

Etude des systèmes échantillonnés

Travaux Pratiques

RPi 1: Introduction au Raspberry Pi

J. Marot

1 Introduction

1.1 Les ordinateurs mono-carte

Le Raspberry Pi[®] (ou RPi) est un ordinateur mono-carte. On justifie facilement l'achat d'un RPi par le fait qu'une tour classique, unité centrale de PC, est plus chère, et pas nécessaire à certaines applications comme du calcul ou de l'hébergement de site web. L'intérêt du RPi est aussi de permettre un interfaçage facile avec une caméra, et de traiter le flux vidéo fourni par cette caméra.

1.2 Objectif du TP

Ce TP est axé sur un système de vision, qui est à la base de tout système de contrôle non-destructif par des méthodes optiques par exemple. Il existe bien d'autres façon de mettre à profit un RPi, de l'électronique embarquée aux serveurs. On souhaite lors de ce TP, faire de l'acquisition d'image avec une caméra Pi, en examinant les différents paramètres d'acquisition, et étudier l'effet d'une méthode de traitement sur une photographie que vous aurez acquise.

2 Mise en route et connexion

Vous disposez d'un Raspberry Pi et d'un ordinateur PC classique.

2.1 Branchement du RPi

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. Branchez le câble Ethernet au réseau. Enfin, branchez l'alimentation.

2.2 Démarrage

Le RPi démarre automatiquement à la mise sous tension.

Le mode de démarrage dépend du système d'exploitation que vous utilisez. Il s'agit soit de Raspbian Wheezy, soit de Raspbian Jessie. Si c'est Jessie qui est installé sur votre carte, le RPi démarre directement en mode 'interface graphique' de type Windows par exemple. Vous n'avez alors pas de login et mot de passe à renseigner.

Si c'est Wheezy qui est installé sur votre carte, la ligne de commande vous est proposée. Vous devez rentrer un login, puis un mot de passe. Le login est `pi`, le mot de passe est `raspberry`. Lorsque vous l'écrivez, il n'est pas affiché, c'est normal.

Question 1: Comment définissez-vous une adresse IP ? En quoi est-elle utile pour communiquer avec un ordinateur ?

Question 2: L'adresse IP de votre RPi doit être fixée. Chaque RPi doit avoir une adresse différente. Demandez-là à l'enseignant et notez-la ! Pour vérifier que l'adresse IP de votre carte micro-SD est fonctionnelle, et pour la changer éventuellement, vous pouvez visualiser le contenu d'un fichier sur lequel l'adresse est inscrite. C'est ce qu'on appelle une adresse IP 'statique'.

Le nom du fichier dépend du système d'exploitation avec lequel vous travaillez.

Si vous travaillez sous Raspbian Jessie:

vous devez vérifier le contenu du fichier `dhcpcd.conf` situé dans le dossier `/etc/`. Pour cela sur

la ligne de commande tapez: `sudo nano /etc/dhcpd.conf`. Tout en bas du fichier vérifiez que l'adresse IP (écrite deux fois) correspond à une adresse valable pour le réseau local. Si c'est le cas vous n'avez rien à modifier.

Si vous travaillez sous Raspbian Wheezy:

vous devez vérifier le contenu du fichier `interfaces` situé dans le dossier `/etc/network`. Pour cela sur la ligne de commande tapez: `sudo nano /etc/network/interfaces`. Quelle est l'adresse indiquée en face de la ligne `address` ? Est-ce bien une adresse qui convient au réseau local ? Par exemple: `172.17.107.154` si l'une des adresses du réseau (sur l'un des ordinateurs) est `172.17.107.XXX` avec `XXX` entre 1 et 255. Si oui vous n'avez rien à modifier.

Vérifiez de même le fichier `/etc/hosts`: adaptez si nécessaire l'adresse IP mentionnée.

Question 3: Si vous avez dû modifier l'adresse IP, redémarrez le RPi avec la commande `sudo reboot`. Quelle est après redémarrage l'adresse IP de votre RPi ? Pour la connaître utilisez l'instruction `ifconfig` ou l'instruction `hostname -I`.

2.3 Connection en SSH

Téléchargez le logiciel Putty (lien www.fresnel.fr/perso/marot/Documents/putty.exe en .exe). Tapez l'adresse IP de votre RPi dans 'Host Name' puis appuyez sur Open. Vous pouvez maintenant prendre le contrôle de votre Raspberry Pi à distance, ou bien directement.

Les login et mot de passe sont toujours: `pi` et `raspberrypi`, quel que soit le RPi et le système d'exploitation utilisés.

Une fenêtre s'affiche sur l'écran de votre PC. C'est le mode 'lignes de commande' qui est proposé, et que nous utiliserons dans un premier temps.

Question 4: Les deux types de systèmes d'exploitation majeurs sont l'interface graphique et la ligne de commande. Pouvez-vous citer un système d'exploitation qui passe par interface graphique ? Un système d'exploitation qui passe par ligne de commande ?

3 Acquisition d'image

Avec quelques lignes de commande, nous allons configurer et tester la caméra 'Pi'. Elle est spécifiquement dédiée au Raspberry Pi.

3.1 Configuration de la caméra Pi

La caméra est implantée *via* sa nappe près du port Ethernet. Le côté bleu de la nappe doit se trouver du côté du port Ethernet. Afin d'acquérir des images et de les traiter, il faut configurer la caméra. Premièrement, il faut autoriser la camera dans les paramètres de la raspberry pi :

on saisit sur la ligne de commande: `sudo raspi-config` suivi de (5) enable camera, faites Entrée, puis allez à Finish, et faites Entrée. Lorsqu'on vous demande si vous voulez rebooter, choisissez Non (N)

Pour réaliser une acquisition test:

```
raspistill -v -o test.jpg
```

Question 5: L'image a-t-elle été bien enregistrée ? Vérifiez avec la commande `ls`.

3.2 Acquisition d'image

Nous souhaitons maintenant changer quelques paramètres d'acquisition, et enregistrer une image en format '.jpg' avec un programme.

Travail à réaliser: Implantez le programme Python ci-dessous d'acquisition d'image. Pour cela, créez un nouveau fichier: `sudo nano Acquisition.py`

Il contiendra les lignes de commande suivantes (concentrez-vous et faites bien attention à la syntaxe):

```
import picamera

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 et l'enregistre en jpg
camera.close()                 ## Arrete la camera
```

Fermez votre fichier avec Ctrl X, tapez sur Y pour l'enregistrer. Vous revenez alors automatiquement à la ligne de commande. Lancez votre programme en tapant: `python Acquisition.py`. Contrairement au cas où la commande `raspistill` est utilisée, la photo ne s'affiche pas. Mais le fichier `image.jpg` doit être enregistré. Vérifiez avec la commande `ls`

Faites diverses acquisitions:
faites varier, la luminosité, l'angle de rotation, en changeant le nom du fichier enregistré: créez les fichiers `image1.jpg`, `image2.jpg`, `image3.jpg`, etc.
Réalisez un sous-échantillonnage de l'image de résolution maximale d'un facteur 2. Pour cela, choisissez les paramètres adéquats de résolution.

Dans la suite du TP, vous allez constater que vous pouvez récupérer à distance les images que vous enregistrez.

4 Transfert d'image et traitement déporté

4.1 Transfert d'image

Il est d'abord nécessaire d'éditer un fichier:

Travail à réaliser: Sur le terminal de lignes de commandes, tapez:

```
sudo nano /etc/hosts
```

En bas du fichier, à gauche de `raspberrypi` mettez à jour *si nécessaire* l'adresse IP, avec celle de votre RPi, par exemple `172.17.107.154`.

Le programme `Client_Web` permet de transférer un fichier de la façon suivante:

- 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 l'ordinateur cible (serveur):

Ouvrir une fenêtre web firefox, et tapez l'adresse IP qui vous a été donnée par le programme `Client_Web` sur la raspberry source. Par exemple:

`172.17.107.154:8000/`

Question 6: Avec la fenêtre web, Visualisez-vous le contenu du dossier dans le quel se trouve l'application `Client_Web`, et en particulier votre image ?

Question 7: Téléchargez les images que vous avez acquises. Sont elles enregistrées sur votre ordinateur ?

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

4.2 Traitement déporté d'une image

4.2.1 Choix 1: Matlab

Votre image peut maintenant être traitée par un logiciel tel Matlab. Effectuez une binarisation en appliquant un seuil aux valeurs de niveaux de gris de l'image:

```
A=double(imread('image.jpg'));
A=A/max(A(:));
B=zeros(size(A));
seuil=1/2;
B(A<max(A(:))*seuil)=1;
figure(1)
imagesc(A)
colormap(gray)
figure(2);
imagesc(B)
colormap(gray)
```

Question 8: Dans le programme ci-dessus, la valeur de seuil pour la binarisation est de 1/2. Pouvez-vous proposer une valeur de seuil qui rende un résultat pertinent ?

4.2.2 Choix 2: OpenCV sur Raspberry

Votre image peut aussi être traitée avec la bibliothèque opencv, installée sur les Raspberry Pi 3.

Lancez un terminal via Menu > Accessories.

Créez le fichier `apply_Threshold.py`:

Pour cela tapez sur le terminal `sudo nano apply_Threshold.py` et complétez-le avec les lignes de code suivantes:

```

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

## Beaucoup d'operation de OpenCV se font en niveaux de gris,
grayscaled = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
print( 'Adaptive threshold ...')
retval2,thresholdedimageimage = cv2.threshold(grayscaled,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)
print('Valeur de seuil calculee')
print(retval2)
cv2.imwrite('image_threshold.jpg',thresholdedimageimage)
## Affichage de l'image avec les visages detectes
print('preparation de l apercu')

namewindow = 'original'
cv2.imshow(namewindow,image)
namewindow = 'Otsu threshold'
cv2.imshow(namewindow,thresholdedimageimage)
print('apercu genere')
print('appuyez sur une touche')
cv2.waitKey(0)
cv2.destroyAllWindows()
print('au revoir')
os._exit(0)

```

Vous travaillez sous Raspbian Jessie car c'est l'OS valable pour les RPi 3.
Placez-vous dans l'environnement opencv en tapant les lignes de code suivantes:

```

source ~/.profile
workon opencv

```

Exécutez votre programme complet de la façon suivante:
python apply_Threshold.py

Question 9: Dans le programme ci-dessus, la valeur de seuil pour la binarisation est calculée automatiquement et affichée. Quelle est sa valeur ?

5 Conclusion

En fin de TP, éteignez proprement le RPi: `sudo shutdown -h now`. Débranchez l'alimentation, et le câble ethernet. Quels sont les avantages majeurs du Raspberry Pi ?

Etude des systèmes échantillonnés

Travaux Pratiques

RPi 2: Travail collaboratif sur Raspberry Pi

J. Marot

1 Introduction

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.

1.2 Partie hardware

Un avantage majeur du RPi est son faible coût comparé à un ordinateur classique -32 €, 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.

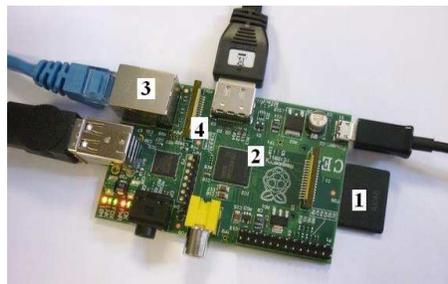


Figure 1: Raspberry Pi (modèle B).

Question 1:

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

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.

2.1 Démarrage du RPi

Le RPi démarre automatiquement à la mise sous tension. le login est `pi`, le mot de passe est `raspberrypi`. 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`.

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 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` ?

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.

3 Travail collaboratif

3.1 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.

3.2 Transfert de programme

Le programme `Client_Web` permet de transférer un fichier entre un RPi source et un RPi cible. La cible doit être un RPi plutôt puissant (tel un RPi 3), muni d'une carte micro-SD sur laquelle OpenCV est installé.

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
```

Question 4:

Quelle est l'adresse IP de votre cible ?

- 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.

3.3 Détection de visages sur RPi

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 nommé `L3SPIGr` où `Gr` désigne votre groupe. 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. Êtes-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
```

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` ?

4 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

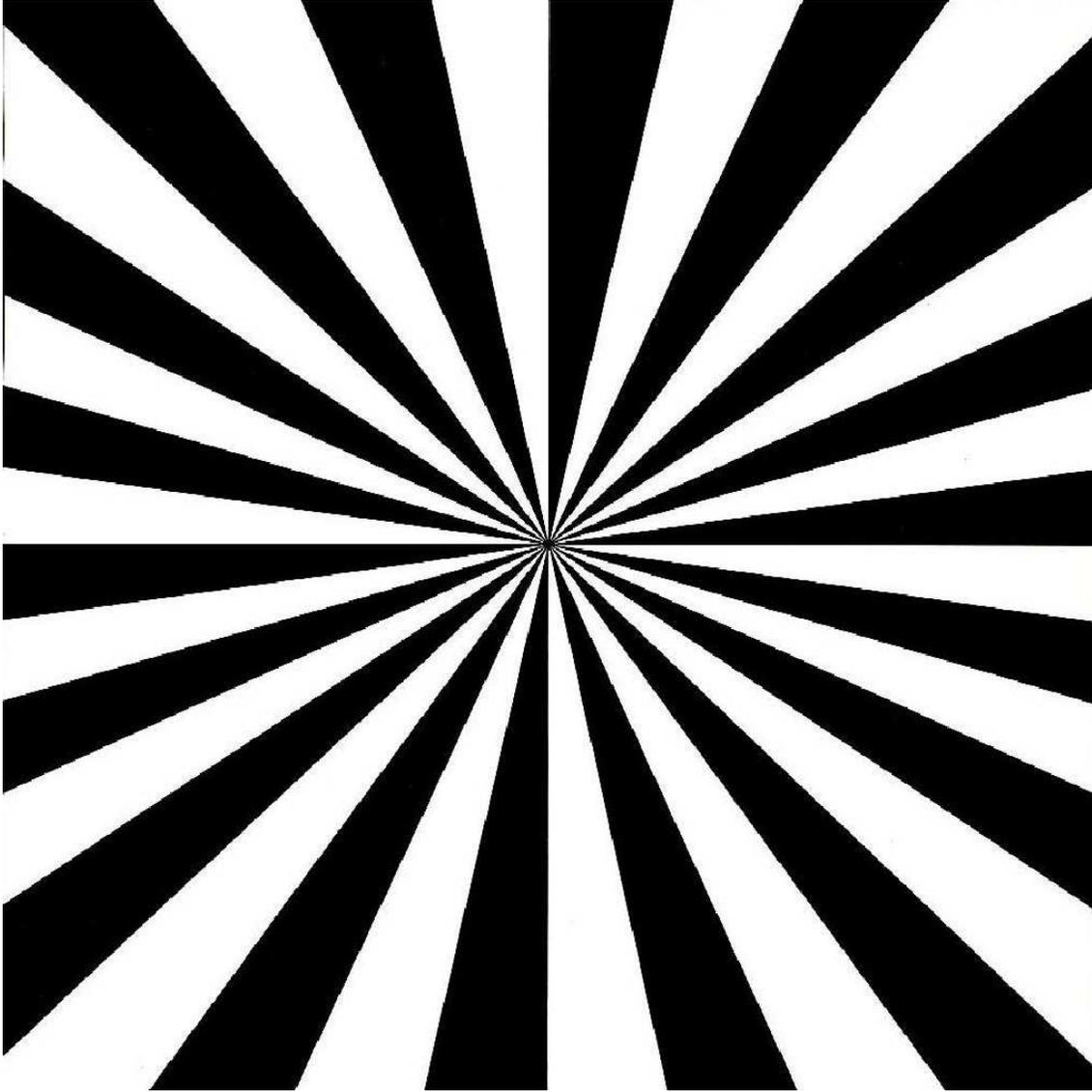


Figure 2: Mire de test pour caméras

Etude des systèmes échantillonnés

Travaux Pratiques

RPi 3: Création de séries numériques
sous Python sur Raspberry Pi

J. Marot

1 Introduction

Vous disposez d'un PC classique et d'un setup Raspberry Pi. L'objectif du TP est de générer des séries numériques avec le logiciel Python, d'une part sur Raspberry Pi, d'autre part sur PC. Le langage Python est du type 'texte structuré'. Il est similaire à Matlab car il inclut des boucles, des tests, et des fonctions. Il est aussi similaire au langage C car on importe des bibliothèques de fonctions (de même que l'on inclut des fichiers .h en c). Il ressemble aussi au C++ car il permet de faire de la programmation orientée objet. Dans le cadre de ce TP, nous souhaitons créer des séries numériques à l'aide de fonctions Python, et lancer ces programmes Python sur une Raspberry Pi. Nous effectuons donc des travaux similaires à ceux réalisés sous Matlab lors d'autres TPs, avec du matériel et un logiciel beaucoup moins chers puisque le Raspberry Pi vaut 35 euros et Python est un logiciel libre gratuit.

2 Setup hardware

Vous disposez pour chaque binôme d'un setup Raspberry Pi complet. 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. Branchez le câble Ethernet au réseau. Enfin, branchez l'alimentation.

Question 1: Sur la ligne de commande tapez `sudo startx`. Vous passez alors à un autre type de système d'exploitation. Lequel est-ce ?

Question 2: Lancez l'interface Python. Pour cela allez dans Menu>Programming. Combien de versions de Python sont installées ? Choisissez Python 2. Créez un dossier sur le bureau à votre nom. Enregistrez dans ce dossier votre nouveau fichier sous le nom `Fonctions.py` et complétez-le avec

```
print("bonjour")
```

Question 3: Exécutez votre programme: à partir du fichier faites Run>Run Module. Voyez-vous `bonjour` apparaître sur la ligne de commande ? Créez un nouveau fichier nommé `SeriesNumeriques.py`.

3 Exemples de fonctions et de programmes Python

Vous devez maintenant compléter vos deux fichiers: celui contenant les fonctions et celui permettant de tester ces fonctions en créant des séries numériques. Voici, dans les algorithmes 1 et 2 ci-dessous, des exemples de fonctions et un exemple d'utilisation de ces fonctions:

Algorithm 1 Fichier de définition *Fonctions.py*

```
def add(a,b):
    c=a+b
    return c

def mult(a,b):
    c=a*b
    return c

def testnombre(a,seuil):
    if a<seuil:
        print("petit")
    else:
        print("grand")

def boucle(indice):
    n=0;
    while n<=indice:
        print(n)
        n=n+1

    print("boucle finie")
```

Algorithm 2 Fichier de test *SeriesNumeriques.py*

```
import math
from Fonctions import add, mult, testnombre, boucle

a=2
b=3

testnombre(a,b)

boucle(b)

c=add(a,b)
print(c)

d=mult(a,b)
print(d)
```

Question 4: Complétez vos fichiers avec les exemples qui vous sont donnés. Respectez bien la syntaxe (les ' : ' et les indent notamment). Parvenez-vous à exécuter le programme `SeriesNumeriques.py` ? Qu'affiche-t-il en résultat d'exécution ?

4 Création de fonctions et de programmes Python

Vous allez maintenant créer vos propres fonctions.

Question 5: Complétez le fichier `Fonctions.py` avec les fonctions `power(a,n)` qui calcule a^n et `factorielle(n)` qui calcule $n!$. Notez pour cela que $n!=n*(n-1)!$.

Testez ces fonctions en complétant le programme `SeriesNumeriques.py`: affichez le résultat à chaque fois avec la fonction `print`.

Question 6: Implantez dans le programme `SeriesNumeriques.py` la série numérique suivante:

$s(n) = \left(\frac{b}{1+a}\right)(1 - (-a)^n)u(n)$ où $u(n)$ est l'échelon unité.

5 Transfert de fichier et test sur PC

Nous souhaitons maintenant tester les programmes sur un PC classique. Pour cela il faut y transférer les programmes. Allumez un PC, bootez sous Ubuntu.

Question 7: Sur votre Raspberry Pi: placez vos programmes Python `Fonctions.py` et `SeriesNumeriques.py` dans le dossier `/home/pi`. Retrouvez-vous le fichier `Client_Web.py` dans ce même dossier ?

Question 8: Vous pouvez, grâce à l'application `Client_Web.py`, autoriser d'autres ordinateurs à accéder à votre Raspberry Pi. Lancez l'application `Client_Web.py`, à l'aide de la commande suivante sur un terminal:

`Python Client_Web.py.`

Ouvrez une fenêtre Firefox sur le PC (au préalable, vérifiez que son câble Ethernet est bien branché au réseau). Ecrivez `172.17.107.15x:8000` sur la barre d'adresse Firefox, où `172.17.107.15x` est l'adresse IP de votre Raspberry Pi. Téléchargez les programmes que vous avez écrits sur votre Raspberry Pi (clic droit > Enregistrer sous). Les deux programmes `Fonctions.py` et `SeriesNumeriques.py` sont-ils présents sur le Bureau ?

Question 9: Sur le PC, sur un terminal, faites `cd Bureau`. De façon similaire à ce qui est montré en figure 1 pour un programme simple `'hello.py'`, exécutez le fichier `SeriesNumeriques.py`: `python SeriesNumeriques.py`. Les résultats obtenus sont-ils les mêmes que sur la Raspberry ?

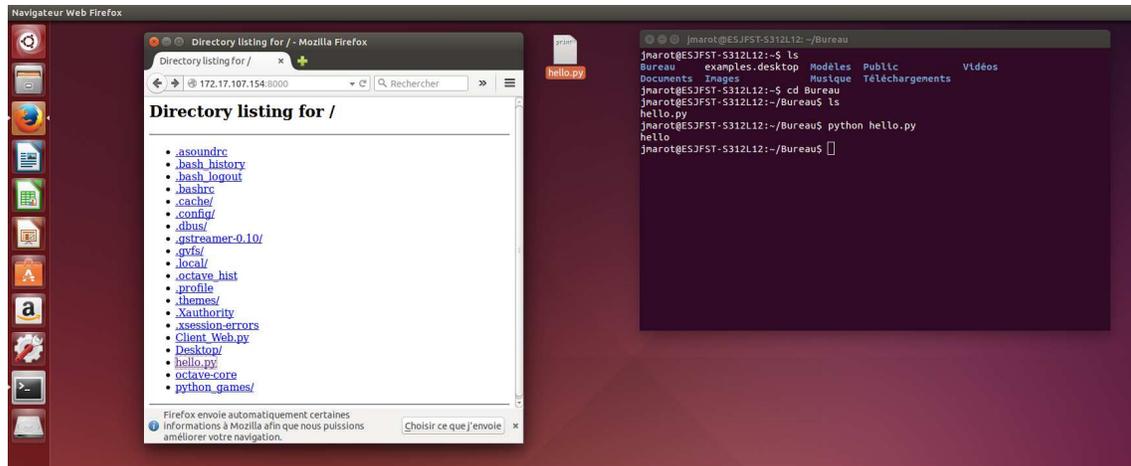


Figure 1: Bureau du PC démarré sous Ubuntu

6 Conclusion

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