Systèmes embarqués
Master TSI

Julien Marot

julien.marot@fresnel.fr

Plan

1 Supports aux systems embarqués

2 Analyse de visages sur Raspberry Pi

3 Détection et suivi de visage dans une application de vending

4 Détection de défauts, contrôle non destructif de pièces métalliques

5 Détection de formes dans des images géo-radar (démo)

1 Supports aux systèmes embarqués

Hardware supports

- Tablets
- SmartPhones
- Personal Computers
- Laptops

Data exchange supports

High rate Internet, Wi-Fi, …

- Problématiques: temps de calcul, transfert d'information entre les supports

Low-cost plateforms, open-source code

Architecture

- Thymio:
  Windows
  several types of sensors

- LegoMindstorms:
  Linux OS
  4 inputs for data acquisition

Programming principles

Visual programming environment
functionnal block diagram, C

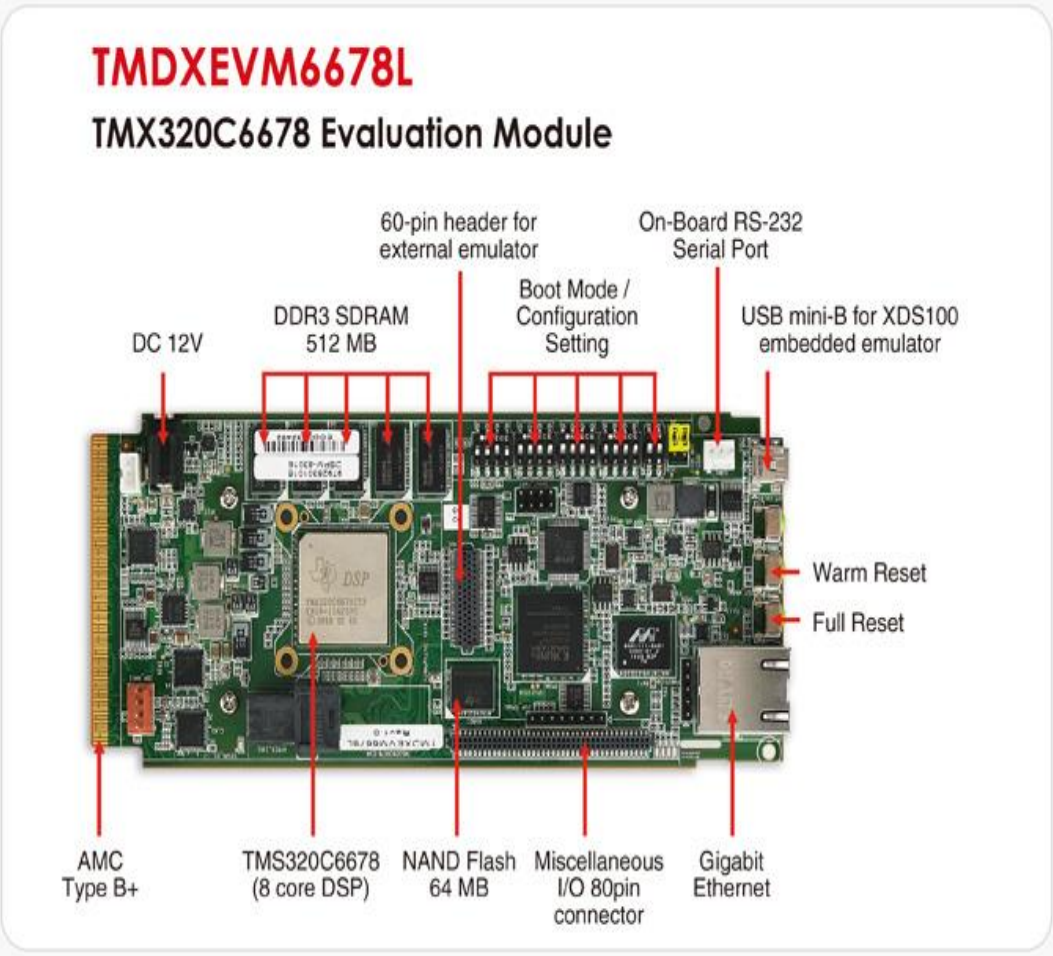
Multi-core DSP
industry oriented courses

Architecture

- T6678:
  8 Core DSP
- Several types of I/O ports

Programming principles

Code Composer Studio



Raspberry Pi

Architecture~Nanocomputer

- Four USB ports
- GPIO port
- Memory chip, and ARM processor underneath
- SD-card: Hard Disk

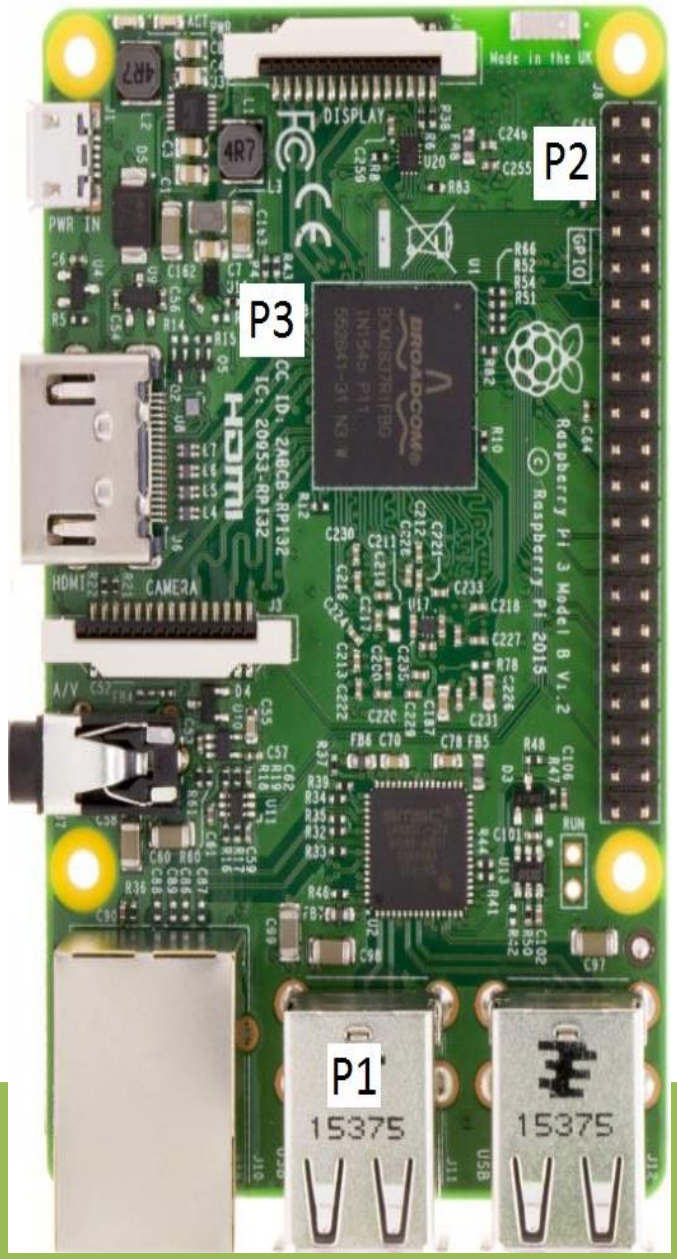
- Model 3: 64 bits,1.2Ghz

Programming principles

OS: Linux command line or GUI, Softwares: Python, OpenCV …

Applications

Image processing, Robotics, …



Hardware

Raspberry Pi
Micro-SD card for the operating system
~ 60 €

Camera Pi
Infra Red or classical
~ 40 €

USB Hub
Peripherals
Screen
Keyboard
Mouse

- Application d’intelligence artificielle

Robot UBBO Axyn:

Robot d’accueil

http://www.axyn.fr/

Operating system:

Raspbian Wheezy:

boots to command line ~ Linux

Raspbian Jessie:

boots straight to the desktop graphical user interface

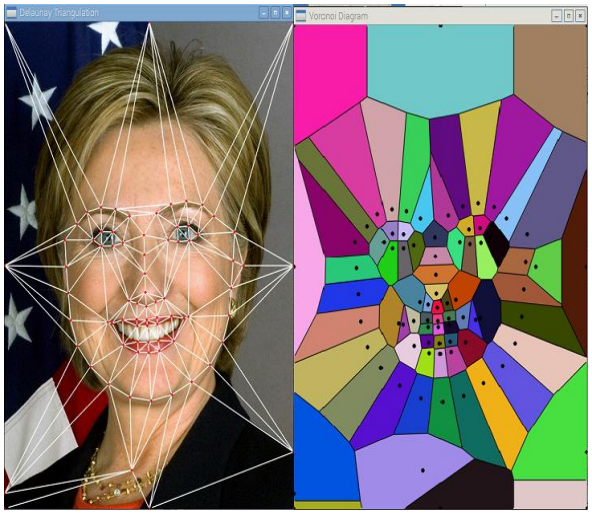
Softwares:

Python

OpenCV on the SD-card

2 Analyse de visages sur Raspberry Pi

Morphing



Control points required

Keypoint detection and matching

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

img = cv2.imread('./alex.jpg')
```

SIFT

```
# Initiate STAR detector
sift = cv2.xfeatures2d.SIFT_create(10)

# find the keypoints with SIFT
kp = sift.detect(img,None)
```

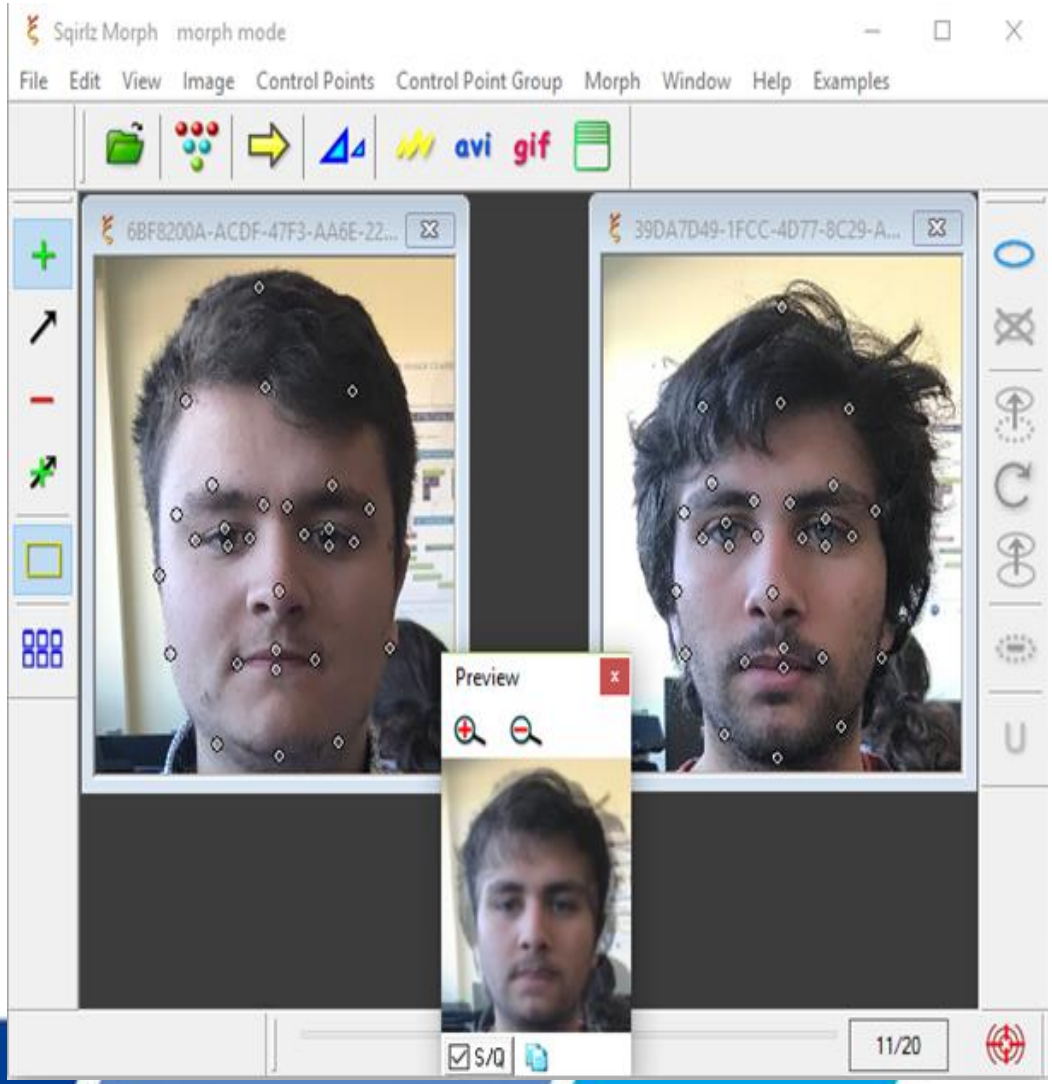
SURF

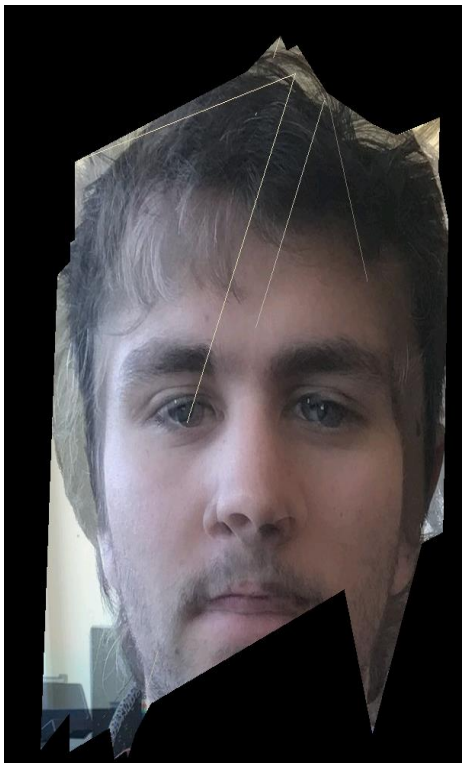
```
# Initiate STAR detector
surf = cv2.xfeatures2d.SURF_create(400,5,5)

# find the keypoints and compute the descriptors with SURF
kp, des = surf.detectAndCompute(img,None)
```

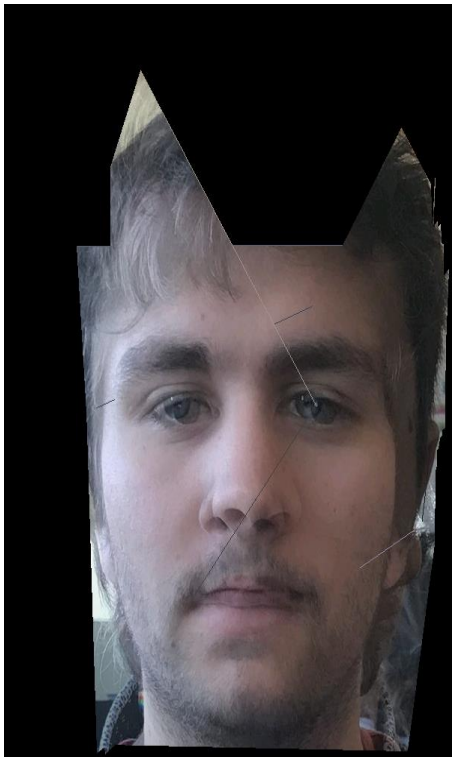
```
# draw only keypoints location,not size and orientation
img2 = cv2.drawKeypoints(img,kp,None,color=(0,255,0), flags=4)
```

Morphing,

With manual ‘control’ keypoint selection on Sqirlz Morph

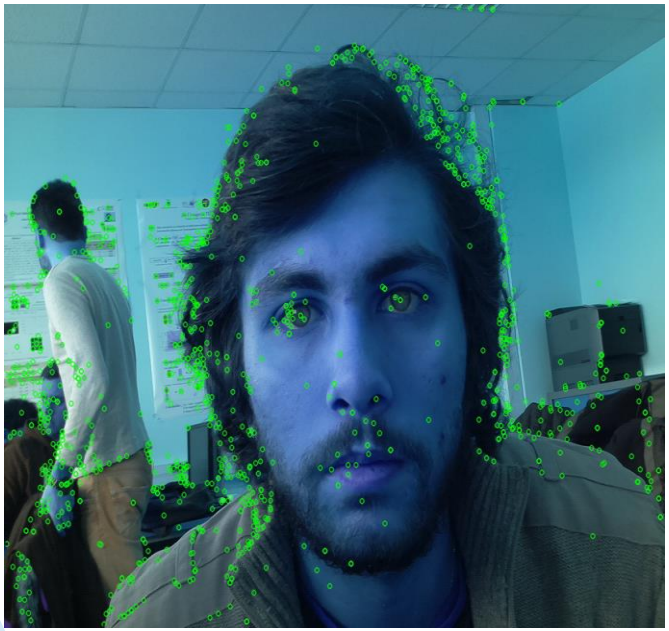


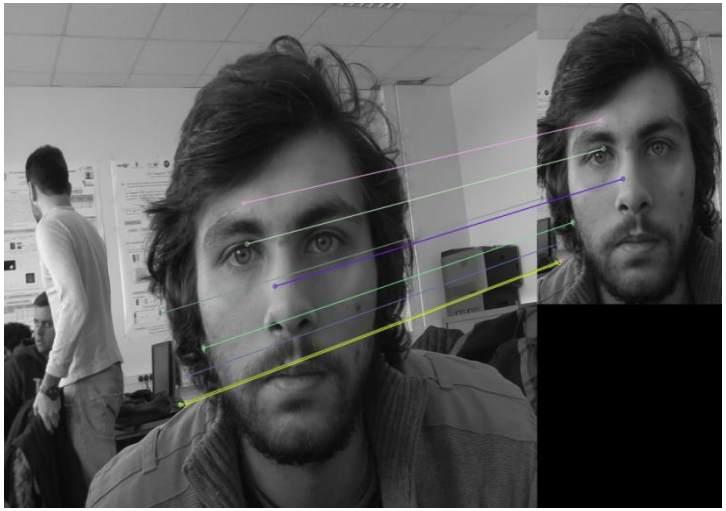




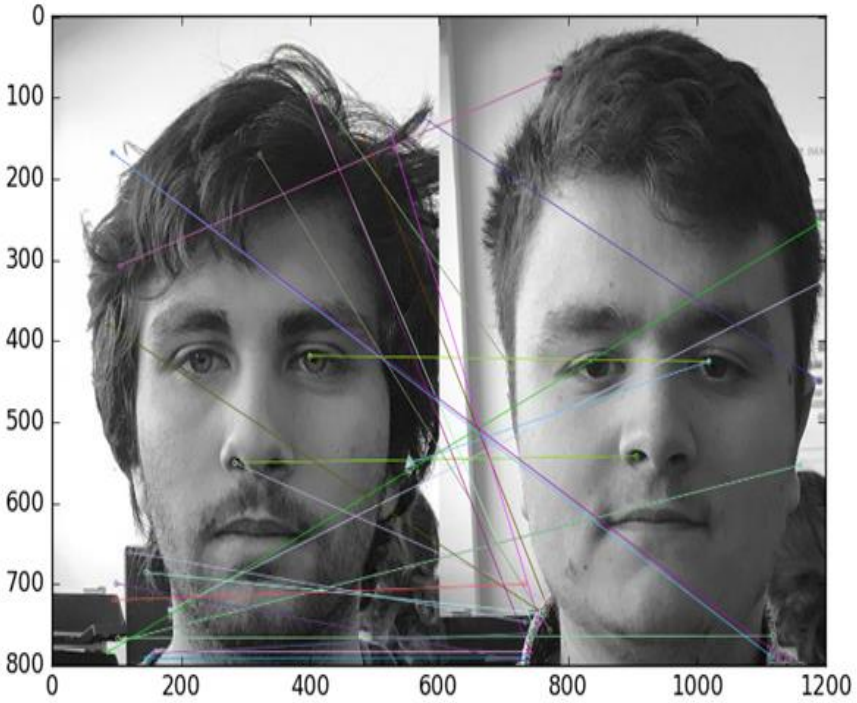
- Image processing applications

Keypoint detection and matching





Même personne: OK

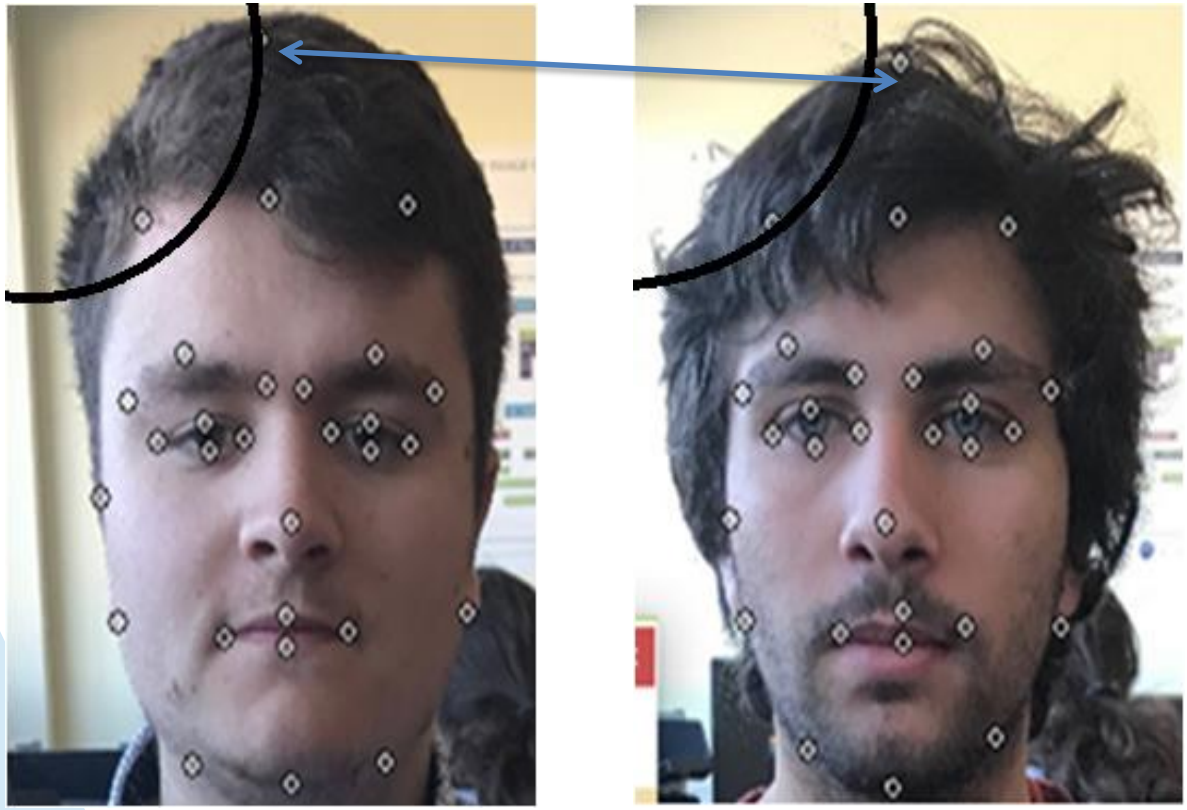


Personnes différentes: Problème de matching

Automatic Keypoint detection for Morphing

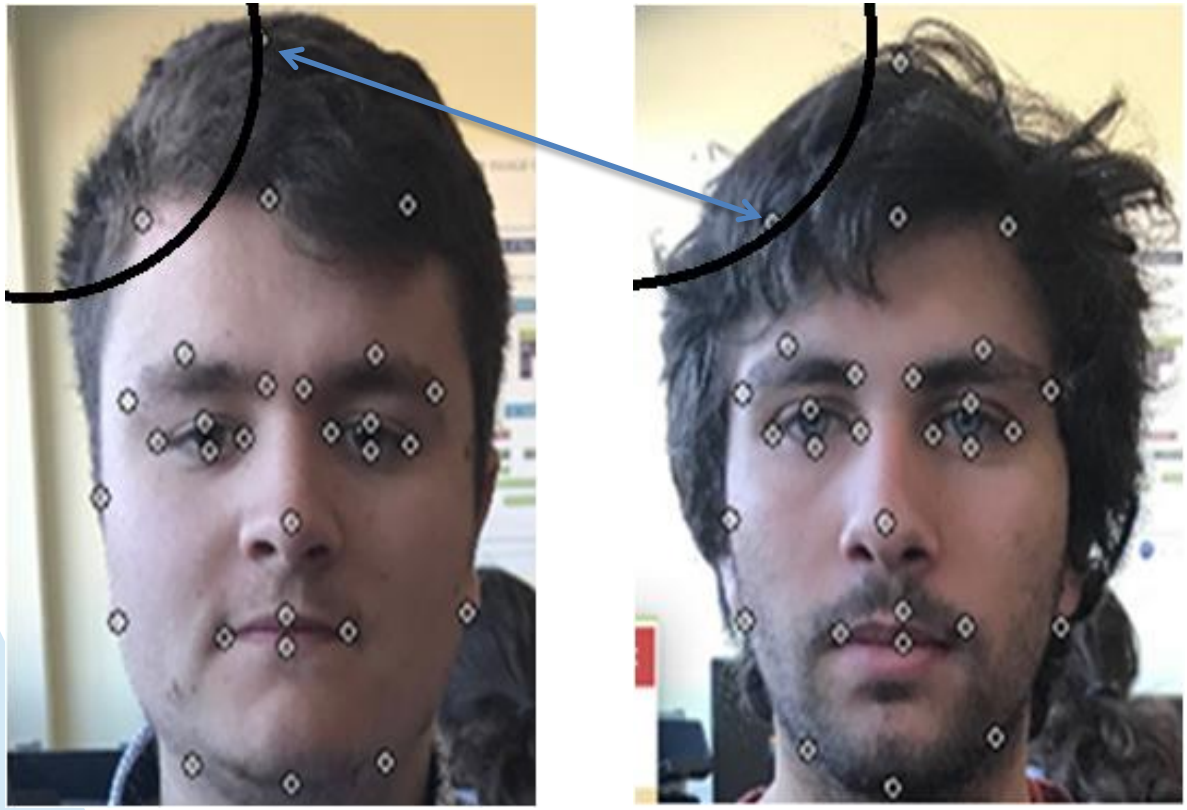
Solution envisagée: critère de distance au coin supérieur gauche

Automatic Keypoint detection for Morphing



Solution envisagée: critère de distance au coin supérieur gauche

Automatic Keypoint detection for Morphing

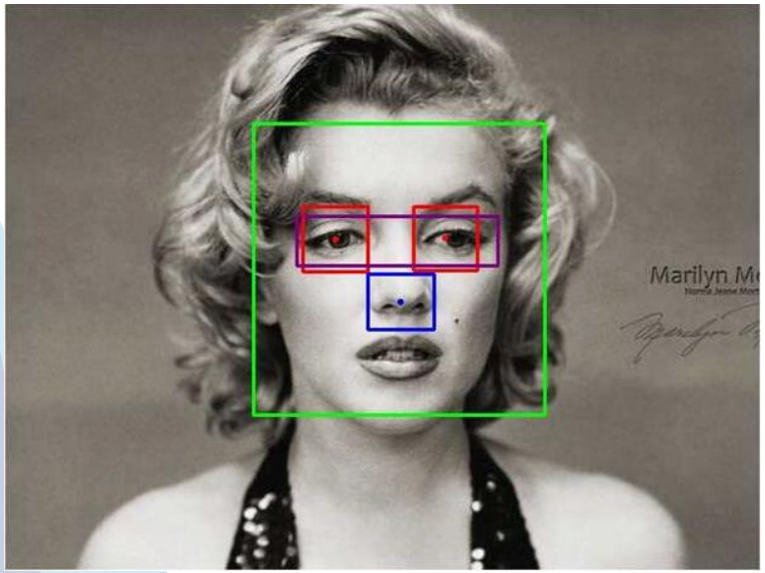


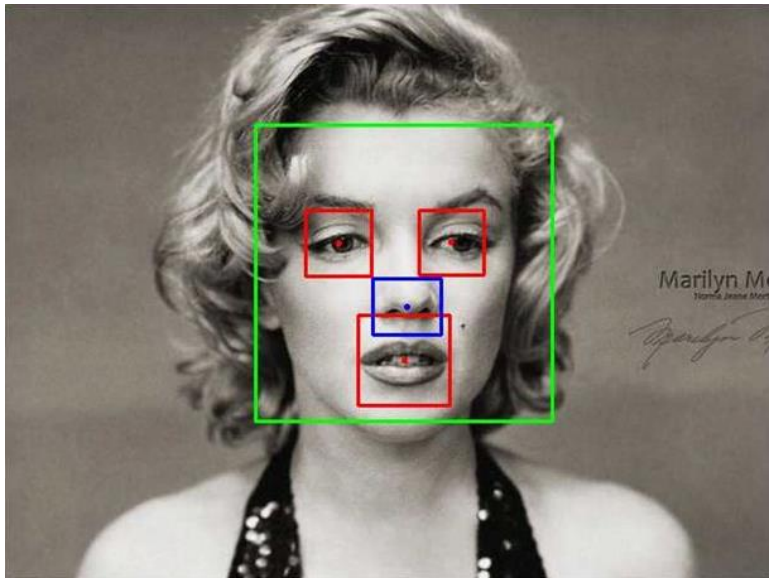
Solution envisagée: critère de distance au coin supérieur gauche

Régions d'intérêt dans les visages

Idea: restrict the keypoints to some regions of interest

```
faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eyeCascade = cv2.CascadeClassifier('haarcascade_eye.xml')
mouthCascade = cv2.CascadeClassifier('haarcascade_mcs_mouth.xml')
noseCascade = cv2.CascadeClassifier('haarcascade_nose.xml')
```





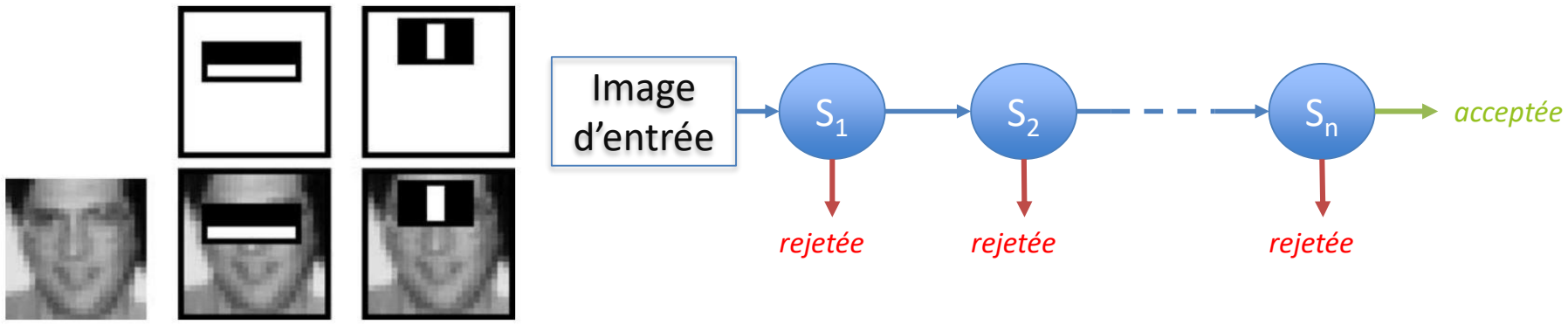
Méthode de Viola-Jones

Détection d'objet:

Localisation automatique fondée sur des caractéristiques connues a priori

Viola-Jones detector:

P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features",2001.



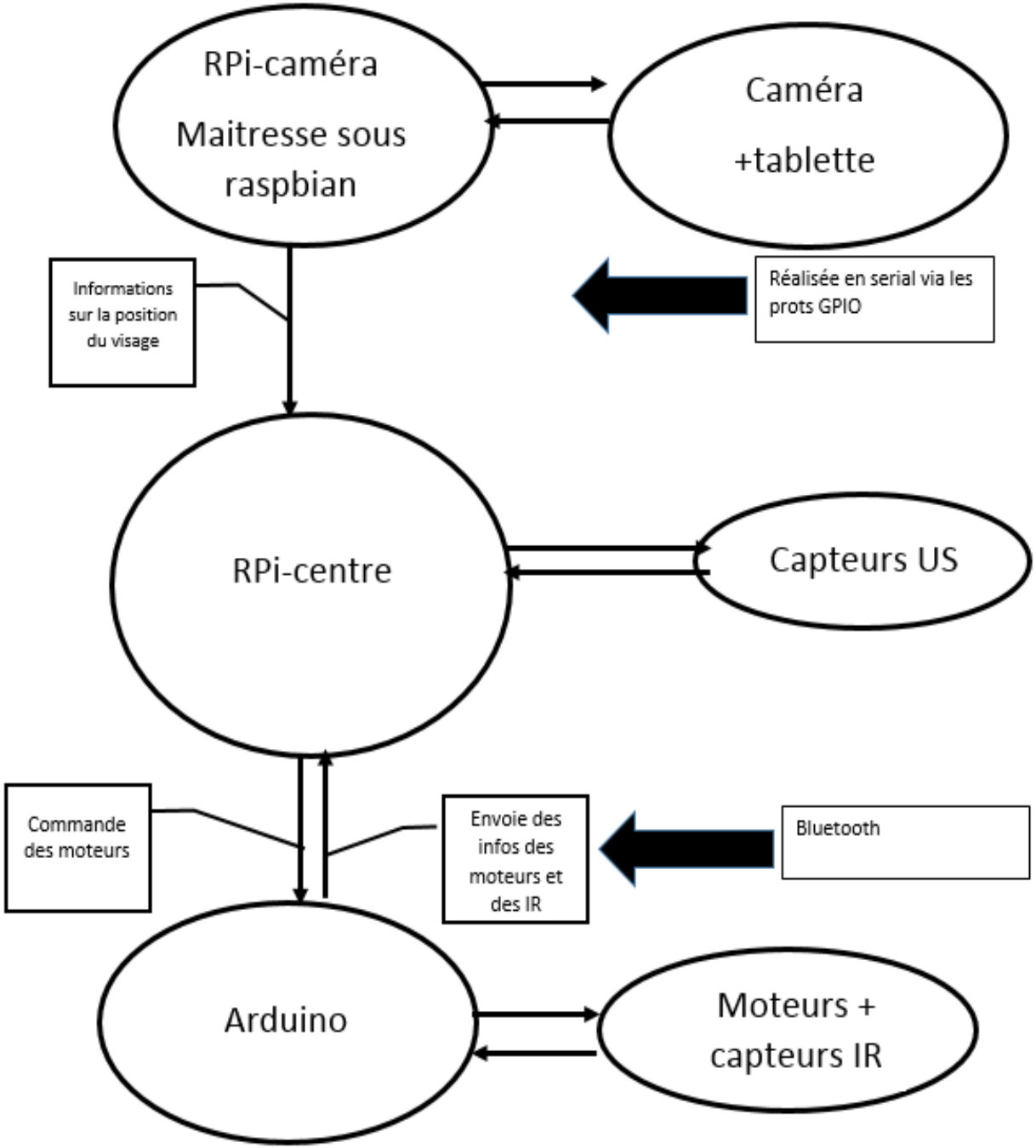
- Application d’intelligence artificielle

Robot UBBO Axyn:

Robot d’accueil

http://www.axyn.fr/


Servomoteur contrôlant la position de la tablette
Picaméra
RPi-Caméra (au dos de la tablette)
Capteurs ultrasons
RPi-Centre
Arduino (sous la coque)


RPi-caméra
Maitresse sous raspbian
Caméra
+tablette
Informations sur la position du visage
Réalisée en serial via les prots GPIO
RPi-centre
Capteurs US
Commande des moteurs
Envoie des infos des moteurs et des IR
Bluetooth
Arduino
Moteurs + capteurs IR

La RPI-caméra montée

Communication entre Rpis

- Les différentes méthodes
- GPIO
- Ethernet
- Bluetooth
- ROS ?

Communication entre Raspberries

- La méthode de communication

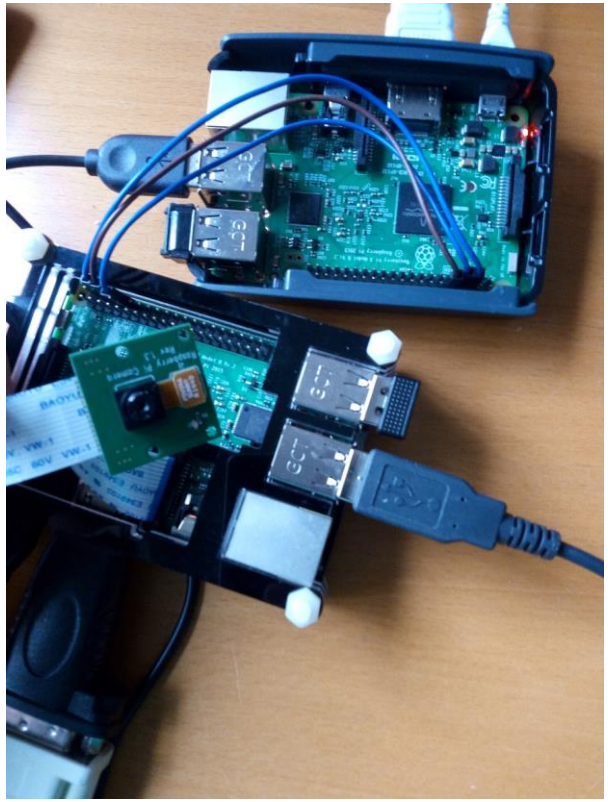
```
msg=''
for i in L:
    if len(i)<3:
        i=(3-len(i))*'0'+i
    msg+=i
msg=msg+'\n'
ser.write(msg.encode())
```

Code de communication (partie émission)

```
x=ser.readline()
if x!= info and x!='':
    print("Nouvelle donnee recue")
    print(x)
    print("analyse...")
    x1=int(x[0]+x[1]+x[2])
    y1=int(x[3]+x[4]+x[5])
    if x[6] == '-' :
        z1=int(x[6]+x[7]+x[8]+x[9])
    else :
        z1=int(x[6]+x[7]+x[8])
    info = x
```

Code de communication (partie réception)

Transfert du Rpi cerveau au Rpi intermédiaire:
GPIO



Connection en GPIO entre 2 RPi

- La méthode de communication

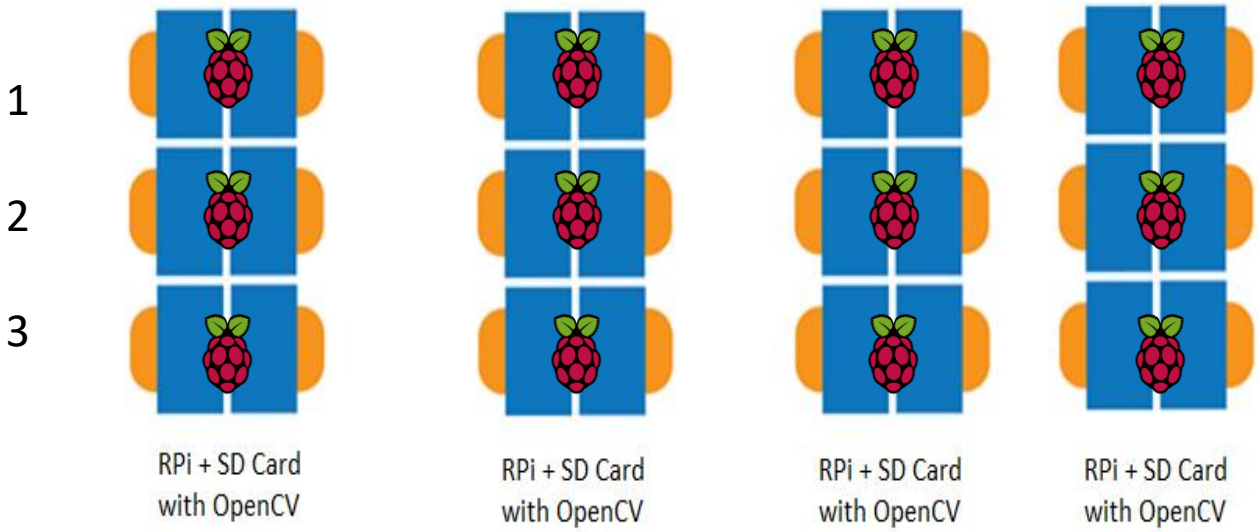
```
msg=''
for i in L:
    if len(i)<3:
        i=(3-len(i))*'0'+i
    msg+=i
msg=msg+'\n'
ser.write(msg.encode())
```

Code de communication (partie émission)

```
x=ser.readline()
if x!= info and x!='':
    print("Nouvelle donnee recue")
    print(x)
    print("analyse...")
    x1=int(x[0]+x[1]+x[2])
    y1=int(x[3]+x[4]+x[5])
    if x[6] == '-' :
        z1=int(x[6]+x[7]+x[8]+x[9])
    else :
        z1=int(x[6]+x[7]+x[8])
    info = x
```

Code de communication (partie réception)

Principe de travail en TP



Équipes de 2 binômes d'étudiants
Pour chaque binôme: 1 Rpi, accès au réseau local
Pour chaque équipe: 1 carte SD avec OpenCV, 1 camera

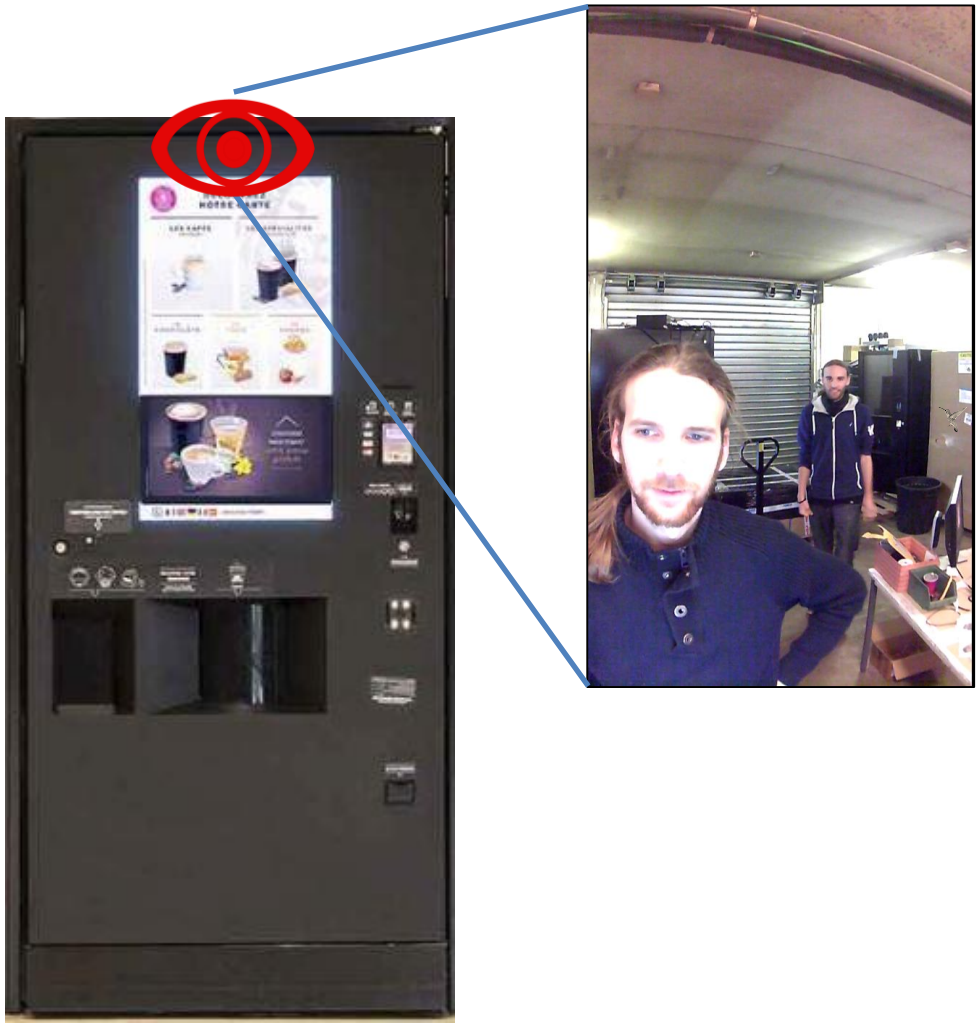
2 Détection et suivi de visages

IntuiSense Technologies

Daltys Coffee

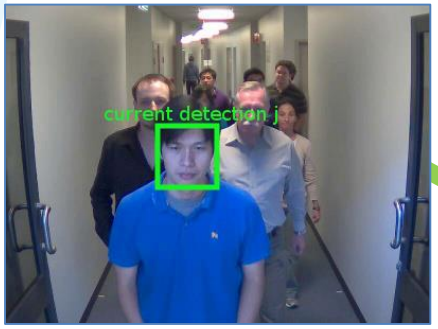
Contraintes industrielles:

- Fond inconnu
- Illumination incontrôlée
- Visages pas de face
- Embarqué: *programme exécutable autonome sur Céléron*
- Temps réel

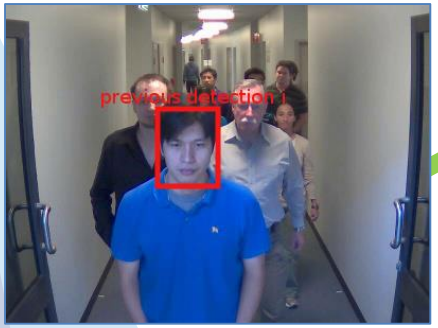


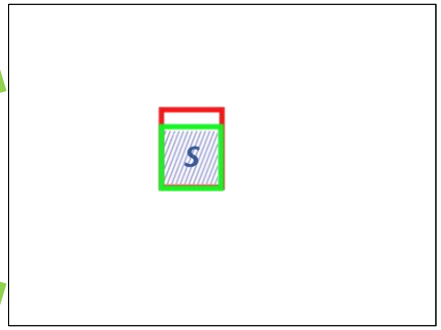
Correspondance de détections

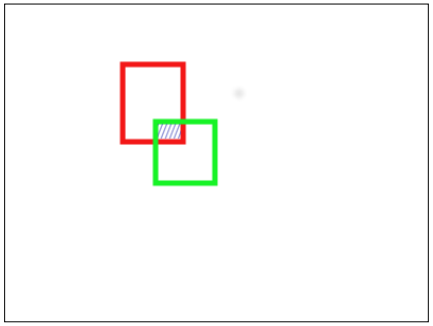
Trame t



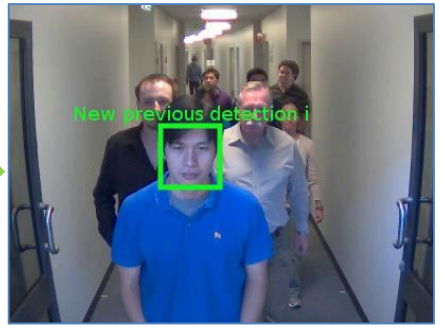
Trame t-1







Correspondance trouvée

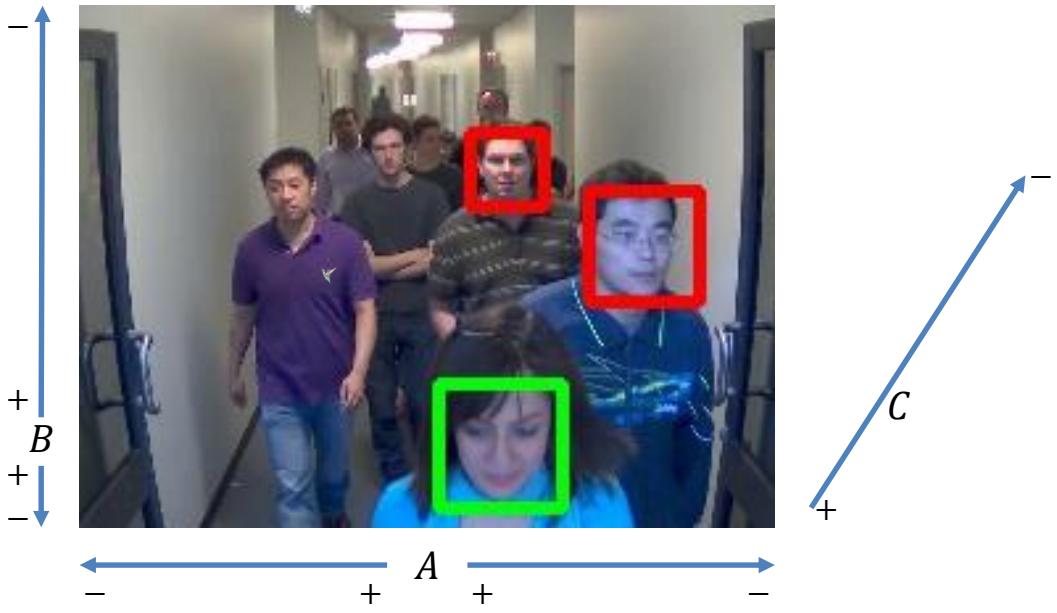


Pas de correspondance

Objectif:

Déterminer quelle personne détectée a la plus forte probabilité d'être l'utilisateur

$$P(D = utilisateur) = A \times B \times C$$

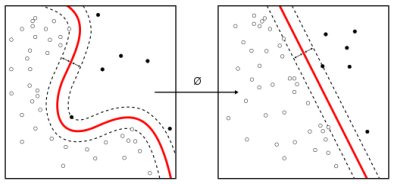


Key points

Reconnaissance de genre:

Attribuer automatiquement une étiquette *'homme' / 'femme' à une image de visage*

Matériel et méthodes:



- Un algorithme de machine learning: machines à vecteurs de support

- Une base de données d'images (FERET)

- Une representation d'images (Local Binary Patterns)

Lien avec l'optimisation

Fonction objectif:

Taux de fausse reconnaissance (False recognition rate FRR)

$$FRR = \frac{nb_{fr}}{nb\_testing\_images} \qquad (fitness)$$

Méthodologie SVM, paramètres à estimer:

$$\min_{w,\varsigma_1,\ldots,\varsigma_l} \left[\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \varsigma_i \right] \qquad with \quad C \in \mathbb{R}^{+*} \qquad (K_1)$$

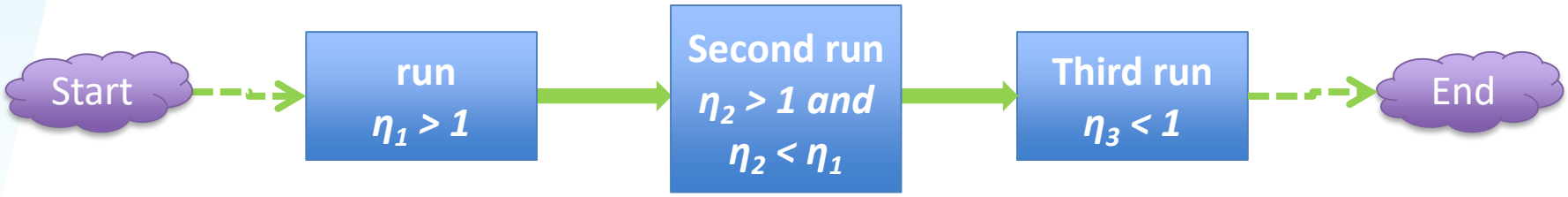
Noyeau Gaussien (RBF) :

$$K(x_i, x_j) = \exp\left(-\frac{\|\varphi(x_i) - \varphi(x_j)\|^2}{2\gamma^2} \right) \qquad with \quad \gamma \in \mathbb{R}^{+*} \qquad (K_2)$$

C	γ	FRR
100	1,00E-03	50,00%
100	1,00E-04	18,50%
1	1,00E-06	26,10%
10	1,00E-06	13,60%

Lien avec l'optimisation

GWO adaptif, version continue : expression de 'a' adaptée sur trois étapes



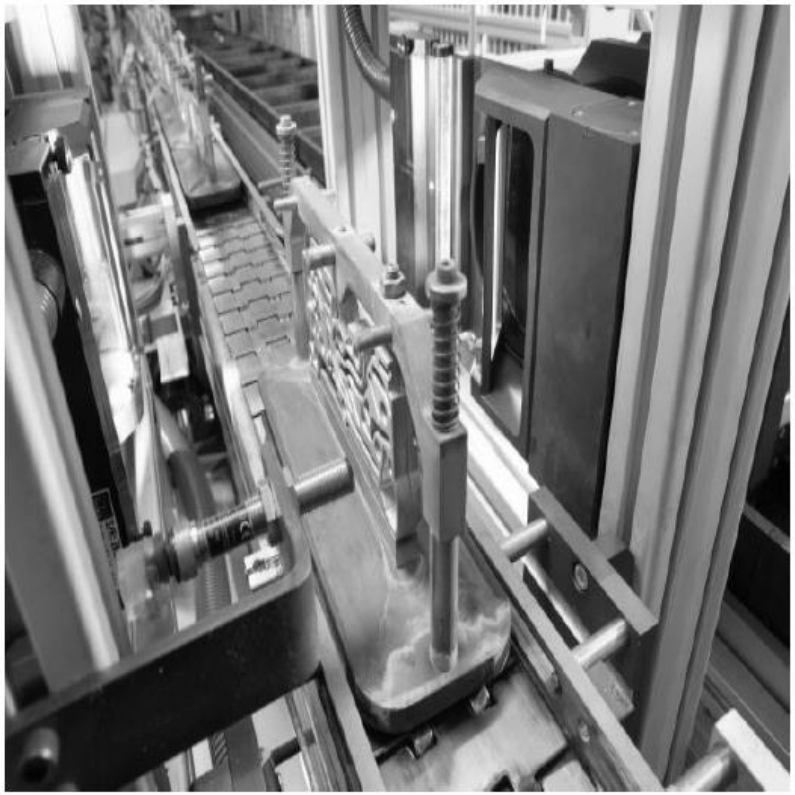
$\eta_1 = 8$ $\quad T_{max} = 15 = (3 \times 5)$

$\eta_2 = 2$ $\quad Q = 20$

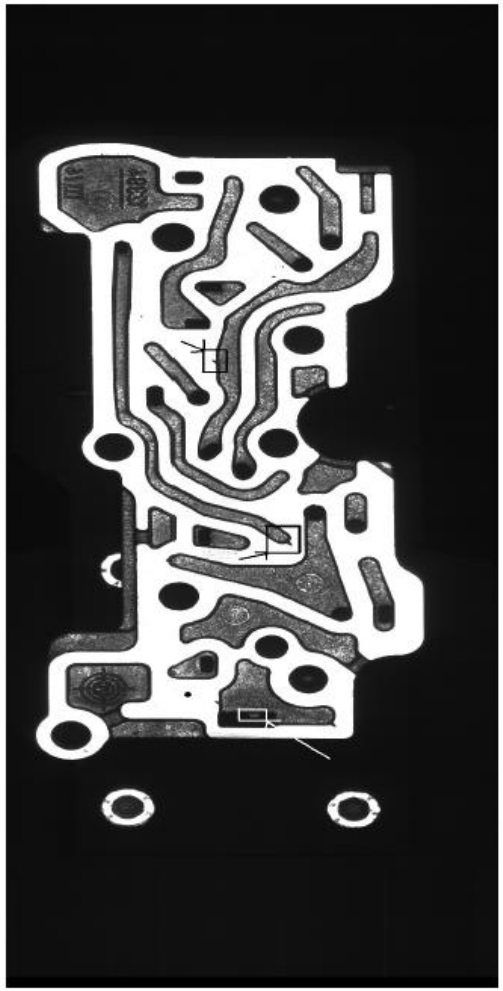
$\eta_3 = 0{,}5$

Method	*Best (γ,C)*	*100 - FRR*	*Time (sec.)*
PSO	(6,7E-07; 600)	91,40%	1970
GWO	(1,6E-07; 800)	91,50%	1970
mGWO	(5,0E-08, 83)	91,90%	1970
aGWO	(4,9e-08; 82)	91,90%	968

3 Détection de défauts, contrôle non destructif de pièces métalliques



(a)



(b)

(a) Système industriel d'acquisition d'images
(b) Pièce métallique avec défauts

Système d'acquisition

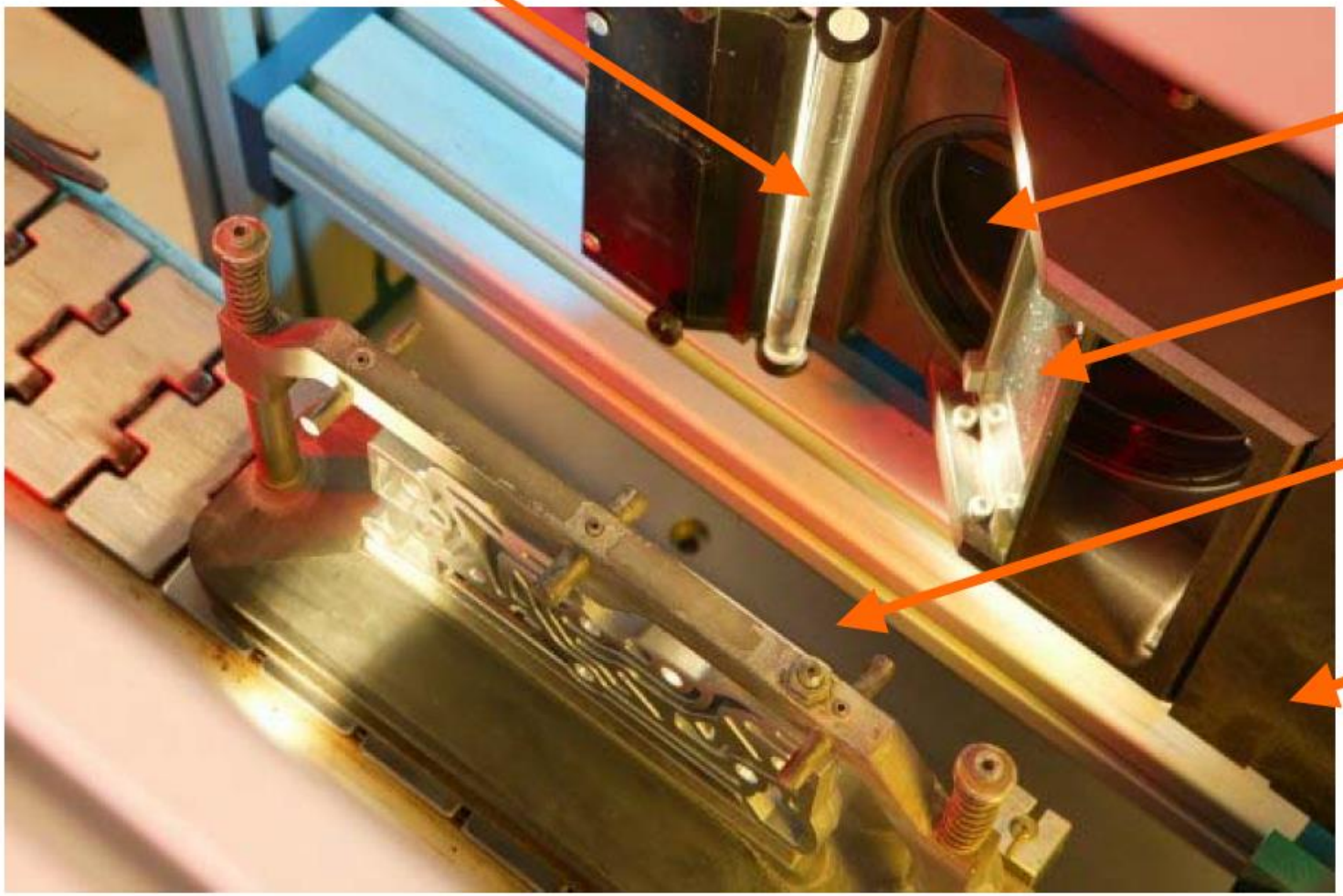
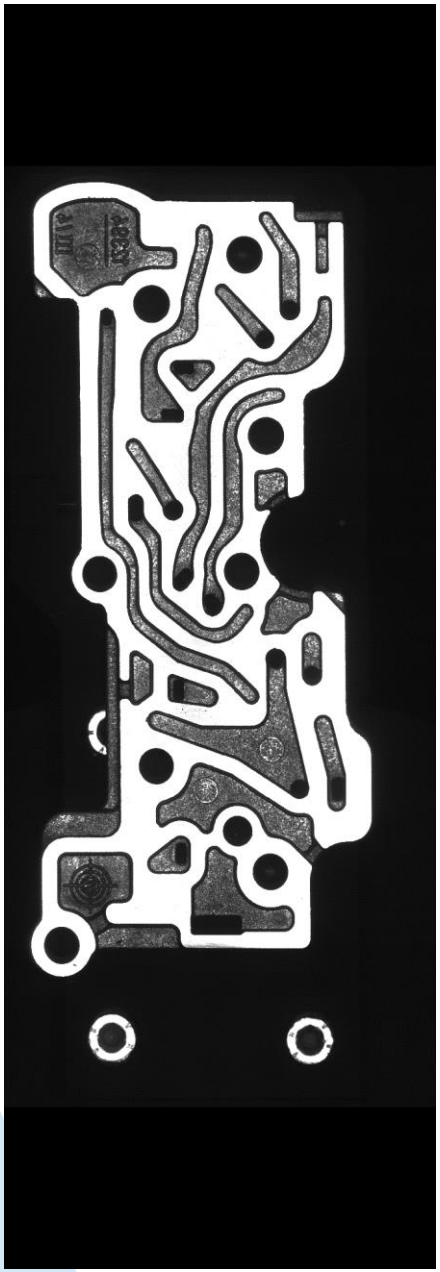
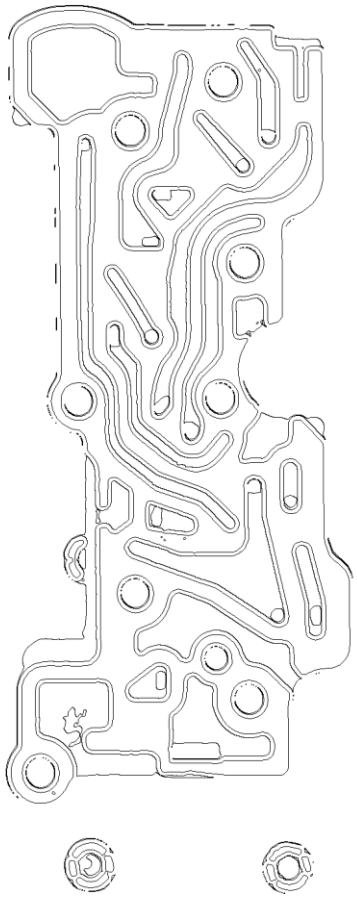


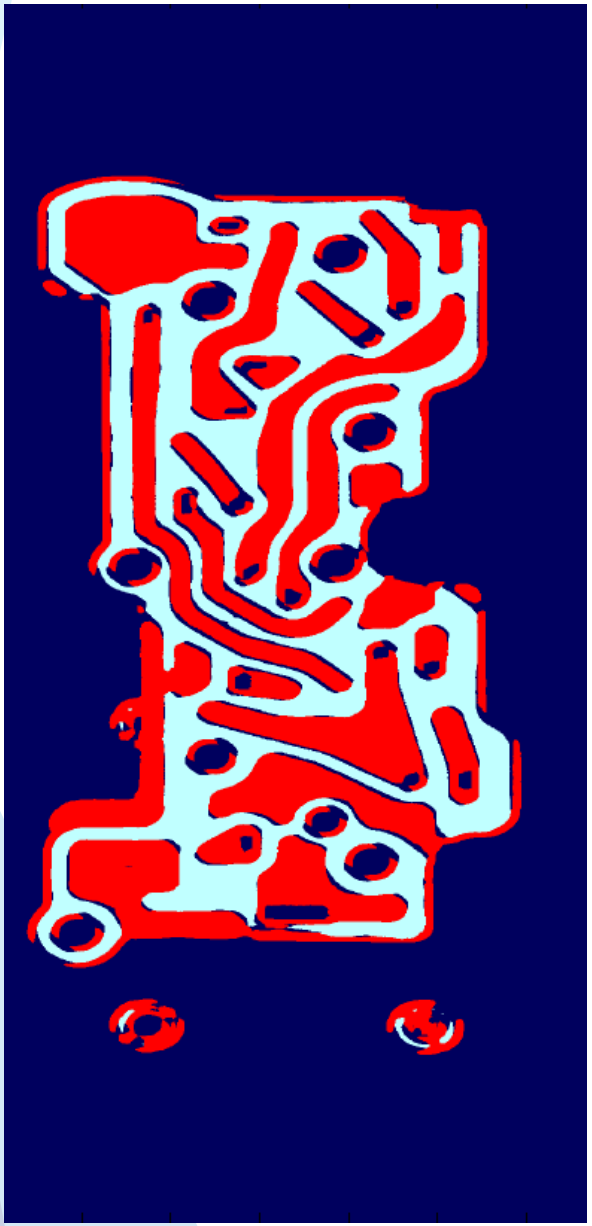
Image de référence et résultat de segmentation



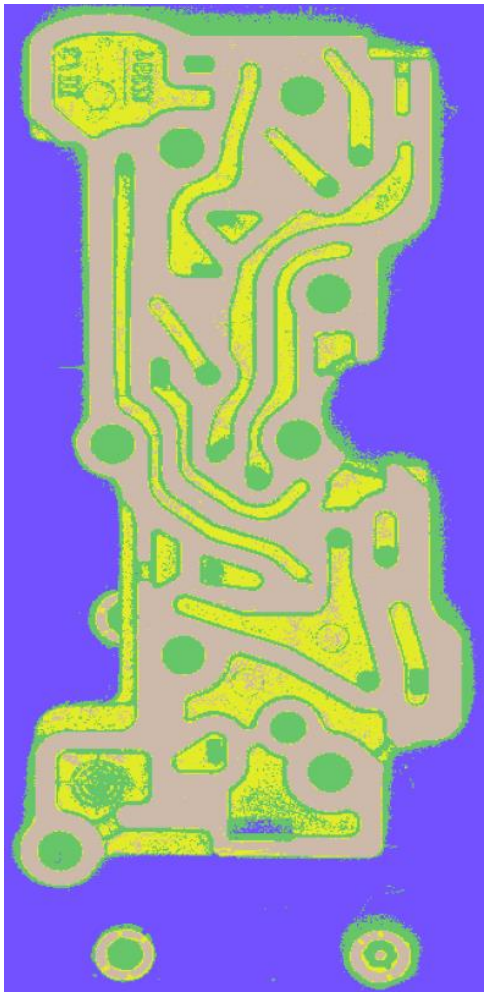


Résultats comparatifs

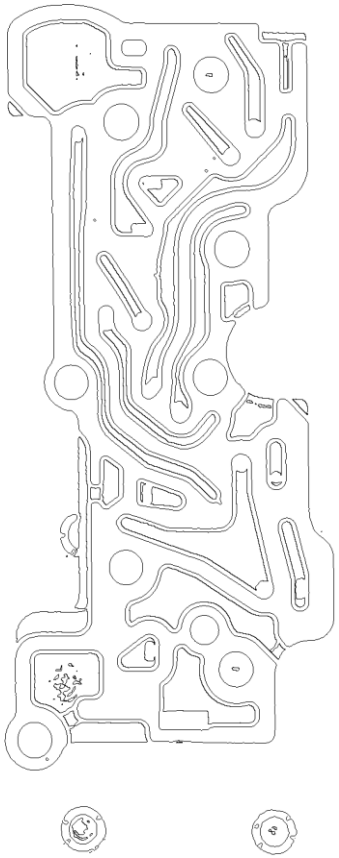
Texture

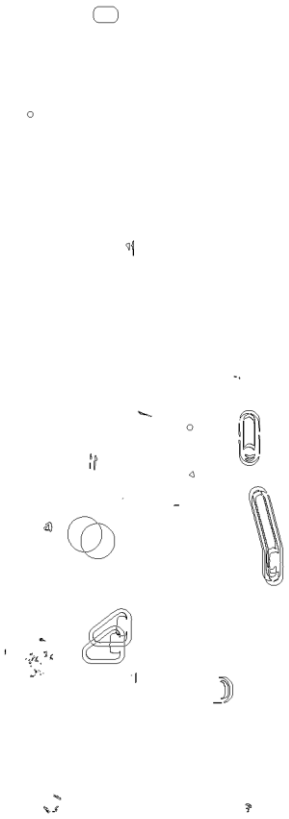


Champs de Markov

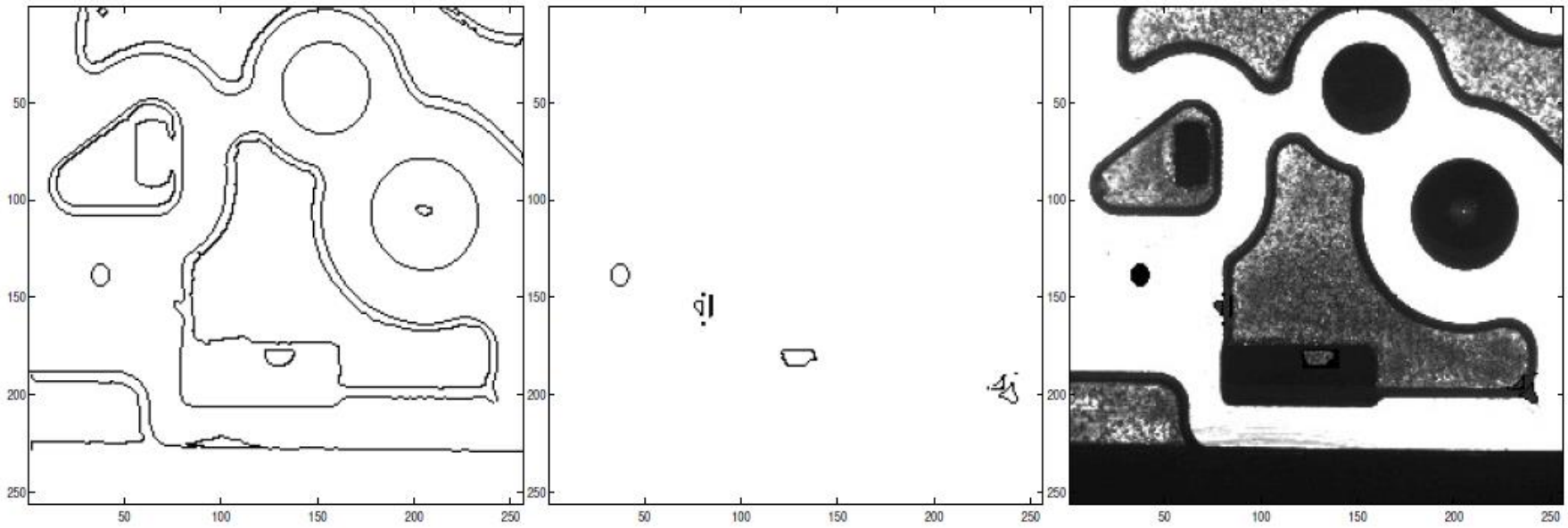


Cartes de contours et défauts



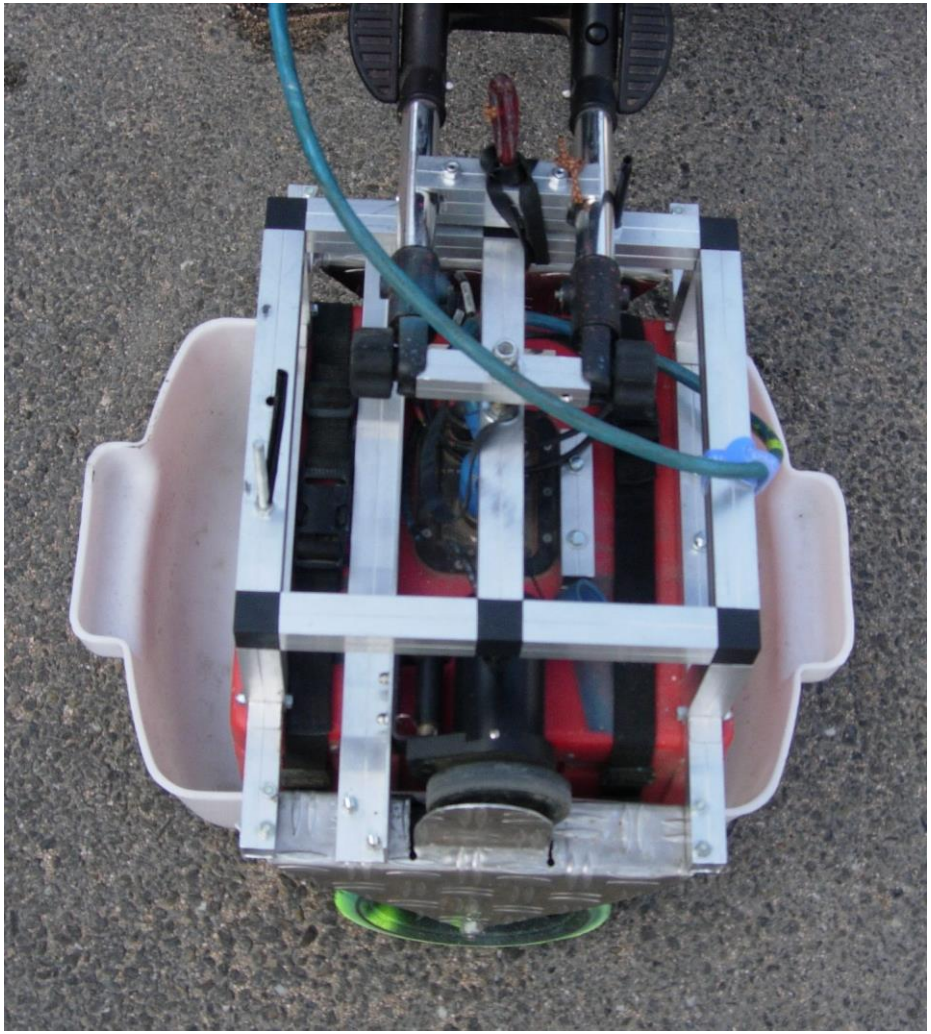


Cartes de contours et défauts



Le système d'acquisition

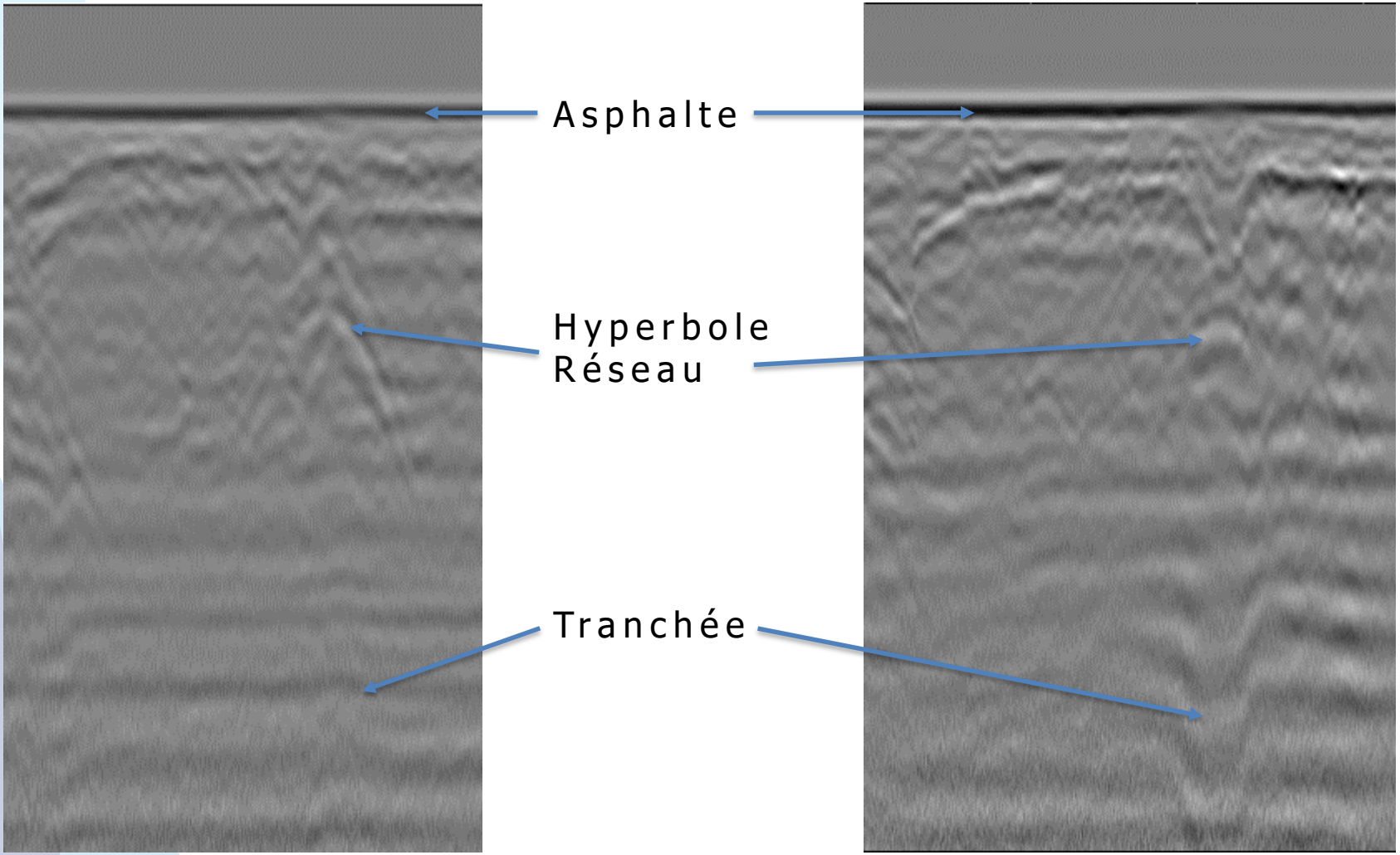
Le système d’acquisition



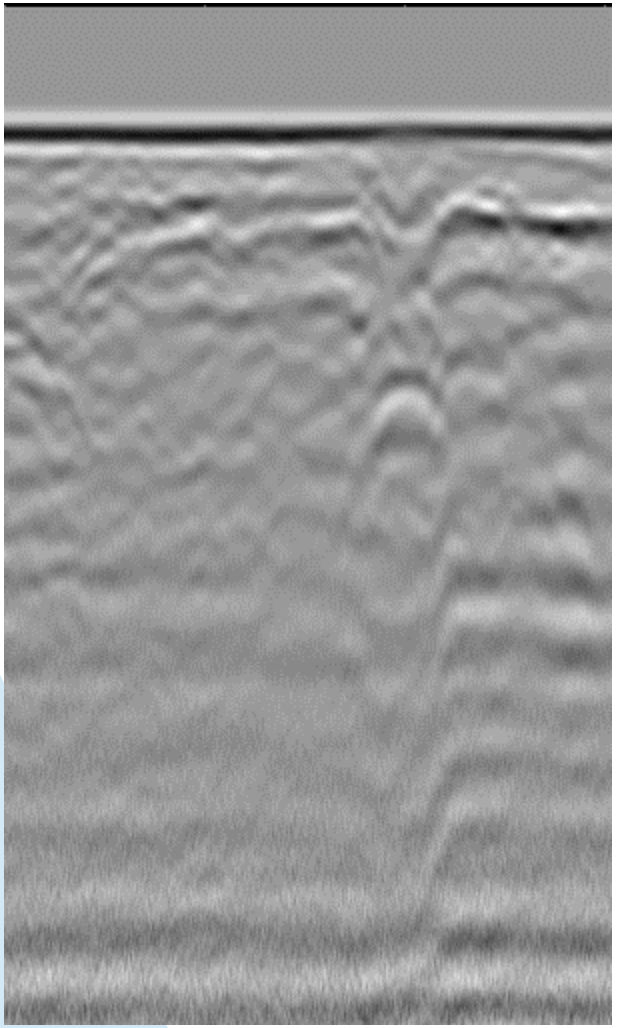
L’interface

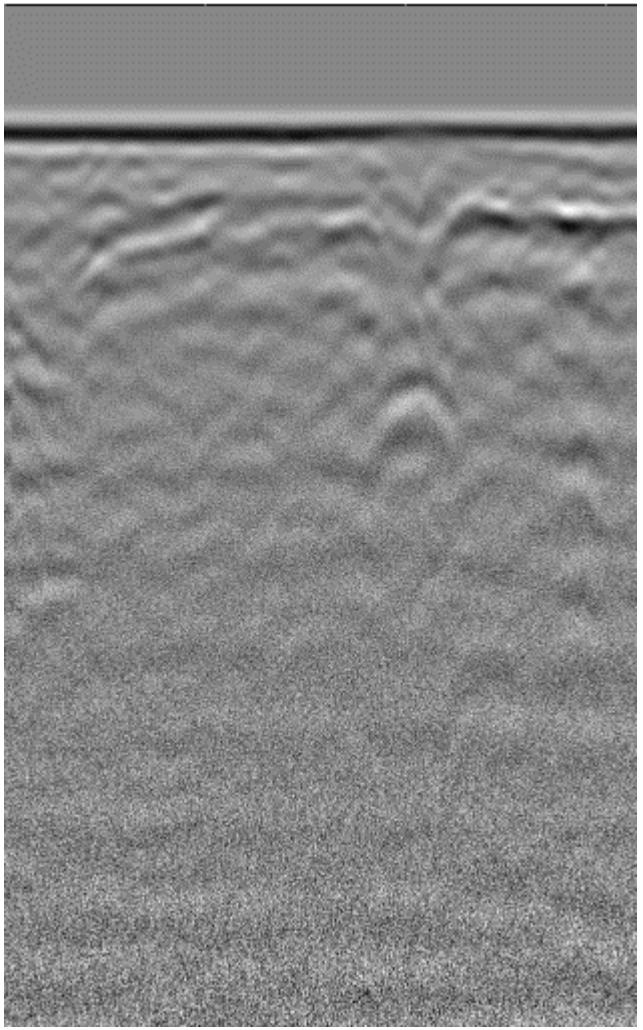
GSSI

Images acquises

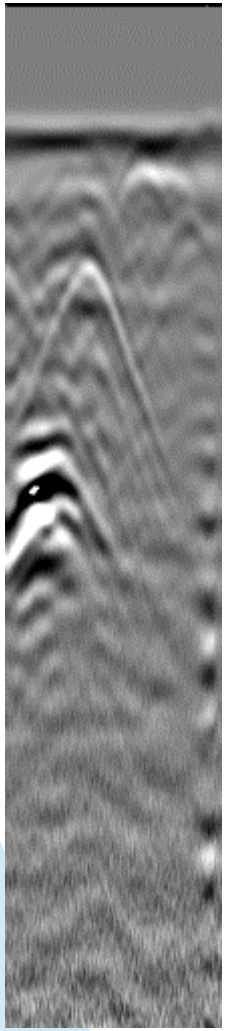


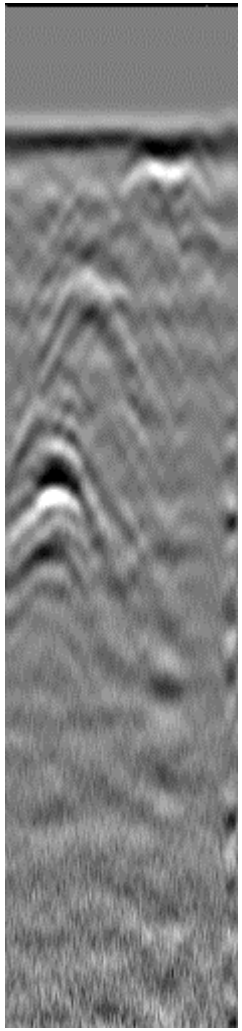
Images acquises

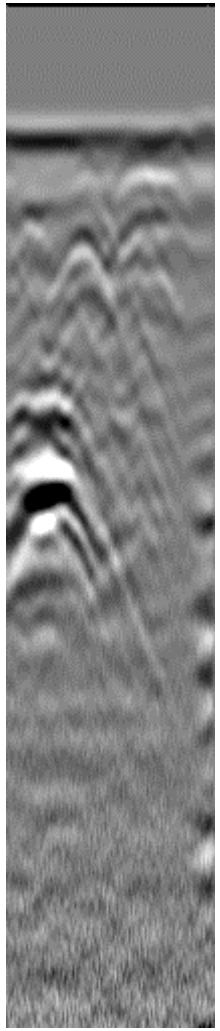




Images acquises

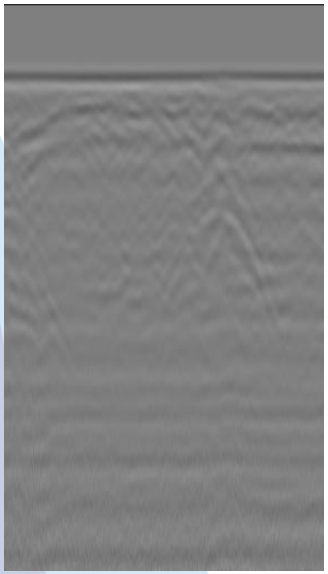


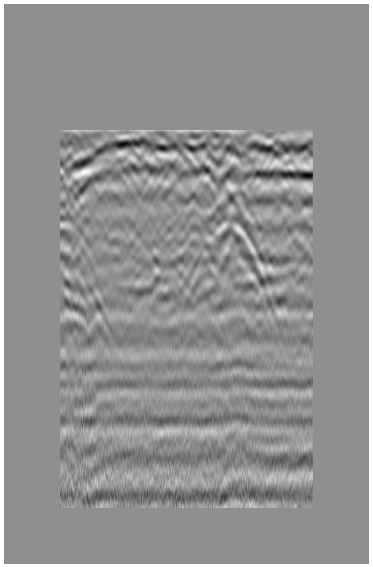


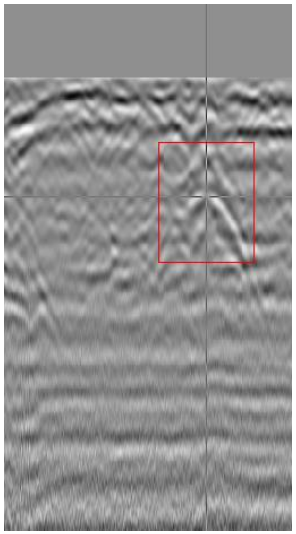


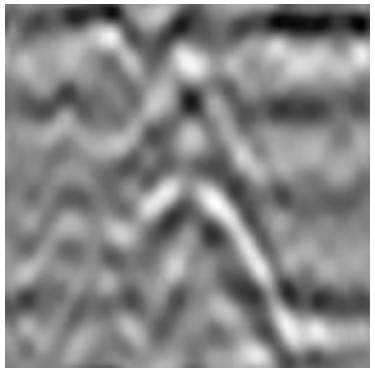
La méthodologie

Image à traiter → Prétraitement, connaissances *a priori* → Sélection *par l'utilisateur pour l'image de référence:* Région d'intérêt (ROI) à retrouver → Débruitage









La méthodologie

Mise en valeur des contours: Égalisation d'histogramme, filtre de Marr-Hildret

→

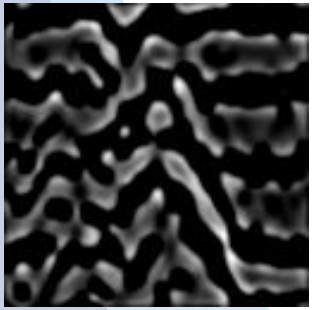
Détection des contours

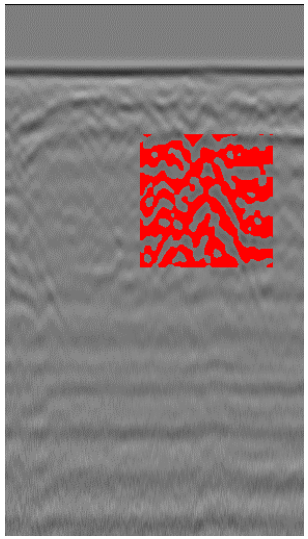
→

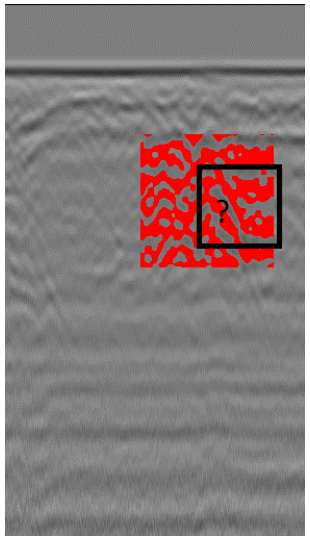
Recherche *automatique dans les images test* De la zone cible par comparaison à la ROI

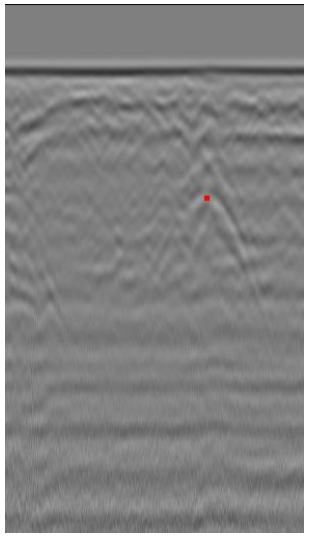
→

Résultat









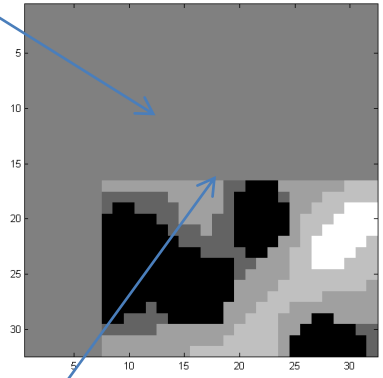
Critère de localisation automatique

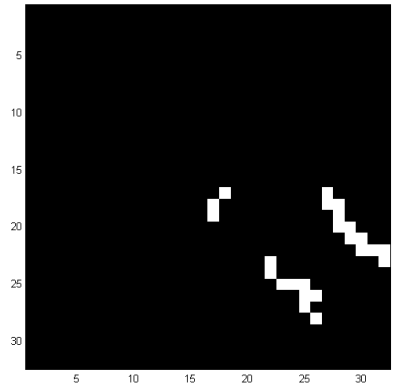
ROI centrée sur le point de référence

Sous-images tests, images critères

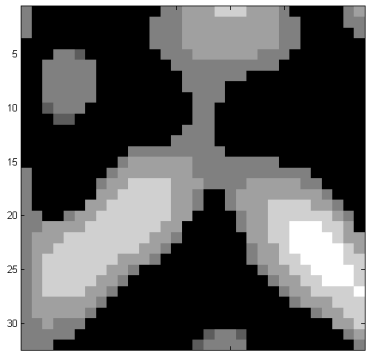
Valeurs critère

Pixels non pris en compte dans le critère

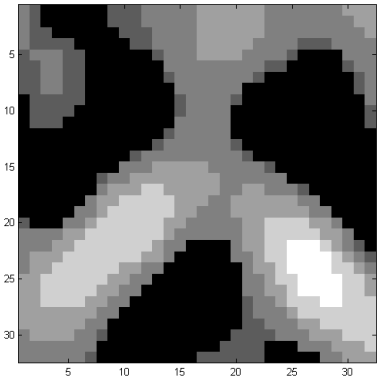


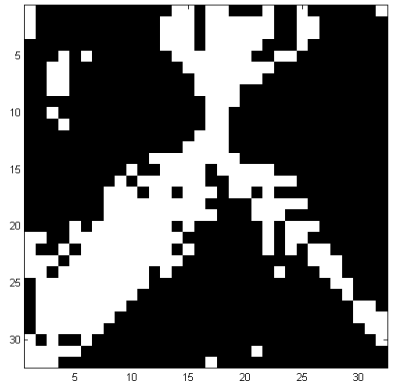


0.05



Le centre de La sous-image parcourt Tous les pixels de la ROI



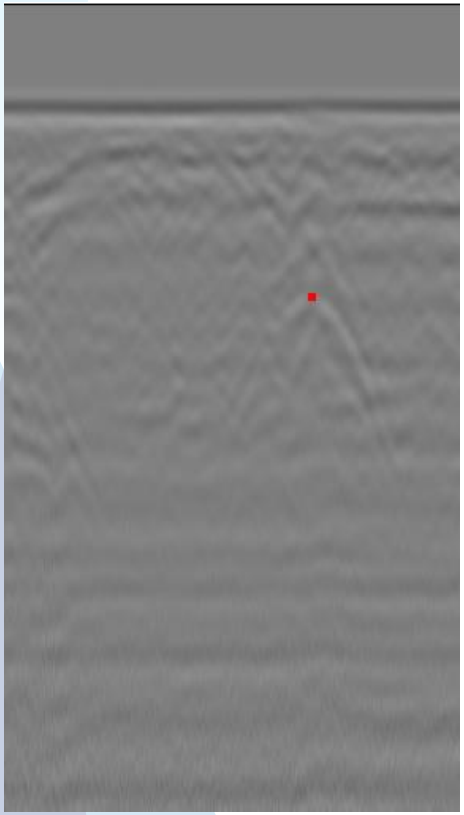


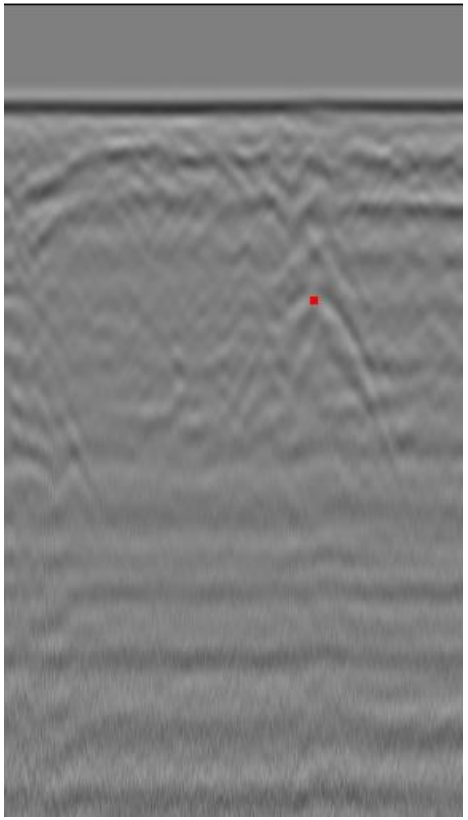
0.59

Image 1

Image n>1

Exemple 1

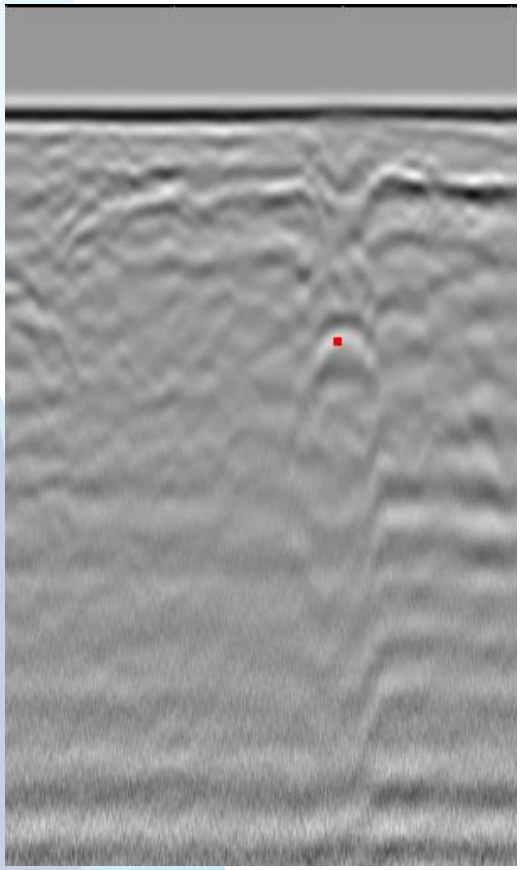


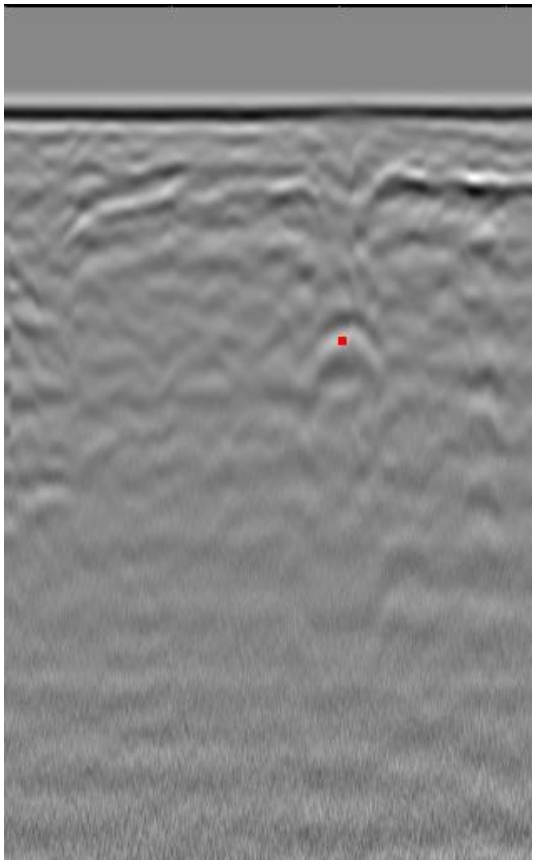


Valeur min de critère: 0.07

Valeur max de critère: 0.59

Exemple 2





Valeur min de critère: 0

Valeur max de critère: 0.67

Merci pour votre attention

Annexe

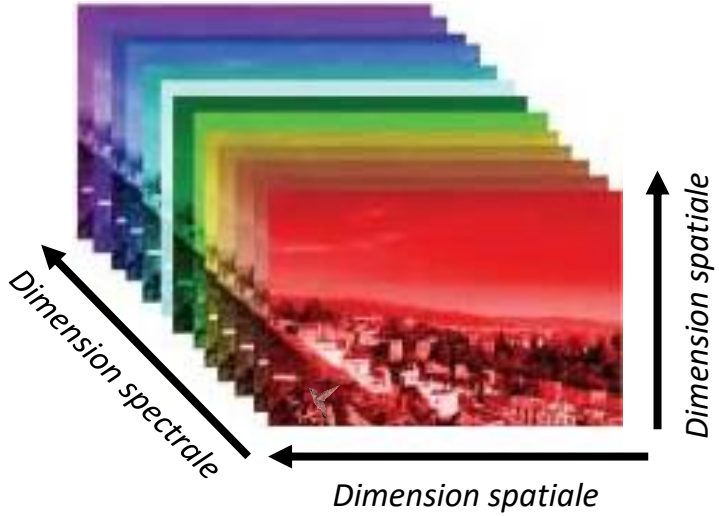
Image multispectrale:

Image composée de plus de 3 canaux

Vue originale

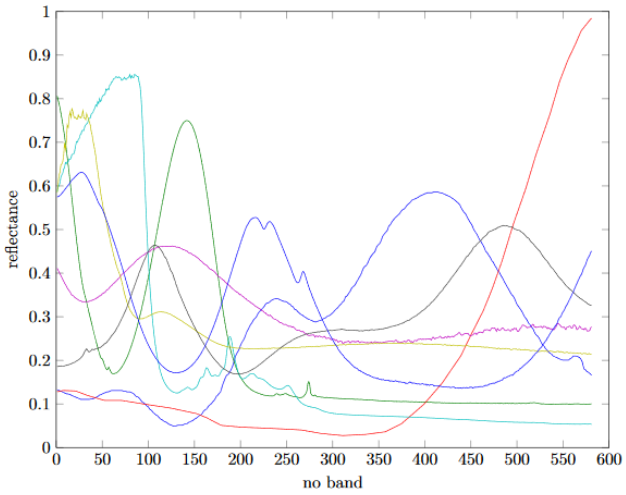
Décomposition RGB

Image multi-spectrale



Démélange de spectres:

Identification de spectres sources (forêt, sol)

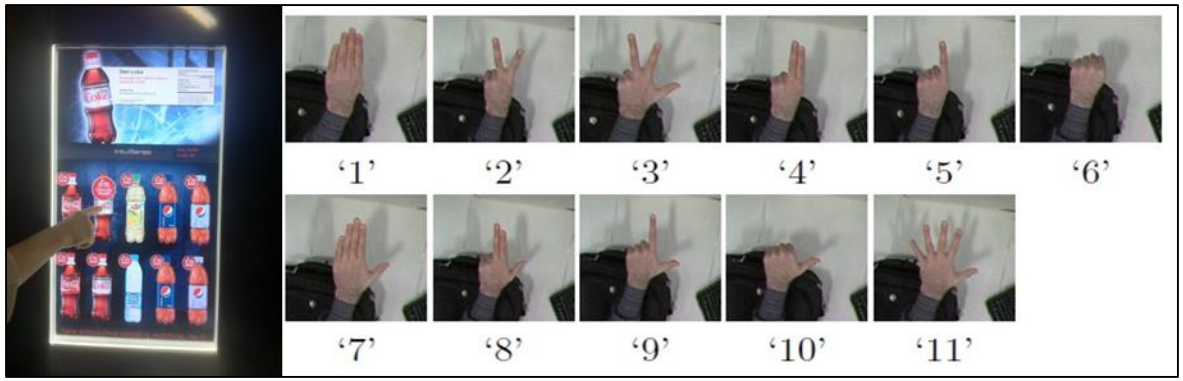


$$y = \lambda_1 s_1 + \lambda_2 s_2 + n$$

Débruitage d'image:

Procédé de suppression selective du bruit dans une image

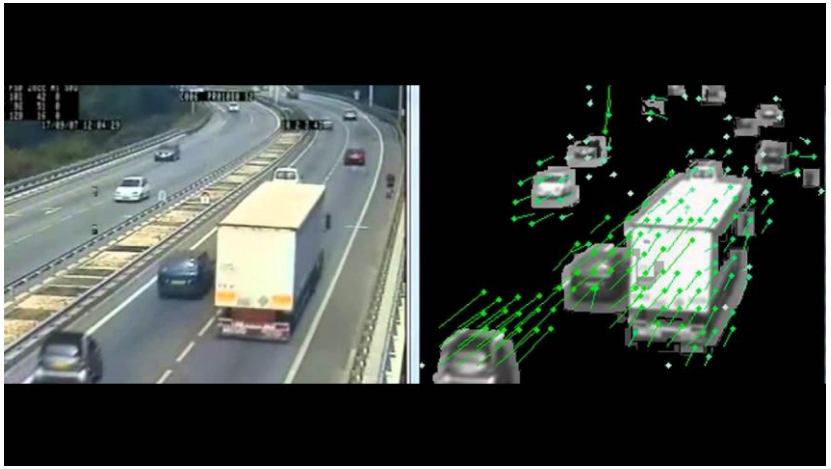
1 Postures de la main



Le flux optique

Principes

- Cible en mouvement
- Pas de contrainte sur le fond
- Pas de contrainte sur la couleur de la cible

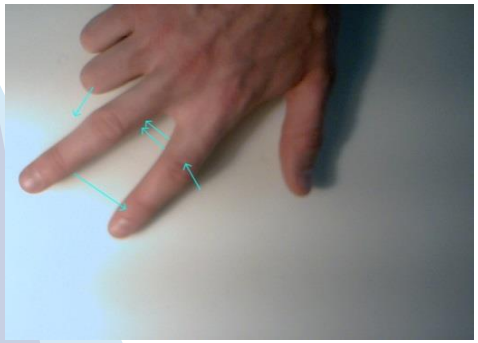


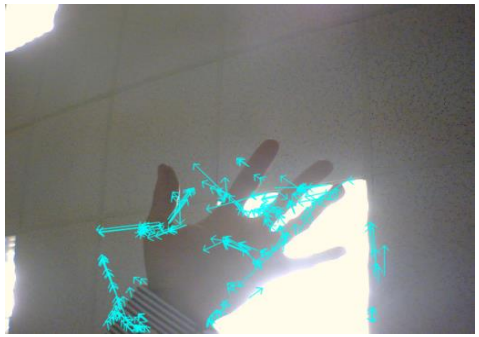
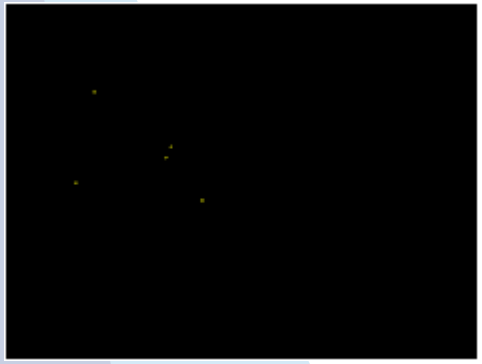
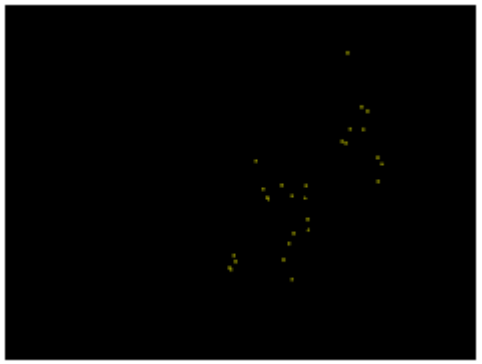
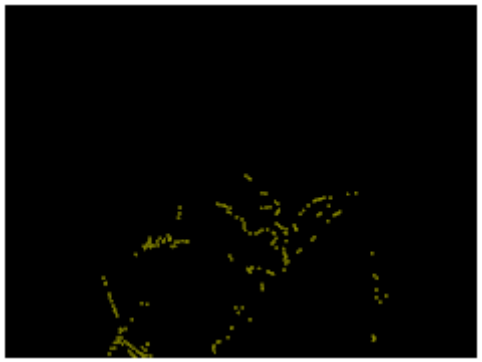
Méthode : flux optique adapté

- Faible mouvement de la main
- Fond non uniforme
- Mains blanches et mains de couleur

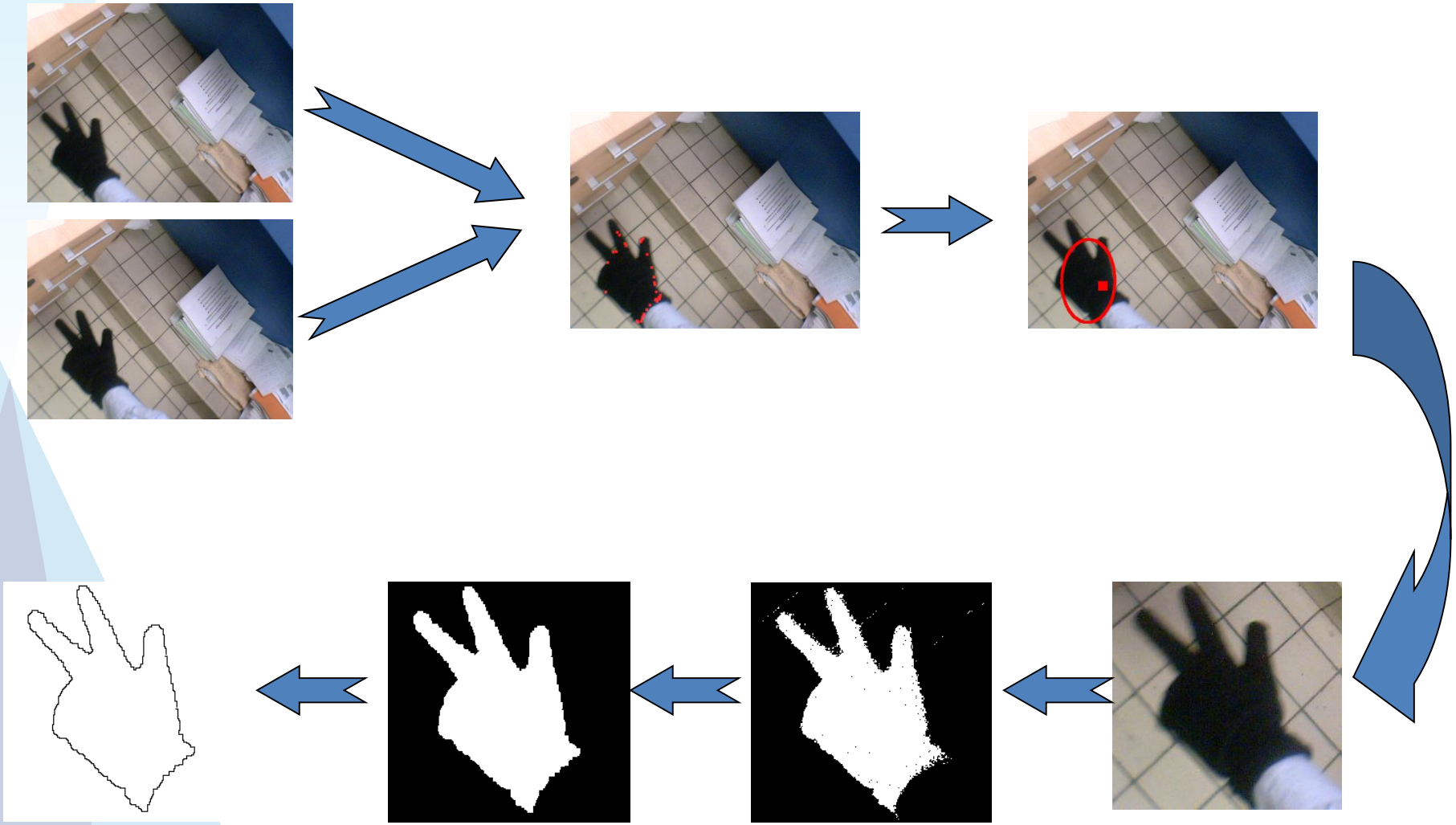
Adaptation du flux optique à la détection d'objets



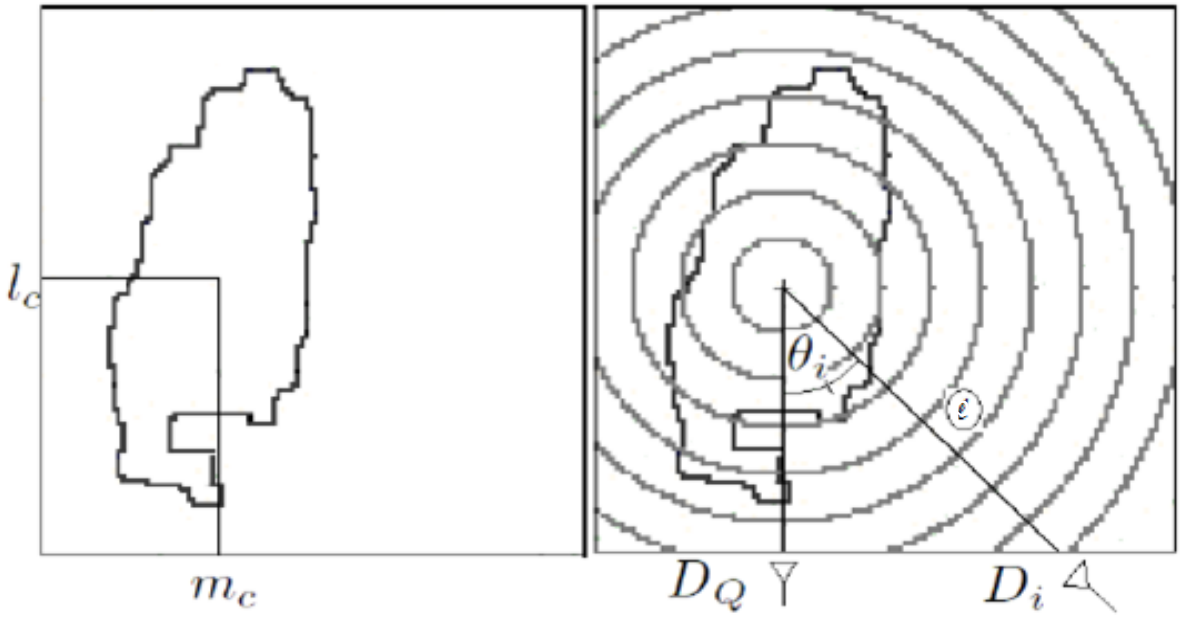



Adaptation du flux optique à la detection d'objets



Nouvelle signature



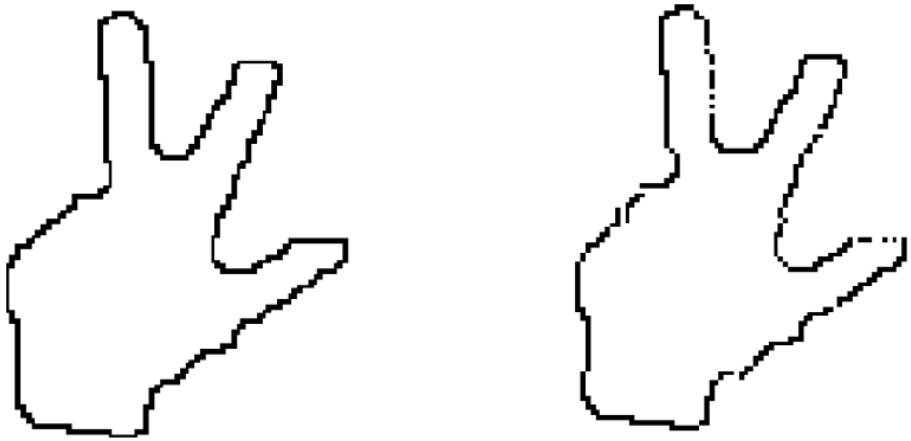
Signature: matrice Z

- Un angle θ_i, une direction D_i ➔ colonne de Z
- Un niveau p, un intervalle de distance ➔ ligne de Z
- Un pixel / direction / niveau

Nouvelle signature

Propriétés d'invariance

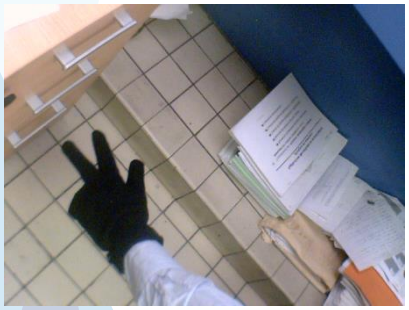
- Translation ➔ boîte englobante
- Rotation + Symétrie axiale ➔ tri par niveau
- Facteur d'échelle ➔ normalisation des $z_{p,i}$



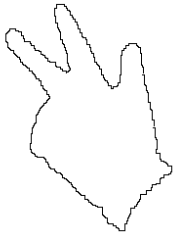
La signature conserve les détails du contour

Identification et classification

Identification : associer une image à une classe
Classification : rassembler des images en plusieurs classes



Détection



Caractérisation

$$\mathbf{Z} = \begin{pmatrix} 0 & \cdots & 0 & \delta_{1,Q} \\ \delta_{2,1} & \cdots & \delta_{2,Q-1} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \delta_{L-1,Q} \\ 0 & \cdots & \delta_{L-1,Q-1} & \delta_{L,Q} \end{pmatrix}$$

Identification et classification