Implementation of Image Processing on Raspberry Pi

Article · May 2015

DOI: 10.17148/IJARCCE.2015.4545

3 authors, including:

Lokesha Hlmg
National Aerospace Laboratories (CSIR)
16 PUBLICATIONS **45** CITATIONS

SEE PROFILE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

Implementation of Image Processing on Raspberry Pi

K.S.Shilpashree[1], Lokesha.H[2], Hadimani Shivkumar[3]

M.Tech, VLSI Design and embedded system, E & C Department, Kalpataru Institute of Technology, Tiptur, India[1]

Senior Scientist, ALD Division, DSP Lab, CSIR-National Aerospace Laboratories, Bangalore, India[2]

Associate Professor, E & C Department, Kalpataru Institute of Technology, Karnataka, India[3]

Abstract: Today image processing are used in various techniques**,** this paper presents the implementation of image processing operations on Raspberry Pi. The Raspberry Pi is a basic embedded system and being a low cost a single-board computer used to reduce the complexity of systems in real time applications. This platform is mainly based on python. Raspberry pi consist of Camera slot Interface (CSI) to interface the raspberry pi camera. Here, the Dark and Low contrast images captured by using the Raspberry Pi camera module are enhanced in order to identify the particular region of image. This concept is used in the real time application of MAV, The MAVs are basically used to capture images and videos through the Raspberry pi camera module. Because of its credit card sized (small) and less weight in the design. However, the image captured by MAVs will consist of unwanted things due to atmospheric conditions; hence it is necessary to remove noise present in the MAVs images.

Keywords: Image Capturing, Raspberry Pi, Camera Module, Python, Enhancement Algorithms.

I. INTRODUCTION:

The image processing is a form of signal processing where the input is an image, like a photograph or video frame, the output of an image processing may be either an image or a video frame or a set of characteristics or parameters related to the image. The acquisition of digital image usually suffers from undesirable camera shakes and due to unstable random camera motions. Hence image enhancement algorithms are required to remove these unwanted camera shakes. This image processing concepts are implemented in Raspberry pi in the application of MAV. The Raspberry Pi is a basic embedded system having a credit card-sized single board computers developed in the UK by the Raspberry Pi Foundation. The Raspberry Pi is based on the Broadcom BCM2835 system on a chip (SOC) which includes an ARM1176JZF-S Core (ARM V6K)700 MHz CPU processor, Broadcom Video Core IV GPU having 17 pins, 3.5W of power, and 512 MB of RAM memory. The Raspberry Pi system has Secure SD card reader (models A and B) or Micro SD card reader (models A+ and B+) sockets for boot media and persistent storage. The system provides Debian Linux operating system Raspbian image for download. Python is used as main programming language for raspberry pi. A micro air vehicle (MAV) is a remote-controlled, Unmanned Aircraft Vehicle (UAV) significantly smaller than typical UAVs that have a size restriction. UAV is an aircraft without a human pilot. Its flight is controlled either autonomously on board computers or by the remote control of a pilot on the ground or in another vehicle. By having a Raspberry Pi camera module available on a MAV the efficiency of this air vehicle increases and new fields of applications are available. It is needed in military Operations, in which targets have to be identified. Such identification is often done by a human on ground, to reduce the probability of mistakes. But a Raspberry Pi camera module is also helpful if a MAV shall autonomously fly through an arch.

II. BASIC CONCEPT OF IMAGE PROCESSING

In general, any digital image processing algorithm consists of three stages: input, processor and output. In the input stage image is captured by a camera. It sent to a particular system to focus on a pixel of image that's gives, its output as a processed image.

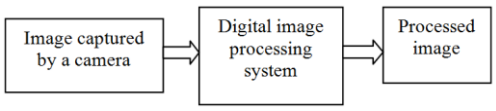

Fig. 1: General Block diagram of image processing

III. SYSTEM HARDWARE DESIGN

The Raspberry Pi board is the central module of the whole embedded image capturing and processing system as given in fig. 2. Its main parts include: main processing chip unit, memory, power supply HDMI Out i.e VGA display, Ethernet port, and USB ports.

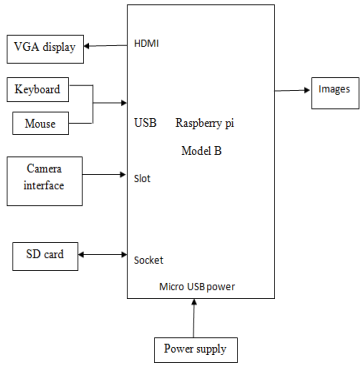

Fig. 2: System Block Diagram

DOI 10.17148/IJARCCE.2015.4545

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

A. RASPBERRY PI BOARD

The main signal processing chip unit used in Raspberry Pi system is a Broadcom 2835 700MHz Chip in which CPU core is a 32 bit ARM1176JZF-S RISC processor designed by Advanced RISC Machines. This main processing chip connects a camera and display. The Raspberry Pi design does not include a built in hard disk or solid state drive, instead used an SD card for booting and long term storage. This board is intended to run Linux Debian based operating systems. This Raspberry Pi module has a Samsung class 4 micro SD card preloaded with the Raspberry Pi NOOBS (New Out of Box Software) package, and a printed Micro SD card adaptor.

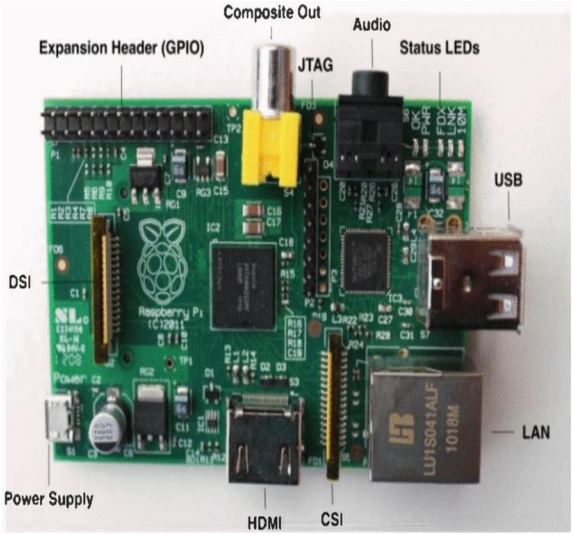

Fig. 3: Raspberry Pi board (Model B).

B. Camera Interface

The camera module used in this paper is raspberry pi camera module as shown in the Fig. 3. The camera module plugs to the CSI connector on the Raspberry Pi. It's able to deliver clear 5MP resolution image, or 1080p HD video recording at 30fps. The camera module attaches to Raspberry Pi by a 15 pin Ribbon Cable, to the dedicated 15 pin MIPI Camera Serial Interface (CSI), which was designed especially for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the BCM2835 processor.

Fig. 4: Raspberry Pi camera board

IV. METHODOLOGY

The proposed method uses the raspberry pi board is the main controller. The latest version of raspbian wheezy is used on to the board. After installing the OS to the board connect all the necessary hardware components and switch on the power supply.

It starts booting up the Board and login the raspberry pi by username and password. It operates on the Linux Debian arch operating system. It mainly works on the python software and checks the network settings to update the python software by commands in the terminal window. Following packages are to be installed for implementing the proposed model. Installation commands have been listed below.

1) sudo apt-get install python-matplotlib
2) sudo apt-get install python-numpy
3) sudo apt-get install python-scipy
4) sudo apt-get install python-imaging

Enable the camera settings on the board to capture the image and save it on the folder. Run the python code to check the enhancement algorithms and remove the noise present in an image. The proposed method implementation as shown in the flow chart in fig 5.

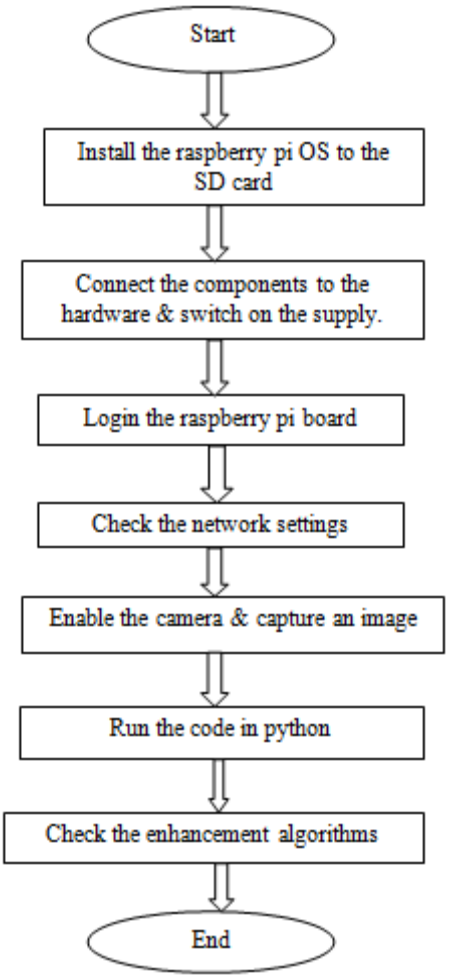

Fig 5: Flow chart of methodology.

V. RESULTS AND DISCUSSION

For the purpose of real time simulation the raspberry pi running the latest version of Raspbian wheezy was used. The development environment was python 2.7.3. Once the

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

user captures the objective image and specifies the reference image, the rest of the process is completely automatic and there is no need for user intervention. Here the algorithm has been applied to the complete image.

Fig. 6: Original Image

Fig 7. Gray Image

Fig 8: Brightness Enhanced Image

Fig 9: Contrast Stretched Enhanced Image.

In the application of micro air vehicle (MAVs) there is a noise present in the images due to the atmospheric conditions, so removing noise from images is important in this application and improving the quality of images. For this method I used the Rudin-Osher-Fatemi de-noising model (ROF). The total variation of a grayscale image I is defined as a sum of gradient norm for a continuous representation is given by

$$J(I) = \int |\nabla I| dx \ .$$

For a discrete setting, the total variation becomes

$$J(I) = \sum_{x} |\nabla I| \ ,$$

In the ROF algorithm it is to find the denoised image U that minimizes

$$\min_{U} \ ||I - U||^2 + 2\lambda J(U),$$

Where the norm ||I-U|| measures the difference between U and gray image I.

The ROF model has the interesting property that it finds a smoother version of the image while preserving edges and structures. The result as shown in below figures.

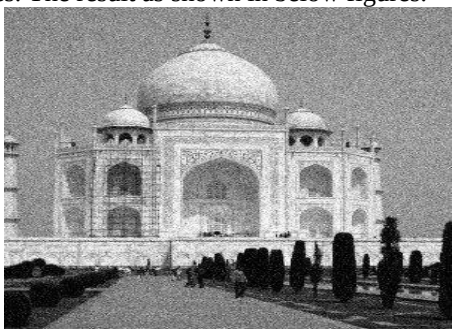

Fig 10: Noisy image

Fig 11: Noise removal image

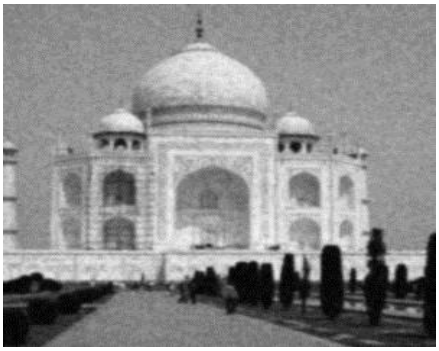

Fig 12: Gaussian filtered image

DOI 10.17148/IJARCCE.2015.4545

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 5, May 2015

VI. CONCLUSION

We implemented the algorithm to enhance an image in different enhancement degree using the raspberry pi. It was found that the algorithm developed for the raspberry pi executes successfully and gives a very colorful image.

REFERENCES

[1]. G.Senthilkumar1, K.Gopalakrishnan2, V. Sathish Kumar3 Embedded Image Capturing System Using Raspberry Pi System, Volume 3, Issue 2March–April 2014 Page 213.
[2]. Sahani, M., Rout, S.K., Sharan, A.K., Dutta, S. real time color image enhancement with a high regard for restoration of skin color by using raspberry pi Communications and Signal Processing (ICCSP), 2014 International Conference IEEE.
[3]. Y.Saahithi, E.Sai Spandana Reddy, P.Samskruthi Reddy, Advanced Embedded Security System With Image Capturing In SD Card, Volume No: 1(2014), Issue No: 12 (December).
[4]. Ajinkya Patil1, Mrudang Shukla2 Implementation Of Classroom Attendance System Based On Face Recognition In Class, IJAET, Vol. 7, Issue 3 July, 2014.
[5]. Umesh P Image Processing in Python, CSI Communications, December 2012.
[6]. Jan Erik Solem, Programming Computer Vision with Python, 2012.
[7]. Raspberry pi – www.raspberrypi.org