

# **Médias interactifs numériques**

## **augmentation**

# **Processing : la captation**

Pierre Cubaud

Département Informatique

CNAM

## Premier exemple avec la webcam



```
trackVideo
```



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

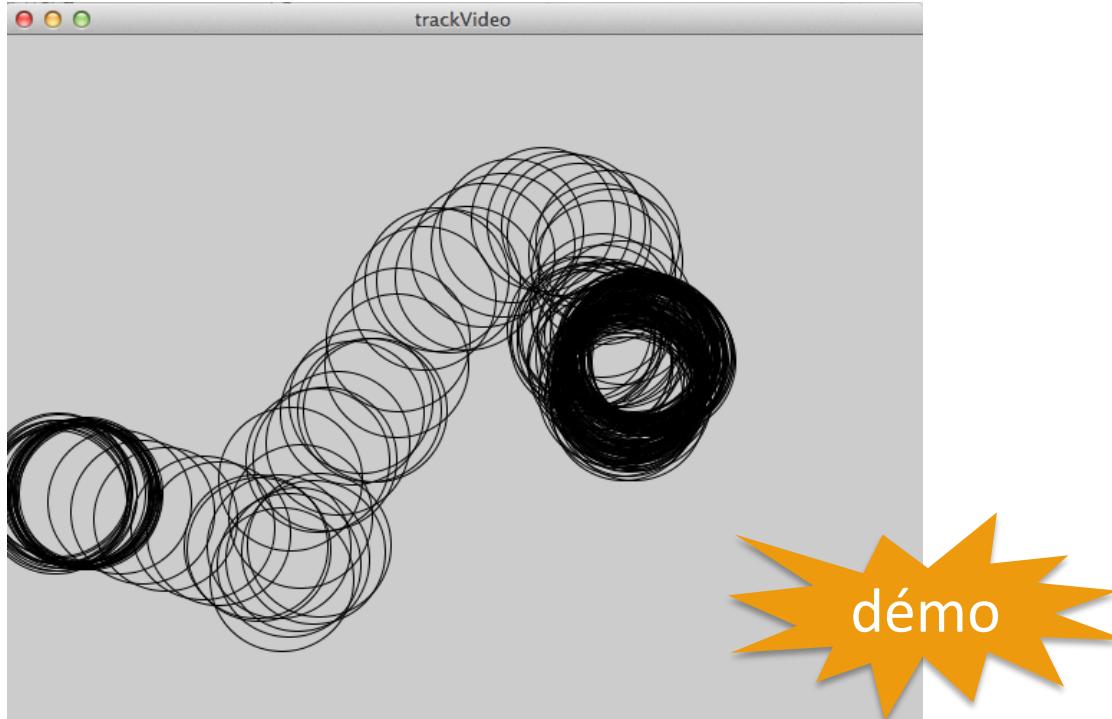
void setup(){
  size(640, 480);
  video = new Capture(this, width, height, 30);
  video.start(); // pour Processing 2.0 et +
  noStroke();
  smooth();
}
void draw() {
if (video.available()){
  float hmax = 0;
  int tx = 0;
  int ty = 0;
  video.read();
  video.loadPixels();
  int index = 0;
  for (int 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

Utilisable comme curseur pour dessiner ?



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

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

variante simple très utilisée : le filtre alpha

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

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;

}

// 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);
}
```

nouvelles déclarations au début du code

à la fin, filtrage avant le tracé

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



# Nintendo Wii (2006)



- accéléromètre 3 axes
  - camera IR + rec. Blobs
  - HP, vibreur
  - Plein de boutons + joysticks
  - Bluetooth
- (et i2c avec le nunchuck)

⇒hacké l'année suivante

40 € wiimote + 20 € nunchuck

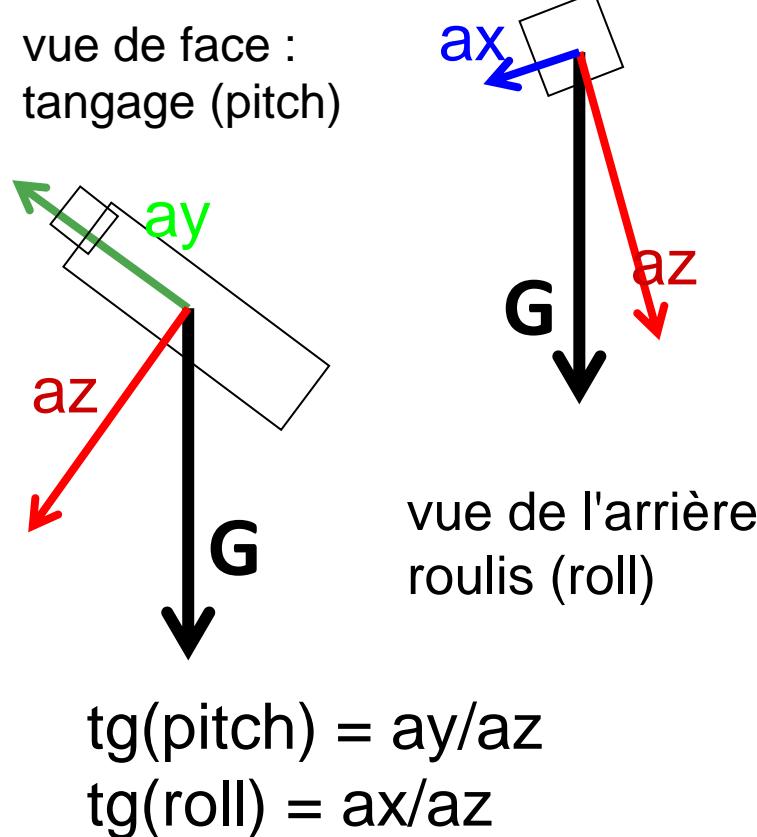
# Utilisation de l'accéléromètre



# application à la mesure d'angles



ma (belle) lunette astronomique



```
demowii  
import lll.wrj4P5.*;  
import lll.Loc.*;  
import wiiremotej.*;  
import wiiremotej.event.*;  
Wrj4P5 mawii;  
  
void setup(){  
  size(600,600);fill(255);noStroke();smooth();  
  rectMode(CENTER);  
  mawii=new Wrj4P5(this).connect();  
}  
  
void draw(){  
  if (mawii.isConnecting()) return;  
  Loc s = mawii.rimokon.sensed;  
  float pitch = atan2(s.z,s.y);  
  float roll = atan2(s.z, s.x);  
  background(100);  
  pushMatrix();  
  translate(300,300);  
  rotate(pitch);  
  rect(0,0,50,300);  
  popMatrix();  
}
```



- ajouter le filtre alpha sur pitch
- ajouter la vue arrière et visualiser le roll
- photo wiimote au lieu de rect() ?

# (installer une librairie pour Processing)

Accéder à la wiimote avec Processing :  
la librairie wrj4p5 !!! n'existe plus pour 2.0 ☹

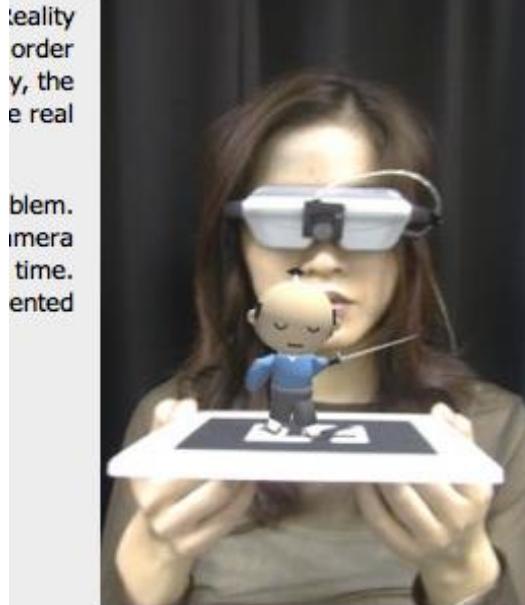
explications (en anglais) sur le site :  
[http://wiki.processing.org/w/How\\_to\\_Install\\_a\\_Contributed\\_Library](http://wiki.processing.org/w/How_to_Install_a_Contributed_Library)

```
Documents
  Processing
    your sketch folders
    libraries
      theLibrary
        examples
        library
          theLibrary.jar
        reference
        src
```

# Réalité augmentée avec ARToolkit

le cnam

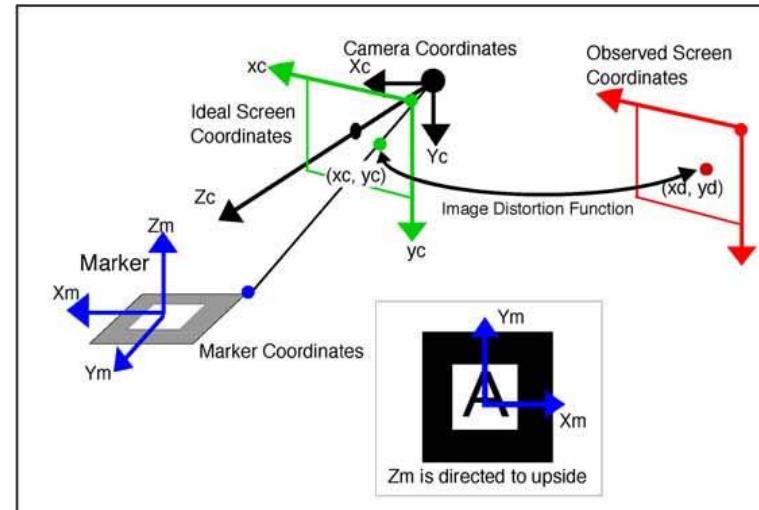
it can be seen by the user in the head set  
e virtual character moves with it and appears



that have used ARToolkit, sample ARToolkit

<http://www.hitl.washington.edu/artoolkit/>

actif 1999-2007 ? un grand succès  
version "pro" par ARtoolworks



# demo "simpleLite" de la librairie NyArtoolkit

[http://nyatla.jp/nyartoolkit/wp/?page\\_id=166](http://nyatla.jp/nyartoolkit/wp/?page_id=166)

le cnam

```
simpleLite
cam=new Capture(this,640,480);
nya=new MultiMarker(this,width,height,"camera_para.dat",NyAR4PsgConfig.CONFIG_PSG);
nya.addARMarker("patt.hiro",80);
}

void draw()
{
  if (cam.available() !=true) return;
  cam.read();
  nya.detect(cam);
  background(0);
  nya.drawBackground(cam);
  if(!nya.isExistMarker(0)) return;
  nya.beginTransform(0);
  fill(0,255,255);
  translate(0,0,20);
  box(40);
  translate(0,0,40);
  fill(255,0,255);
  sphere(20);
  nya.endTransform();
}
```



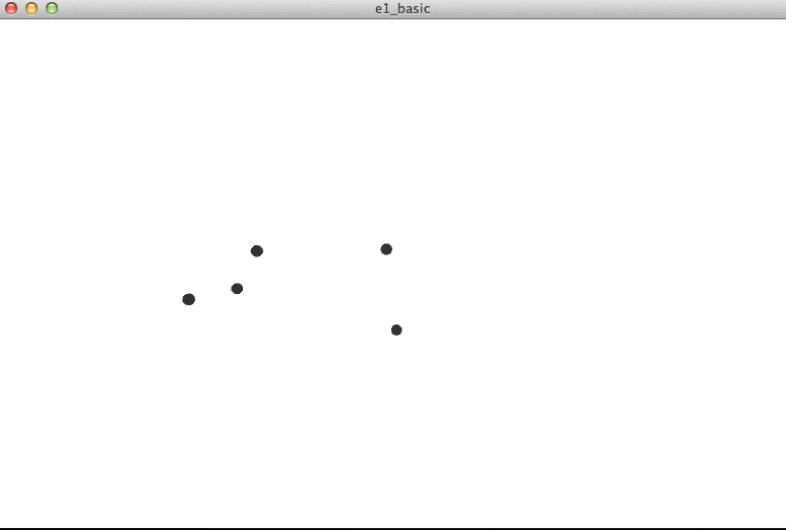
# Leap Motion (2013)



Leap Motion is almost amazing. I  
... . . .

essais au CNAM (oct. 2013)

# librairie seulement pour Processing 2.1

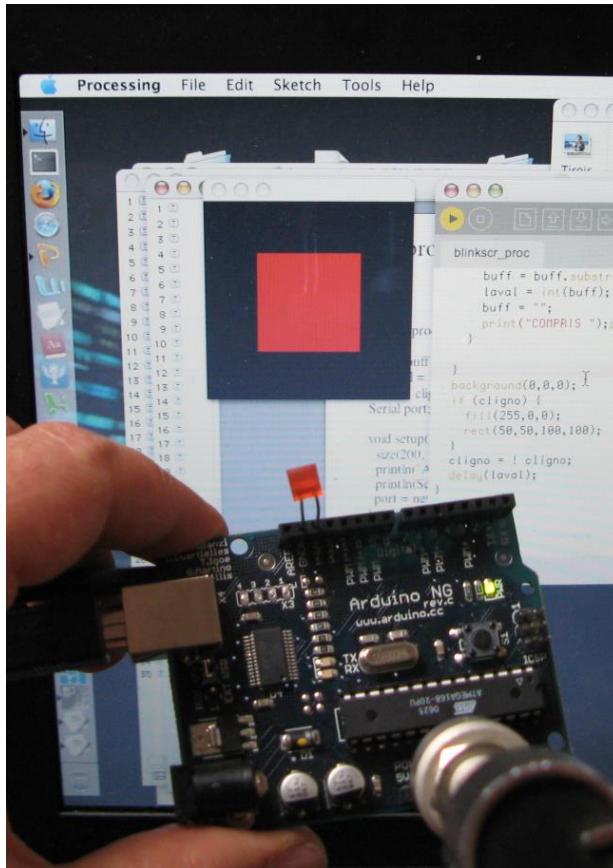


```
import de.voidplus.leapmotion.*;  
  
LeapMotion leap;  
  
void setup(){  
  size(800, 500, P3D);  
  background(255);  
  noStroke(); fill(50);  
  // ...  
  
  leap = new LeapMotion(this);  
}  
  
void draw(){  
  background(255);  
  // ...  
  int fps = leap.getFrameRate();  
  
  // HANDS  
  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
```

resultat de la  
demo "e1\_basic" :-)

# pour finir : Arduino

le cnam



<http://arduino.cc>

20 à 30 euros la carte  
de base

# exemple "physical pixel" d'arduino

