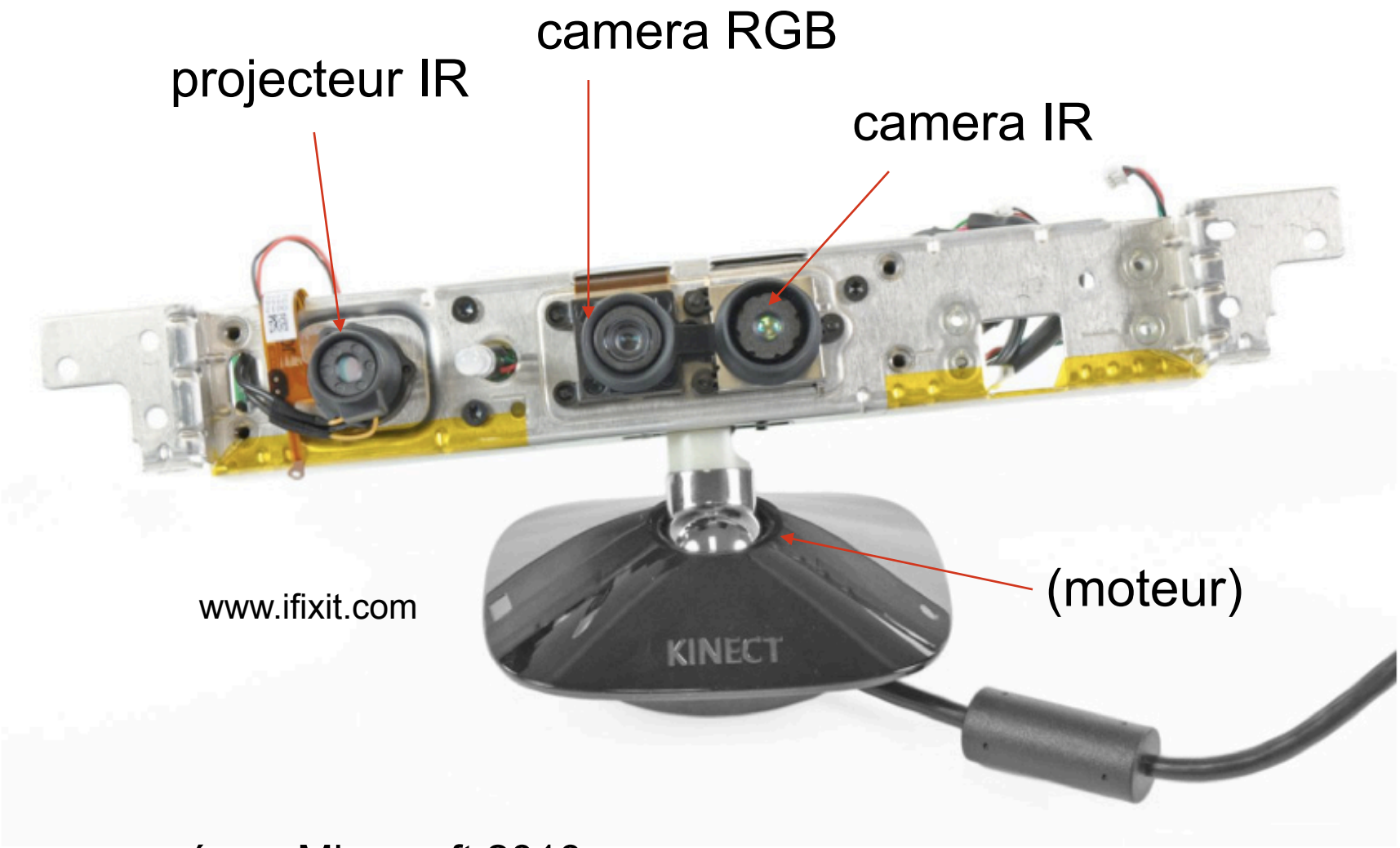
Seule chose qui change :-)

Autres cascades à tester ...

Nom	Date de modification	Taille	Type
haarcascade_clock.xml	20 janv. 2014 23:50	104 Ko	Texte XML
haarcascade_eye_tree_eyeglasses.xml	20 janv. 2014 23:50	1,1 Mo	Texte XML
haarcascade_eye.xml	20 janv. 2014 23:50	506 Ko	Texte XML
haarcascade_frontalface_alt_tree.xml	20 janv. 2014 23:50	3,6 Mo	Texte XML
haarcascade_frontalface_alt.xml	20 janv. 2014 23:50	920 Ko	Texte XML
haarcascade_frontalface_alt2.xml	20 janv. 2014 23:50	837 Ko	Texte XML
haarcascade_frontalface_default.xml	20 janv. 2014 23:50	1,3 Mo	Texte XML
haarcascade_fullbody.xml	20 janv. 2014 23:50	637 Ko	Texte XML
haarcascade_lefteye_2splits.xml	20 janv. 2014 23:50	323 Ko	Texte XML
haarcascade_lowerbody.xml	20 janv. 2014 23:50	531 Ko	Texte XML
haarcascade_mcs_eyepair_big.xml	20 janv. 2014 23:50	358 Ko	Texte XML
haarcascade_mcs_eyepair_small.xml	20 janv. 2014 23:50	410 Ko	Texte XML
haarcascade_mcs_leftear.xml	20 janv. 2014 23:50	313 Ko	Texte XML
haarcascade_mcs_lefteye.xml	20 janv. 2014 23:50	778 Ko	Texte XML
haarcascade_mcs_mouth.xml	20 janv. 2014 23:50	720 Ko	Texte XML
haarcascade_mcs_nose.xml	20 janv. 2014 23:50	1,6 Mo	Texte XML
haarcascade_mcs_rightear.xml	20 janv. 2014 23:50	325 Ko	Texte XML
haarcascade_mcs_righteye.xml	20 janv. 2014 23:50	1,4 Mo	Texte XML
haarcascade_mcs_upperbody.xml	20 janv. 2014 23:50	1,5 Mo	Texte XML
haarcascade_profileface.xml	20 janv. 2014 23:50	1,1 Mo	Texte XML
haarcascade_righteye_2splits.xml	20 janv. 2014 23:50	325 Ko	Texte XML
haarcascade_upperbody.xml	20 janv. 2014 23:50	1 Mo	Texte XML
hogcascade_pedestrians.xml	20 janv. 2014 23:50	130 Ko	Texte XML
lbpcascade_frontalface.xml	20 janv. 2014 23:50	52 Ko	Texte XML

Macintosh HD ▸ Utilisat ▸ pcubau ▸ Docum ▸ Process ▸ librarie ▸ opencv ▸ library ▸ cascade-files

3) La Kinect



- annoncé par Microsoft 2010
- record de vente (10M en 2011)
- soft+puce de reconnaissance de forme : société PrimeSense (Israel)

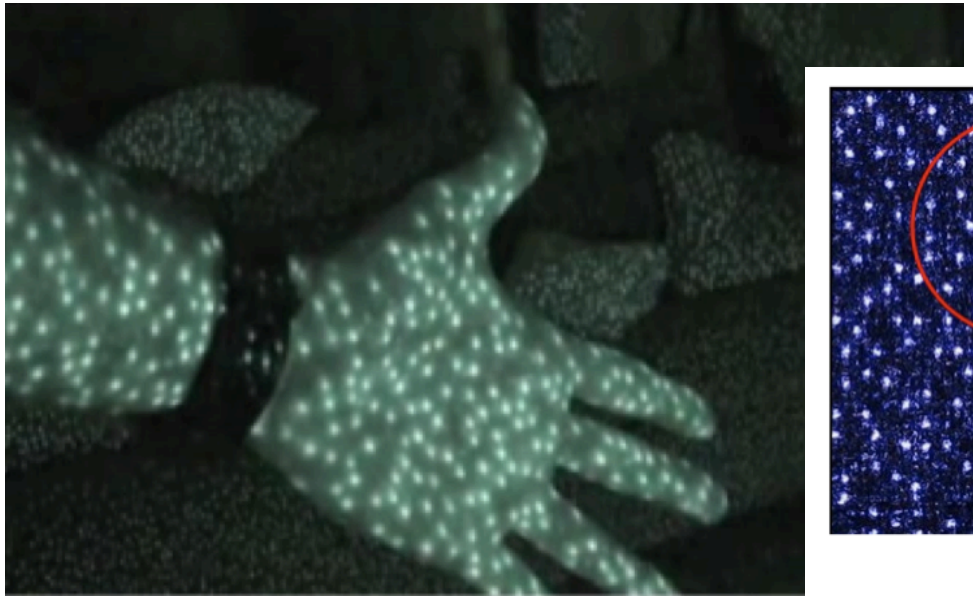
Principe de la reconstruction 3D :

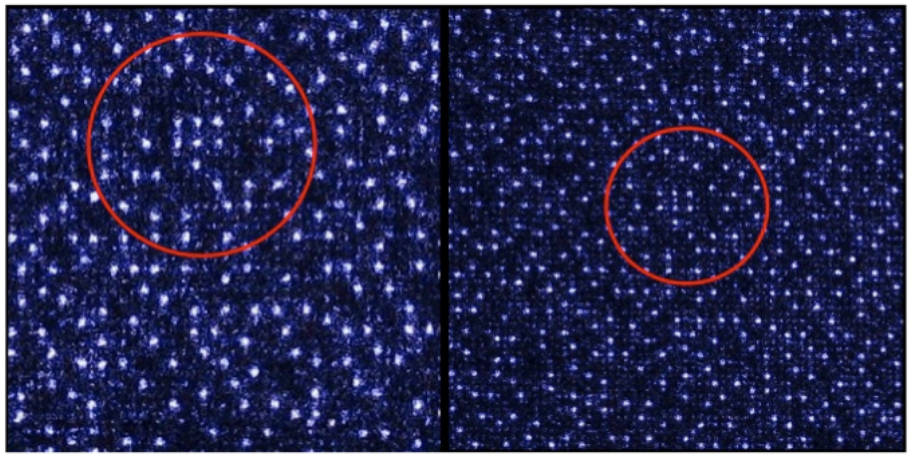
-le projecteur IR diffuse un semis de point sur les surfaces à analyser

-la caméra IR récupère l'image

-la position 3D est déduite en mixant 2 méthodes

(1) la dimension des points (2) la triangulation





Pas tout à fait aléatoire !

il y a 3 types de motifs, adaptés aux distances

=> bon topo en ligne: H. Wannous, Telecom Lille

Installation du wrapper Processing de OpenNI

http://code.google.com/p/simple-openni/

La carte de profondeur (depthmap et depthimage)

esssaiKinect1

```
import SimpleOpenNI.*;
SimpleOpenNI kinect;
void setup() {
  size(640*2, 480);
  kinect = new SimpleOpenNI(this);
  kinect.enableDepth();
  kinect.enableRGB();
  kinect.alternativeViewPointDepthToImage();
}
void draw() {
  background(0,255,0);
  kinect.update();
  image(kinect.depthImage(), 0, 0);
  image(kinect.rgbImage(), 640, 0);
}
```

- séparation fond/humains
- identifications des humains

Reprise du jeu incroyable

avec le point le plus proche dans la depthmap comme pointeur

basé sur G. Borensten "Making things see" O'Reilly

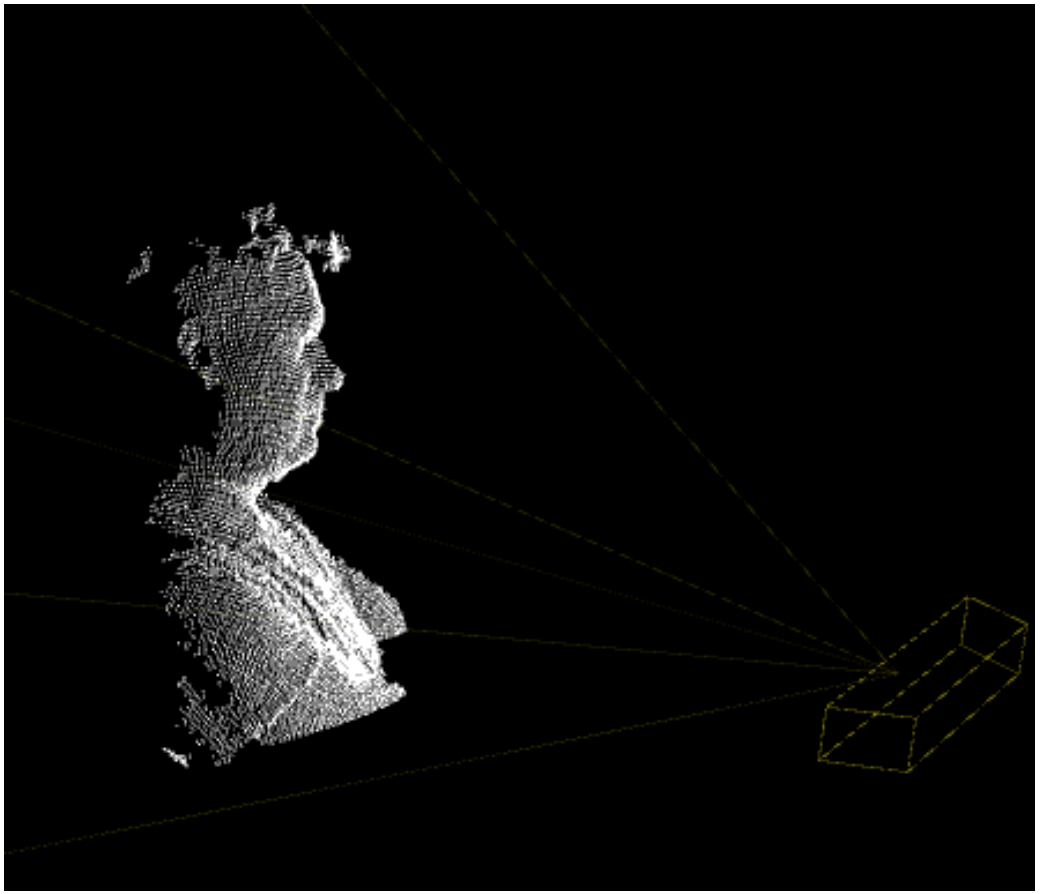
Code du livre

```
import SimpleOpenNI.*;
SimpleOpenNI kinect;

int closestValue;
int closestX;
int closestY;

void setup() {
  size(640, 480);
  kinect = new SimpleOpenNI(this);
  kinect.enableDepth();
}
```

```
void draw() {
  closestValue = 8000;
  kinect.update();

  // get the depth array from the kinect
  int[] depthValues = kinect.depthMap();

  for(int y = 0; y < 480; y++){
   for(int x = 0; x < 640; x++){
       int i = x + y * 640;
       int currentDepthValue = depthValues[i];
       if(currentDepthValue > 0 && currentDepthValue < closestValue){
         closestValue = currentDepthValue;
         closestX = x;
         closestY = y;
       }
     }
   }

  //draw the depth image on the screen + le point le plus proche
  image(kinect.depthImage(),0,0);
  fill(255,0,0);
  ellipse(closestX, closestY, 25, 25);
}
```

il faudra
filtrer le point

Reconstruction 3D basée sur la depthmap



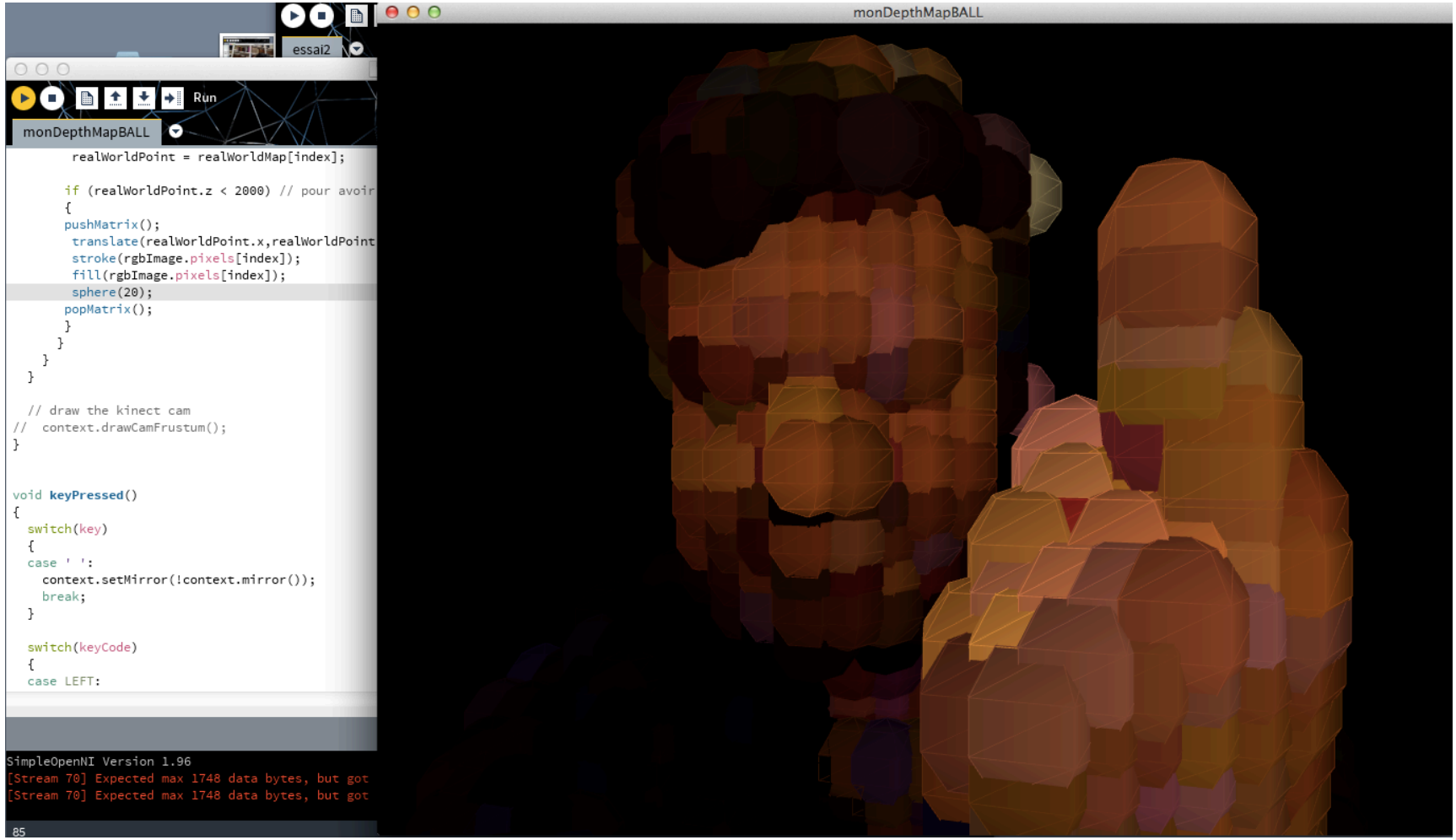
demo "DepthMap3d" de SimpleOpenNI

vimeo Join Log In Create Watch Upload Search VIDEOS

Body Dysmorphic Disorder

from **flight404** PLUS 3 years ago NOT YET RATED

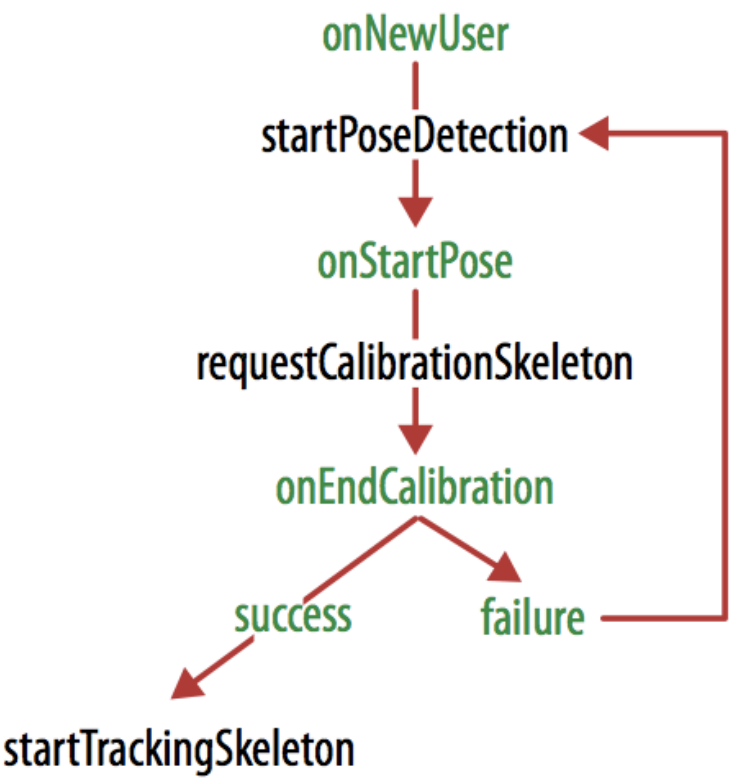
Made with Cinder and the Kinect sensor. Runs in realtime.
flight404.com/blog/?p=472


essai2
Run
monDepthMapBALL
realWorldPoint = realWorldMap[index];
if (realWorldPoint.z < 2000) // pour avoir
{
pushMatrix();
translate(realWorldPoint.x,realWorldPoint
stroke(rgbImage.pixels[index]);
fill(rgbImage.pixels[index]);
sphere(20);
popMatrix();
}
}
}
}
// draw the kinect cam
// context.drawCamFrustum();
}
void keyPressed()
{
switch(key)
{
case ' ':
context.setMirror(!context.mirror());
break;
}
switch(keyCode)
{
case LEFT:
SimpleOpenNI Version 1.96
[Stream 70] Expected max 1748 data bytes, but got
[Stream 70] Expected max 1748 data bytes, but got
85
monDepthMapBALL

```
// ici variables globales comme dans
// le code de "depthmap3d"

void setup(){
 // ici debut  comme dans code depthmap3d
  sphereDetail(5);
}
```

```
void draw() {
 // debut  comme dans code depthmap3d

 PVector[] realWorldMap = context.depthMapRealWorld();

  // draw pointcloud`
  for(int y=0;y < context.depthHeight();y+=steps)  {
    for(int x=0;x < context.depthWidth();x+=steps)   {
      index = x + y * context.depthWidth();
      if((depthMap[index] > 0))      {      // draw the projected point
        realWorldPoint = realWorldMap[index];

         if (realWorldPoint.z < 2000) {      // pour avoir juste moi
           pushMatrix();
           translate(realWorldPoint.x,realWorldPoint.y,realWorldPoint.z);
           stroke(rgbImage.pixels[index]);
           fill(rgbImage.pixels[index]);
           sphere(20);
          popMatrix();
         }
      }
    }
  }

  // draw the kinect cam
//  context.drawCamFrustum();
}
```

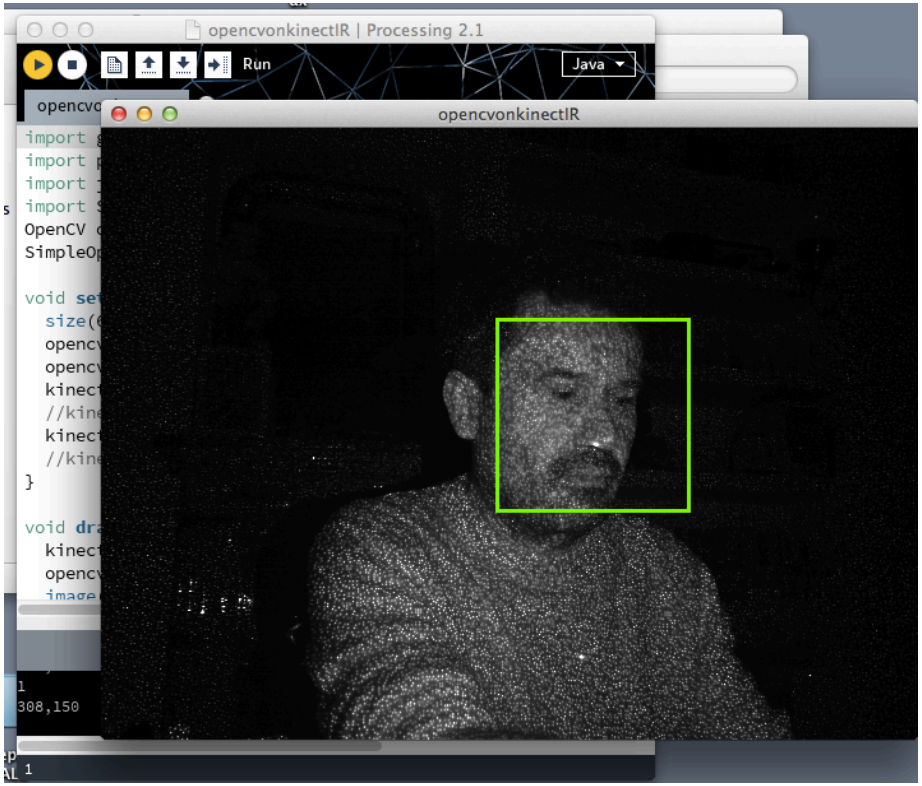
Suivi du corps : demo "User" de simpleNI



G. Borensten "Making things see" O'Reilly

Couplage Kinect - openCV

avec la depthmap



avec la camera IR

```
import gab.opencv.*;
import processing.video.*;
import java.awt.*;
import SimpleOpenNI.*;
OpenCV opencv;
SimpleOpenNI kinect;

void setup() {
  size(640, 480);
  opencv = new OpenCV(this, 640, 480);
  opencv.loadCascade(OpenCV.CASCADE_FRONTALFACE);
  kinect = new SimpleOpenNI(this);
  kinect.enableIR();
}
```

```
void draw() {
  kinect.update();

opencv.loadImage(kinect.irImage());

image(kinect.irImage(), 0, 0 );
 noFill(); stroke(0, 255, 0); strokeWeight(3);

Rectangle[] faces = opencv.detect();

  for (int i = 0; i < faces.length; i++) {
    println(faces[i].x + "," + faces[i].y);
    rect(faces[i].x, faces[i].y, faces[i].width, faces[i].height);
  }
}
```

A voir ...

Microsoft Translator | Choose language

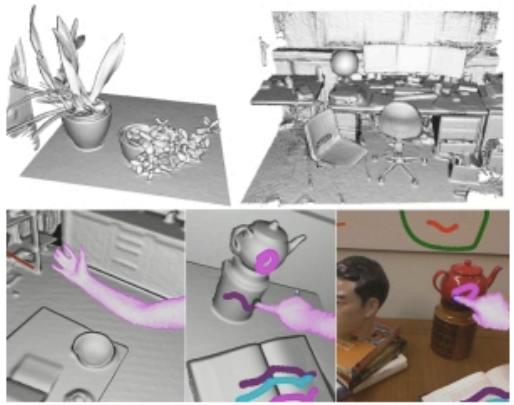
Microsoft Research

Search Microsoft Research

Our research Connections Careers About us

All Downloads Events Groups News People Projects Publications Videos

KinectFusion Project Page



This project investigates techniques to track the 6DOF position of handheld depth sensing cameras, such as Kinect, as they move through space and perform high quality 3D surface reconstructions for interaction. Other collaborators (missing from the list below): Richard Newcombe (Imperial College London); David Kim (Newcastle University & Microsoft Research); Andy Davison (Imperial College London)

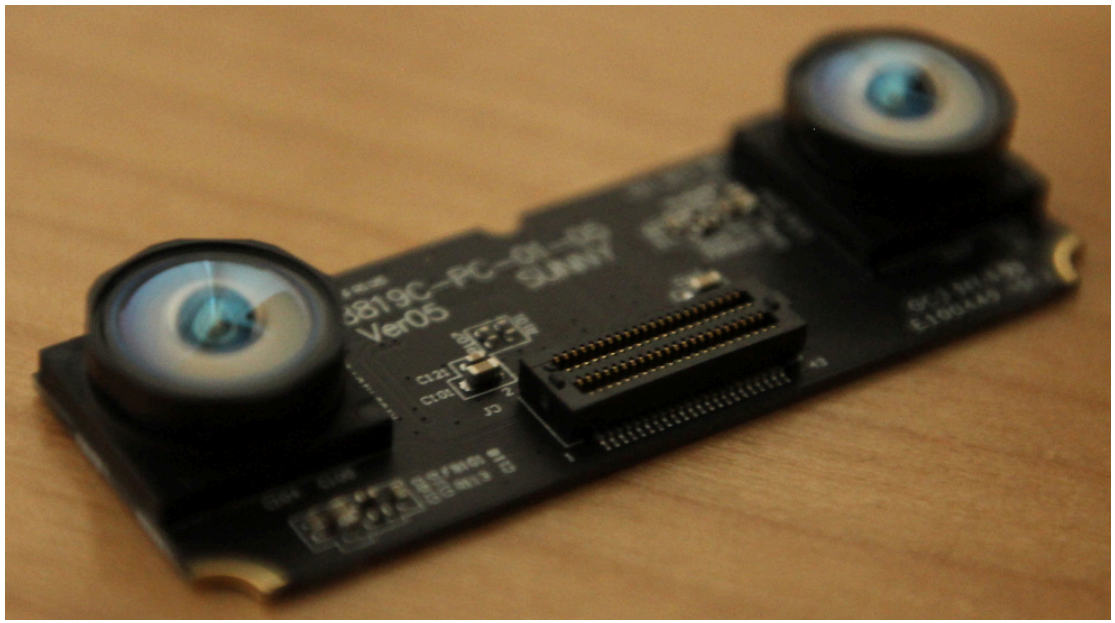
Publications

- Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon, KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera, ACM Symposium on User Interface Software and Technology, October 2011
- Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon, KinectFusion: Real-Time Dense Surface Mapping and Tracking, in *IEEE ISMAR*, IEEE, October 2011

Videos

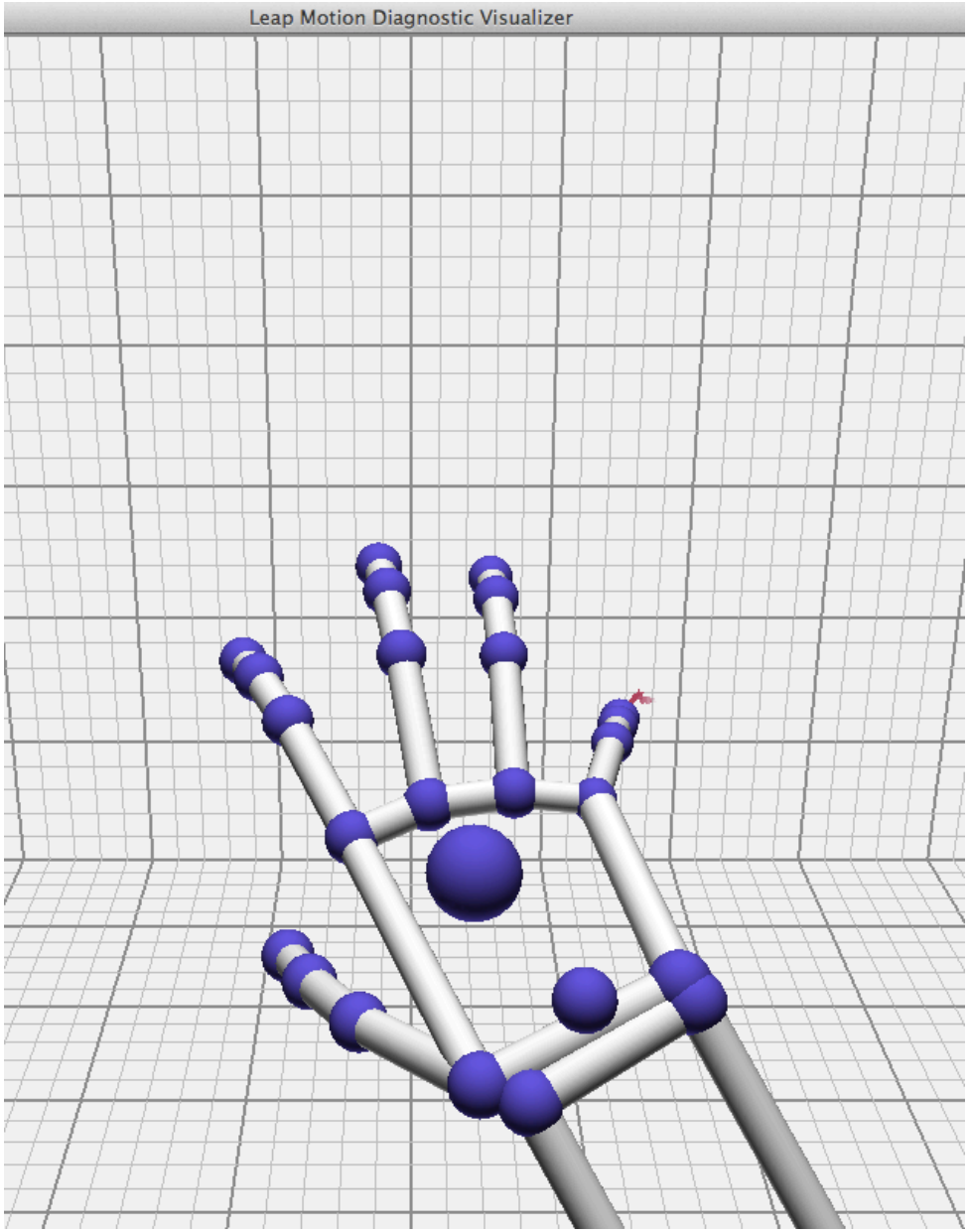
+ des centaines d'autres projets de recherche

5) Leap Motion (v1)

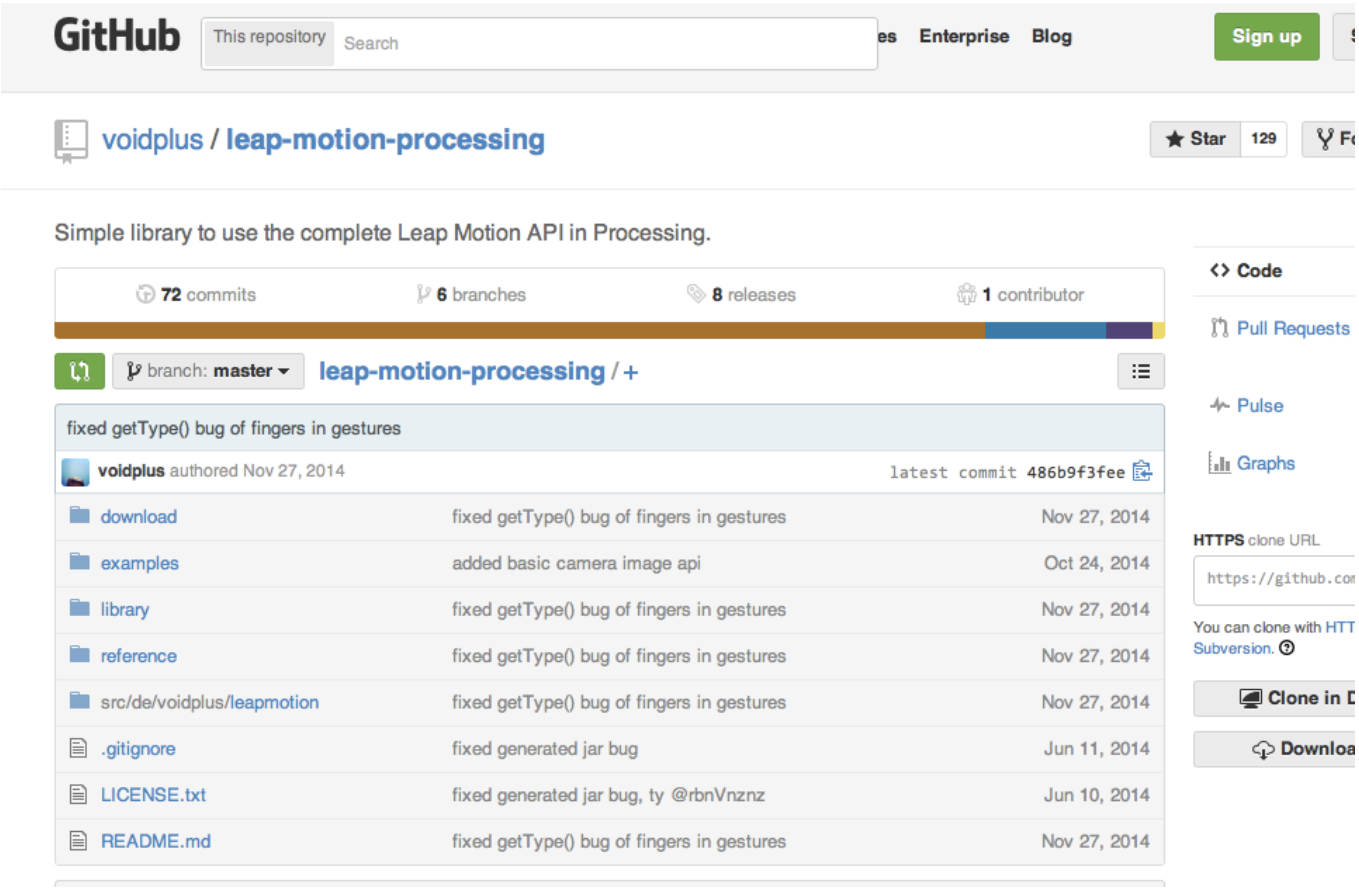


www.ifixit.com

Application
de test



installer la librairie Processing



https://github.com/voidplus/leap-motion-processing

la demo "e1_basic"

e1_basic | Processing 2.1

Run

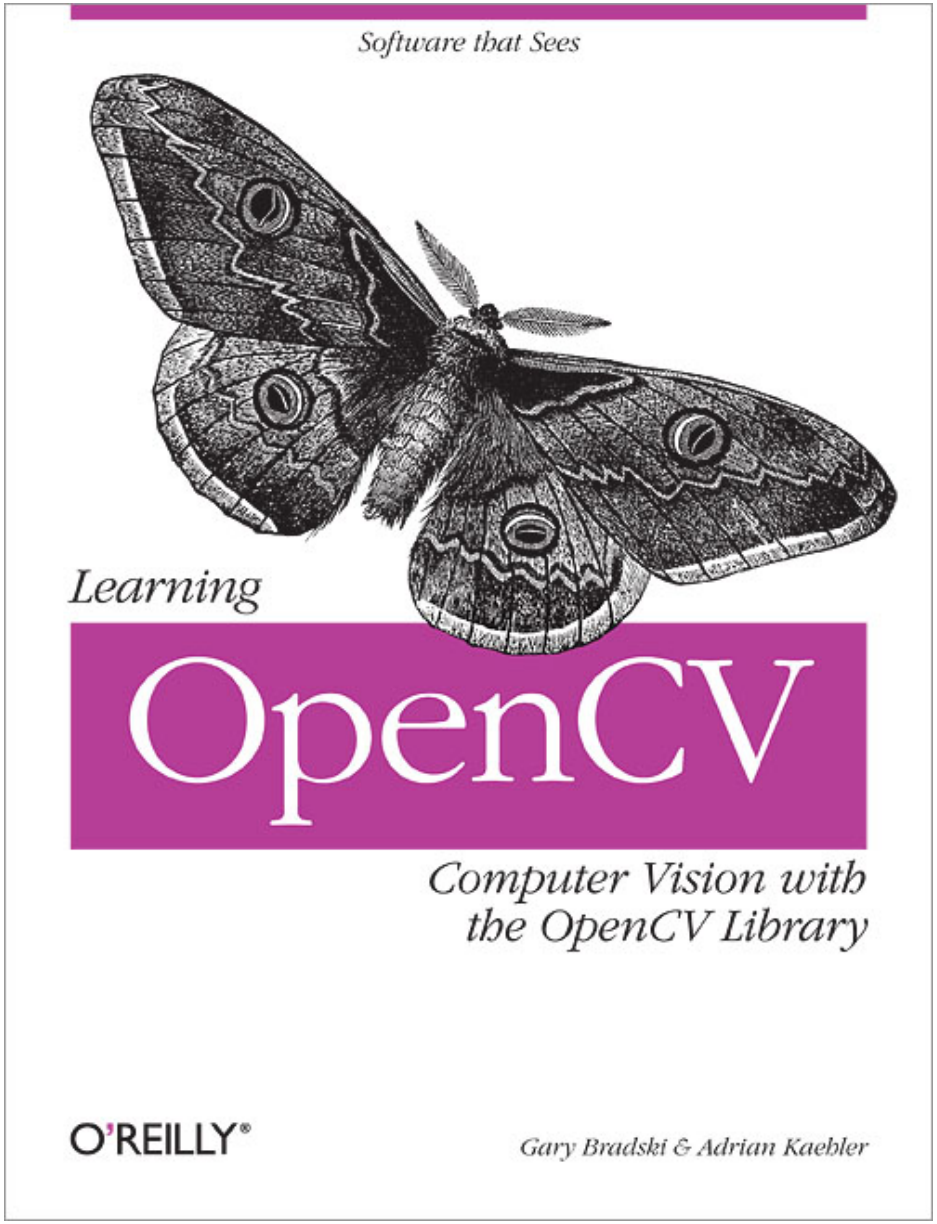
e1_basic

```
import de.voidplus.leapmotion.*;

LeapMotion leap;

void setup(){
  size(800, 500, P3D);
  background(255);
  noStroke(); fill(50);
  // ...

  leap = new LeapMotion(this);
}

void draw(){
  background(255);
  // ...
  int fps = leap.getFrameRate();

  // HANDS
  for(Hand hand : leap.getHands()){

    hand.draw();
    int     hand_id          = hand.getId();
    PVector hand_position    = hand.getPosition();
```

```
# LeapMotion-Library v1.1.2 - LeapMotion-SDK v0.8.1.6221 -
https://github.com/voidplus/leap-motion-processing
```

1

e1_basic

Bibliographie

Plus technique