

# Raspberry Pi for Image Processing Education

Julien Marot  
AMU, CNRS,  
Institut Fresnel, France  
marot@fresnel.fr

Salah Bourennane  
CNRS, Centrale Marseille,  
Institut Fresnel, France  
bourennane@fresnel.fr

**Abstract**—This paper firstly describes the development and evaluation of a project course which yields university students building a complete hardware and software chain for a digital image processing application. To arouse the interest and learning initiative of students, we propose them to build a setup including a Raspberry Pi® and image processing programmes. This inexpensive single board computer answers today's issues in energy saving and permits to review fundamental hardware and software principles. Secondly, we propose a low-cost setup for a time-limited practical work: a Raspberry Pi® is shared and controlled remotely by several student pairs; we emphasize the interest of collaborative work, and we provide knowledge and skills about micro-computers to a large number of students simultaneously.

## I. INTRODUCTION

Students who reach nowadays a digital signal processing (DSP) or an micro-computer course are known as 'digital natives'. Referring to a Program for International Student Assessment (PISA) report published in 2012 [1], 99% of 15-year old pupils in an OECD country afford a computer and an internet connection at home. They are aware of the diversity of hardware supports smartphones, tablets, PC computers [2]. It is then difficult to maintain experimental setups attractive for them. For this, they have to be rather inexpensive, but also to offer the possibility for the students to grab easily, with some visual information, the results provided by the hardware and software setup they have worked on. This paper is related to prior work in the following way: In [3], the authors aim at improving computer science education through algorithm visualization. However, information and communication technologies are not reduced to algorithms, and it is important that students understand the link between hardware and software. The technology acceptance model (TAM) is considered a successful model in studying learners' behaviour regarding technology use in education [4]. One conclusion of TAM is that people are eager for novel technologies if they grasp their usefulness [5]. Hence

the need for an application when teaching informatics. The main contribution of this paper is as follows: To teach micro-computers and their interest for digital signal processing applications, we propose an approach that uses a low-cost single board computer. We also demonstrate its interest for a collaborative practical work. We propose a setup including a Raspberry Pi® (RPI) to perform somehow advanced digital signal processing algorithms [6], [7]. As advised in [8], [9], or [10], real-time DSP applications performed for instance on Raspberry Pi and Arduino boards are efficient to let the students understand the interest of signal processing. Moreover, some advanced algorithms can be implemented on a Raspberry Pi, for various applications in computer science and computer vision [11], [12], [13], [14].

We want to answer the following questions: what is the best way to learn the significance of fundamental terms in computers, the best setup for the students to grab the interest of image processing ? How can we enhance the value of collaborative work during a time-limited practical work session ? In section II we present our setup and student tasks in the frame of a project course. In section III we present a student evaluation of our project course and of its theoretical 'blackboard' version. In section IV, we show the interest of the RPI to create a classroom setup involving a collaborative work.

## II. MATERIALS AND EXPERIMENT PROCEDURE

### A. Hardware

A major advantage of the RPI is its low cost compared to a classical PC computer -ten times less than the DSP card cited in section I-, and good tradeoff between size and performance: The B model is provided with 512 MB of Random Access Memory (RAM), and a clock speed of 700MHz. The model 2 is provided with 1GB of RAM, and contains a Quad Core processor, each with a clock speed of 900MHz.

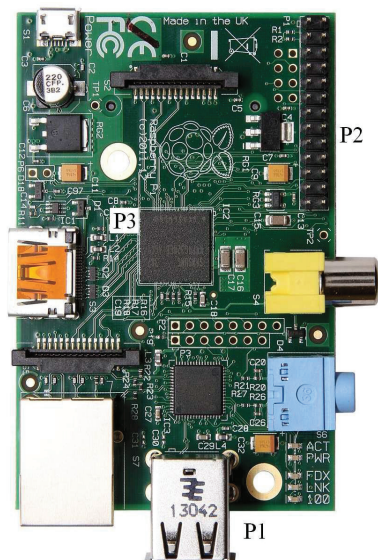


Figure 1. RPi Model B.

Figure 1 shows the RPi model B, where P1 locates the two USB ports, P2 denotes a GPIO port, and P3 the Samsung<sup>®</sup> memory chip, with the "ARM" processor underneath. A USB hub is necessary if more than two USB peripherals are needed. Here is an example of hardware setup which is sufficient to interact with the RPi: protection case, 16 Gb Micro-SD card -and SD adapter for RPi A and B-, Micro-USB power supply, USB Hub with power supply, mouse, keyboard, USB camera or 'Pi Camera' plugged onto a specific port of the RPi, HDMI Cable, HDMI-DVI adapter, HDMI screen. We notice that, although the cost of the RPi itself is relatively low, the whole budget is up to 180 US \$. Owing to the cost of the peripherals, a faster single board computer such as the Banana Pi Pro<sup>®</sup> (provided with a quad-core 1GHz processor, and 2GB of RAM) could be used despite a slightly higher cost compared to the RPi. Still for cost purposes an advantage of RPi is its ability to run freeware softwares. The students are asked to write a short report on the structure of the RPi, with compulsory keywords such as 'RAM', 'processor'... As all hardware elements are visible on the RPi, the educational target of this task is that the students get familiar with these elements.

### B. Operating system and softwares

This subsection is divided into three tasks: the students set the operating system, install programmes, and finally create their own image processing programmes.

1) *Operating system:* The first task for the students consists in installing the operating system on the micro-SD card by a PC computer through a software such as Win32Disk manager or Linux Live USB creator. They need an 'image file' to install the 'Raspbian' operating system. Raspbian is the RPi foundation's official operating system [15], and is based on Debian. The educational target of this task is the following: letting the students perform the installation starting from scratch is relevant so that they understand that the operating system is placed on the micro-SD card which is plugged into the RPi, and neither on the Samsung<sup>®</sup> memory chip, nor on the "ARM" processor placed underneath. They are asked to run the following command lines, for the purpose of software update or installation:

```
$ sudo apt-get update,
$ sudo apt-get upgrade,
$ sudo apt-get install.
```

This can be done by the students in a project setup, where the time constraint is not strong, but should be performed by the teacher in the framework of time-limited practical works. Last but not least, we emphasize the difference between the two modes for users to interact with their RPi: graphical user interface (GUI) and command line interface (CLI). This allows them to understand the difference between the "Windows"<sup>®</sup> - type GUI and the "Linux"<sup>®</sup> or "MS-DOS" - type CLI. On a RPi, you start the GUI from the CLI with the following command line:

```
sudo startx.
```

The CLI takes far less system resources but the GUI is more convenient to create program files, and toggle through program files and folders.

2) *Programmes:* We aim at showing to the students that they develop programmes based on free softwares and free libraries of programming functions. Although the mathematics concerning signal processing may be easier for 1-D than 2-D, we decided to consider images as processed data:

- students find it easier to grasp the consequences of some processings while looking at result images than while hearing sounds;
- students appreciate to process data that they have acquired themselves, such as photographs acquired with a camera connected to the RPi at USB or a specific port;
- a large panel of image processing and computer vision functions are available for free, as OpenCV [6] for C programming or SimpleCV [7] for Python

language.

To implement image processing algorithms, the students use Octave and Python programming languages. The students are assumed to have basic knowledge of programming languages: instruction list such as assembly, structured text such as C or Octave<sup>®</sup>. For this purpose, their first task consists in having a look at basic Octave and Python programmes. The educational target of this task is that they understand the similarities and differences between Octave, Python and other structured text languages. The second task for them is to study some image processing programmes in Python, aiming for instance at extracting the visually salient points in an image. The third task consists in writing their own image processing programme with Python language: keypoint extraction and matching. The steps of the programme are as follows: importation of required SimpleCV modules; definition of the size of the image provided by the camera; definition of the display window; definition of the variable which will contain the matrix image provided by the camera; performing binarization, dilation, and erosion.

### C. Application

In order to show the students the practical interest of image processing programmes, we provide them with various classical papers of the special issue [16] concerning hardware implementation of image vision systems, and invariant feature transforms [17], [18]. The students process images acquired with the Webcam. The hardware part includes a RPi B with a processor @700 MHz, the programmes are written under Python. The goal of the students is to detect salient points, and match these points between two images (see an example on Fig. 2). The salient points are Scale-Invariant Keypoints. The students are asked to compare the computational load of SIFT [17] (scale invariant feature transform), and SURF [18] (speeded up robust features).

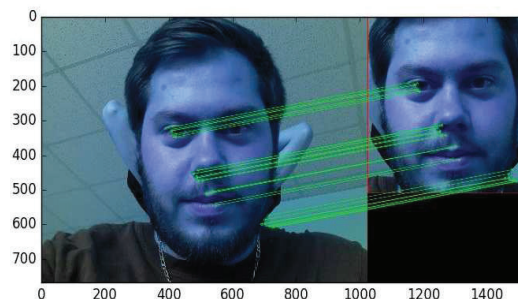


Figure 2. Image processing results: detection of salient points and matching.

## III. RESULTS

In this section, we propose a comparative evaluation of two ways of teaching micro-computers.

This evaluation was performed with  $N_s=46$  students for the 'black-board' course, and with 6 students for the project. We show how the traditional 'black-board' course, and the project were evaluated by the students. Their satisfaction was rated with four possible answers from  $A=1$  (rather bad) to  $A=4$  (rather good). Firstly, the questionnaire proposed to the 46 students following the theoretical course was as follows:

- 
- 1) Were the objectives of the course clearly presented ?
  - 2) Did the progression during the course facilitate the assimilation of the presented notions ?
  - 3) Did the complex notions presented during the course facilitate your comprehension ?
  - 4) Was the time available during the practicals sufficient to reach the objectives which were planned ?
  - 5) Did the practicals permit to understand the notions presented during the lessons ?
  - 6) Were your previous knowledge sufficient to tackle with this course ?
  - 7) Do you think this course is important for your education ?
- 

Let  $A_{ij}$  denote the answer to question  $i$  provided by student  $j$ . The mean value  $M_i$  for question  $i$  computed over  $N_s$  students is defined as:

$$M_i = \frac{1}{N_s} \sum_{j=1}^{N_s} A_{ij} \quad (1)$$

$$STD_i = \sqrt{\frac{1}{N_s} \sum_{j=1}^{N_s} (A_{ij} - M_i)^2} \quad (2)$$

The mean value  $M_i$ ,  $i = 1, \dots, 12$  and standard deviation  $STD_i$ ,  $i = 1, \dots, 12$  of the answers provided by the students are presented in Table I.

Question index	$M$	$STD$
1	2.71	1.03
2	2.64	0.97
3	2.61	0.87
4	3.27	0.87
5	2.81	0.97
6	2.25	1.07
7	2.77	1.13

Table I  
EVALUATION OF THE THEORETICAL COURSE: MEAN AND STANDARD DEVIATION VALUE OF THE ANSWERS TO QUESTIONS 1 TO 7.

Secondly, the questionnaire proposed to the 6 students performing the project was as follows:

- 1) "Did your experiments, performed during the project, facilitate your understanding of micro-computer and digital signal processing applications?"
- 2) "Did you often connect issues of the project to your prior experiences and knowledge?"
- 3) "Was the fact that you had a common goal despite the division of the project into sub-tasks motivating?"

To question 1, 5 students answered '4' and 1 answered '2'. The mean value was then 3.7. To question 2, 4 of them answered '4' and 2 answered '2'. The mean value was then 3.3. To question 3, 5 of them answered '4' and 1 answered '3'. The mean value was then 3.8.

From Table I, we notice in particular that, to question 3, the mean value of the answers was 2.61, and standard deviation 0.87; to question 6, the mean value of the answers was 2.25, and standard deviation 1.07. This shows, in our opinion, that technical aspects of micro-computers remain recondite when presented in a theoretical manner: students are aware of the topic, but notions such as the different categories of logic devices, and the way they are programmed are difficult to teach 'on the blackboard'. For the two questions to the project students, the mean value of the answer is more elevated than the mean value obtained in the frame of a 'blackboard' course. Although they have been obtained from a reduced sample of students, they permit to draw some conclusions: The students appreciated the modularity of

the RPi, and the fact that it is appropriate to understand the architecture of a computer; the students are eager to perform projects with a technical realization involving a team work. This is the first reason why we extended this project into a classroom setup.

The second reason is that we wish to raise awareness of the interest of RPi's for digital signal processing among the largest possible number of students; and the third reason is that we think the RPi is appropriate for a collaborative work: we show in the next section how one RPi can be shared by several pairs of students and controlled remotely.

#### IV. COLLABORATIVE CLASSROOM SETUP

In this section, we propose an extension of our setup to a collaborative practical work dedicated to a classroom of 24 students. The RPi is particularly appropriate for such a framework because it affords an Ethernet port and can be controlled remotely through an SSH (secure shell) process. We focus on the example of a classroom with 24 students, using 4 RPi's. We set 4 teams of 6 students. Each team is divided into 3 pairs of 2 students. Each team works independently from the other teams, and affords one RPi with its own IP address. For each team, the hardware setup includes a RPi, an Ethernet cable, and a power supply. Each pair affords a computer connected *via* Ethernet to a local network. The objective for the students is to control their RPi remotely, through an SSH process, and to create a functional image processing programme through a collaborative work.

We split the practical work session into three phases: 10 to 20% for a large group lecture/face-to-face, 35 to 45% for a pair working, and 35 to 45% for the combination of the codes produced by each pair. During the second phase, each pair logs to the provided RPi with the 'Putty' software, and checks that two other pairs of their team are also connected with the `$ who` command line. A common folder is created and each pair is supposed to edit files inside this folder. Then, each pair creates a python program (first typing `sudo nano Prog.py` on the CLI). During a 4-hour session for instance, several exercises can be performed. In the following we detail a face detection application: pair 1 encodes image acquisition, pair 2 encodes face detection and pair 3 encodes image display. The code must be preceded by the following code lines to import of the required packages:

```
from SimpleCV import Camera, Display
from SimpleCV import Image
```

During the third phase, the 3 pairs of each team combine together their codes, firstly checking that the acquisition and the display of the image are running correctly, secondly inserting the face detection step.

We notice that this extension to a classroom setup exhibits two advantages: firstly, we provide knowledge and skills about RPi's to a large number of students simultaneously with a low-cost setup, secondly we encourage a collaborative work where one hardware setup is shared by several pairs of students aiming for a common goal but performing different tasks. This favors trust and emulation among students.

## V. CONCLUSION

This paper describes a new manner to teach new trends on micro-computers and digital signal processing. It is centered on the Raspberry Pi®. As it is a low-cost micro-computer, designed entirely using open source software components, it is a valuable candidate to teach the students a whole chain of real-time image processing system. Firstly, we propose a project approach as an alternative to the theory-oriented introductory courses. It aims at improving the students' understanding of the computers: the difference between hardware and software, between graphical user interface and command line interface, between operating system, memory chip and processor. The interest of the students for our project approach has been confirmed by an evaluation. Secondly, we propose a time-limited practical work approach: we took advantage of the possibility to control an RPi remotely, to create a low-cost classroom setup where several student pairs share one RPi, to emphasize the interest of collaborative work.

## ACKNOWLEDGEMENTS

The authors are very grateful to PhD student Benoit Martin, who largely contributed to this work through student supervision and pieces of advice about salient point detection.

## REFERENCES

- [1] [www.oecd.org/pisa/keyfindings/PISA-2012-students-computers-france-eng.pdf](http://www.oecd.org/pisa/keyfindings/PISA-2012-students-computers-france-eng.pdf)
- [2] OECD (2015), "Students, Computers and Learning: Making the Connection", *PISA, OECD Publishing*, Paris.
- [3] A. T. Avancena, A. Nishihara, and C. Kondo, "Developing an Algorithm Learning Tool for High School Introductory Computer Science", *Education Research International*, Vol. 2015, Article ID 840217, 11 pages.
- [4] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology", *MIS Quarterly*, vol. 13, no. 3, pp. 319-339, 1989.
- [5] M. M. Alhabahba, O. H. A. Mahfoodh, A. Pandian, Y. M. Mohammad, E. W. Ahmed, A. Albdour, and H. Al Bazar, "Check This Word Out! Exploring the Factors That Affect Students' Vocabulary Learning Using Smartphones via Partial Least Squares", *Education Research International*, Vol. 2014, Article ID 672031, 10 pages.
- [6] [http://docs.opencv.org/doc/user\\_guide/user\\_guide.html](http://docs.opencv.org/doc/user_guide/user_guide.html)
- [7] <http://simplecv.org>
- [8] D. Sharma, A. Poddar, S. Manna, and P.A. Naylor, "The SAS project: speech signal processing in high school education", *Proc. of Signal Processing Conference (EUSIPCO)*, pp. 1781-1785, Aug. 2015.
- [9] J. Sobota, R. Pisl, P. Balda, M. Schlegel, "Raspberry Pi and Arduino boards in control education", *IFAC Proceedings Volumes*, Vol. 46, no 17, 2013, pp. 7-12.
- [10] P. Jamieson and J. Herdtner, "More missing the Boat - Arduino, Raspberry Pi, and small prototyping boards and engineering education needs them", *IEEE Frontiers in Education Conference*, El Paso, TX, 2015, pp. 1-6.
- [11] Md. Raihan, M. S. Rahaman, M. K. Sarkar, and S. Mahfuz, "Raspberry Pi Image Processing based Economical Automated Toll System", *The Global Journal of Research in Engineering*, Vol. 13, No. 13-F, 2013.
- [12] G. Senthilkumar, K. Gopalakrishnan, and V. Sathish Kumar, "Embedded image capturing system using Raspberry Pi System", *International Journal of Emerging Trends and Technology in Computer Science*, Vol. 3, No. 2, pp. 213-215, 2014.
- [13] R. Neves and A. C. Matos, "Raspberry PI based stereo vision for small size ASVs," *2013 OCEANS - San Diego*, 2013, pp. 1-6.
- [14] Md. Maminul Islam, Md. Sharif Uddin Azad, Md. Asfaqul Alam, N. Hassan, "Raspberry Pi and image processing based Electronic Voting Machine (EVM)", *International Journal of Scientific and Engineering Research*, Vol. 5, No. 1, pp. 1506-1510, 2014.
- [15] <http://www.raspberrypi.org>
- [16] G. Botella, C. Garcia, and U. Meyer-Base, "Hardware implementation of machine vision systems: image and video processing", *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no 1, p 152, 2013.
- [17] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, vol. 60, no 2, pp. 91-110, Nov. 2004.
- [18] H. Bay *et. al.*, "Speeded-Up Robust Features (SURF)", *Computer Vision and Image Understanding*, vol. 110, no 3, pp. 346 - 359, Nov. 2008.