

Systemes embarqués

Travaux Pratiques

Détection de visages sur Raspberry Pi

1 Introduction et premières acquisitions

1.1 Introduction

1.1.1 Objectif du TP

On souhaite lors de ce TP, faire du traitement d'image collaboratif avec Raspberry Pi. Le Raspberry Pi est un nano-ordinateur dont le disque dur est constitué d'une carte micro-SD. C'est sur cette carte que sont installés le système d'exploitation et les programmes d'acquisition et de traitement. Le travail sera réparti entre trois binômes formant un groupe, pour écrire un programme commun de détection de visages. Ce programme ne fonctionne que sur l'une des cartes micro-SD fourni au groupe.

1.1.2 Partie hardware

Un avantage majeur du RPi est son faible coût comparé à un ordinateur classique -32 euros-, et un bon compromis entre la taille et les performances : 512 MB RAM et 700MHz de vitesse d'horloge pour le modèle B ; 1GB de RAM, Quad Core, et 900MHz de vitesse d'horloge pour le modèle 2.

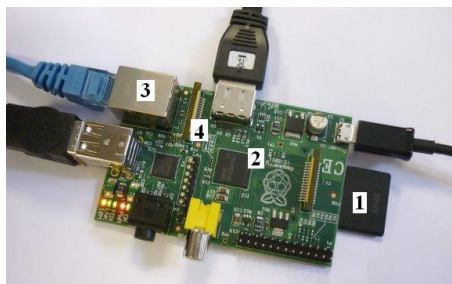


Figure 1.1 – Raspberry Pi (modèle B).

Question 1 :

Sur la figure 1.1, que représentent les éléments 1, 2, 3, 4 ?

1.2 Mise en route et test d'acquisition

C'est en dernier que l'on alimente le RPi !

Branchez le RPi à un écran via un adaptateur HDMI-VGA et un câble VGA. Branchez une souris, un clavier, et un câble Ethernet, une caméra Pi. Branchez le câble Ethernet au réseau. Enfin, branchez l'alimentation.

1.2.1 Démarrage du RPi

Le RPi démarre automatiquement à la mise sous tension. Si vous utilisez Raspbian Wheezy (un Pi A), le login est `pi`, le mot de passe est `raspberry`. Lorsque vous l'écrivez, le mot de passe n'est pas affiché, c'est normal. Si vous travaillez avec le système d'exploitation Raspbian Jessie, vous n'avez pas à fournir de login et mot de passe au démarrage.

Si vous travaillez avec Raspbian Wheezy, placez-vous en mode GUI (graphical user interface) avec la commande `sudo startx`.

1.2.2 Acquisition d'image

Pour réaliser une acquisition test, lancez un terminal, et écrivez :

```
raspistill -t 2000 -o imagetest.jpg -q 5
```

Vous avez alors une visualisation de deux secondes. Le paramètre `-q` désigne le taux de compression jpeg. Il est ici fixé à 5%.

Question 2 :

Faites quelques acquisitions de la mire en Fig. 2.1 en faisant varier la qualité de la compression. Renommez à chaque fois le fichier `imagetest.jpg`. Par exemple, `imagetest1.jpg` pour une compression à 1%. Comment varie l'aspect de l'image quand le taux de compression prend les valeurs $q=1, 2, 5, 10, 20, 50$, et 100 ?

1.2.3 Acquisition de vidéo

Pour réaliser une acquisition vidéo, sur le terminal :

```
raspivid -t 5000 -o video.h264 -b 3500000
```

Dans l'exemple ci-dessus, on enregistre un clip de 5 sec. au rythme spécifié (3.5Mbits/sec.). Pour la visualiser, vous pouvez la transférer sur le PC (avec le programme `Client_Web.py`) et utiliser par exemple VLC si vous êtes sous Windows.

2 Travail collaboratif et tests de détection

2.1 Travail collaboratif

2.1.1 Détection de visage

Formez un groupe de trois binômes. Chaque groupe doit avoir à disposition au moins un Raspberry Pi 3. C'est sur les cartes micro-SD des Raspberry Pi 3 que se trouve une boîte à outils de traitement d'image nommée 'OpenCV' qui permet de faire du traitement d'images.

Question 3 :

Implantez le programme 1 de détection de visage : chaque binôme implante sa partie du programme, désignée par

Partie 1, ##### Partie 2, et ##### Partie 3 dans l'algorithme 1 de détection de visage. Sauvegardez vos programmes avec le nom suivant :

Prog1, ##### Prog2, et ##### Prog3.

2.1.2 Transfert de programme

Le programme Client_Web permet de transférer un fichier entre un RPi source et un RPi cible.

Il peut déjà être présent sur votre RPi, sinon vous pouvez le télécharger sur la page suivante :

<http://www.fresnel.fr/perso/marot/#Teaching>

La cible doit être un RPi plutôt puissant (tel un RPi 3), muni d'une carte micro-SD sur laquelle OpenCV est installé.

Question 4 :

Quelle est l'adresse IP de votre cible ?

Algorithm 1 Détection de visage

```
##### Partie 1

import picamera
import cv2

camera = picamera.PiCamera()  ## Declare la camera
camera.resolution = (1024,768) ## Choix de la resolution (max par def.)
camera.brightness = 60
camera.rotation = 0  ## 0 90 180 ou 270
camera.crop = (0.0,0.0,1,1)  ##

camera.capture('image.jpg')  ## Prend une photo
camera.close()  ## Arrete la camera

##### Partie 2

## Lecture de l'image
image = cv2.imread('image.jpg')

## Charge le classifieur pour la detection de visage
faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

## Detection des visages
faces = faceCascade.detectMultiScale(image, 1.3, 5)

## Placement des rectangles sur les visages
for(x, y, w, h) in faces:
    cv2.rectangle(image, (x,y), (x+w, y+h),(0, 255, 0), 2)

## Enregistrement de l'image résultat en jpg
cv2.imwrite('image_output.jpg',image)

##### Partie 3

## Affichage de l'image avec les visages detectes
namewindow = "Faces found"
cv2.imshow(namewindow,image)  ## Affiche l'image
##cv2.resizeWindow(namewindow,800,600)  ## Permet de redimensionner la fenêtre
cv2.waitKey(0)  ## Attend appui sur une touche et...
cv2.destroyWindow(namewindow)  ## Ferme la fenêtre
```

- Sur Raspberry Pi source (client) :

Lancer `Client_Web`. Pour cela, sur la ligne de commande, tapez :
`python Client_Web.py`.

L'adresse de l'ordinateur source (inscrite dans le fichier `hosts`), ainsi que le port utilisé (choisi automatiquement par le programme `Client_Web.py`) sont fournis au lancement de l'application.

- Sur le RPi cible (serveur) :

Ouvrir une fenêtre web (bouton planète - flèche en haut à gauche), et tapez l'adresse IP qui vous a été donnée par le programme `Client_Web` sur la raspberry source. Par exemple :

`http://172.17.107.161:8000/`

Question 5 :

Lancez `Client_Web` sur chaque source. Combien y a-t-il de sources et quelle est leur adresse IP ?

Question 6 : Avec la fenêtre web, Visualisez-vous le contenu des dossiers dans le quel se trouve l'application `Client_Web`, et en particulier vos programme ?

Question 7 : Téléchargez les programmes que vous avez écrits. Sont-ils enregistrés sur le Raspberry cible ?

Une fois que vous avez téléchargé vos programmes depuis votre RPi, tapez `Ctrl C` sur la commande du RPi. Cela termine l'application `Client_Web`. Vous pouvez utiliser vos programmes avec les logiciels installés sur votre Raspberry.

2.2 Tests de détection de visages

Question 8 : Dans le dossier `home/pi` du Raspberry cible, dans lequel vous avez téléchargé vos programmes, se trouve le fichier `haarcascade_frontalface_default.xml`. Créez un dossier à votre nom. Dans ce dossier, copiez le fichier

haarcascade_frontalface_default.xml, et vos programmes. Votre dossier est-il créé? Contient-il les fichiers de programmes python et le fichier xml?

Question 9 :

Vous pouvez maintenant combiner vos fichiers en un seul programme. Pour cela, ouvrez les programmes avec clic droit > 'text editor'. Copiez et collez les parties 2 et 3 sur le programme qui contient la partie 1. Vous obtenez ainsi le programme complet. Enregistrez le programme sous le nom `Detection_Visages.py`, et fermez le programme. Etes-vous certains qu'il n'y a pas de redondance?

Question 10 : Exécutez votre programme complet. Pour cela :

Lancez un terminal via Menu > Accessories. Vous travaillez avec un RPi 3, sous Raspbian Jessie, donc vous devez vous placer dans l'environnement opencv en tapant les lignes de code suivantes :

```
source ~/.profile  
workon opencv
```

Sur certaines cartes il est possible que vous ayez plutôt à écrire :

```
source ~/.profile  
workon cv
```

Cela dépend de la façon dont OpenCV a été installé.
Exécutez votre programme complet de la façon suivante :

```
python Detection_Visages.py
```

Est-il nécessaire d'appliquer une rotation à l'image? Si oui, comment devez-vous modifier votre programme pour faire en sorte de retourner l'image? Y a-t-il enregistrement des images `image.jpg` et `image_output.jpg`?

Question 11 : Quelles sont les conditions dans lesquelles les visages sont bien, ou mal détectés?

2.3 Conclusion

Vous avez utilisé un système complet d'acquisition et de traitement de signaux échantillonnés. Quels étaient ces signaux en définitive ?

De ce TP, que déduisez-vous de la capacité d'un Raspberry Pi à bas coût à exécuter des programmes de traitement d'image avancés ? Pour répartir les tâches dans la conception de tels programmes, quels sont les éléments essentiels à maîtriser ?

Débranchez votre setup Raspberry en commençant par l'alimentation. Rangez les cartes micro-SD, et les adaptateurs. Refaites les branchements des ordinateurs aux périphériques que vous avez utilisés pour vos Raspberry Pi.

Annexe

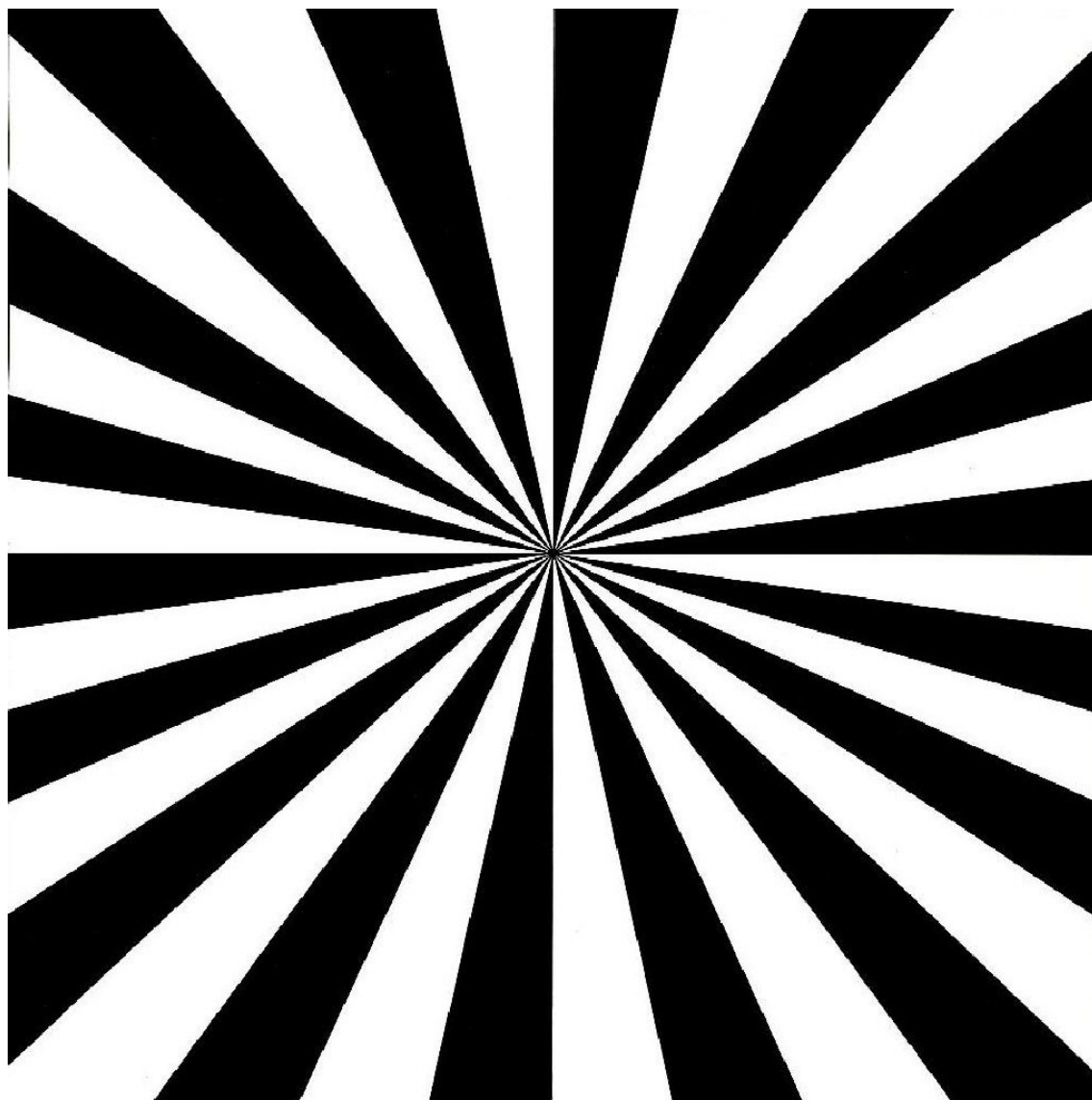


Figure 2.1 – Mire de test pour caméras