Real Time Application of Raspberry Pi in Compression of Images

Sahitya S, Lokesha H, Sudha L K

Abstract— **An image contains large amount of digital data and it is necessary to reduce digital data volume for transmission and preservation by using image compression. This paper mainly concentrates on image compression using Raspberry Pi processor which helps to preserve large number of images and in retaining its quality. Raspberry Pi processor allows the implementation of most widely used 2-D Discrete Cosine Transform (DCT) compression method to give the Joint Photographic Experts Group (JPEG) format using Open CV platform.JPEG compression algorithm is used to compress the full-color still images which give good human visual perception (HVP) after the compression. Experimental results give the high quality compressed images and it is compared with different quality factors. Implementation of image compression on Raspberry Pi has its real time application of Micro Air Vehicles (MAVs).The proposed method helps MAVs to store large number of images captured by it.**

Keywords—JPEG compression, DCT, Raspberry Pi, Python, Open CV, quality factor.

I. INTRODUCTION

Image compression is the process of reducing the digital data volume of the image, as image contains the large amount of digital data. Unadulterated images require large space which is prohibitively cost more in terms of space. Therefore, there are different types of image compression techniques i.e. lossy and lossless image compression.

The main point of interest of this paper is most widely used JPEG compression algorithm for full-color still images. JPEG compression algorithm is centered on the DCT. DCT transforms the image into different frequencies then quantizes to remove all redundant pixels and further encoded as the bit stream, after encoding it is decoded and the reconstructed image is compressed image which will have the file size less than that of original image without affecting its quality measures.

Finally, implementing the JPEG compression algorithm on Raspberry Pi processor in Open CV platform using python as its programming language. Python coding channelize the image compression algorithm with Raspberry Pi Processor. Open CV provides modules for programming of algorithm.

The MAVs capture images using Raspberry pi camera as it is credit card size with less weight and provides flexibility for image processing operation. Because of this flexibility and efficiency it is used for real time process. MAVs are small sized Unmanned Aircraft Vehicle (UAV) which is remote controlled and restricted with size. MAVs are remote controlled by pilot in ground station otherwise by other MAVs.

The Raspberry Pi is placed in the MAV which helps to capture images. Raspberry Pi processor captures the image with storage space of 2.4MB where the total memory space is 512MB-1GB. Hence, it is important to compress images taken by Raspberry Pi. As it unable us to see its application in weather forecasting as we can capture many more image than normal and have precise readings.

Sahitya S, Bangalore Institute of Technology, Bengaluru-04, India (sahitya.siddaiah@gmail.com)

Lokesha H, National Aerospace Laboratories, Bengaluru-17, India (loki_hlmg@nal.res.in)

Sudha L K, Bangalore Institute of Technology, Bengaluru-04, India (sudhvenk@yahoo.co.in)

II. BASIC PRINCIPLE FOR IMAGE COMPRESSION ON RASPBERRY PI

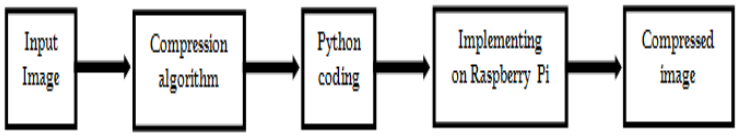

Fig 1. Basic block diagram for image compression on Raspberry Pi

Fig.1 explains the process of image compression on Raspberry Pi. The image captured by MAVs has to be compressed to store in large numbers. The images captured by MAVs will have raw data with more storage space hence it is necessary to compress image using DCT compression algorithm. The DCT algorithm is coded using Raspberry Pi programming language python. Thus implementing the image compression algorithm on Raspberry Pi generates the compressed image. The compression ratio and quality level can be controlled by different quality factors.

III. EXPERIMENTAL ANALYSIS

A. *Hardware Analysis*

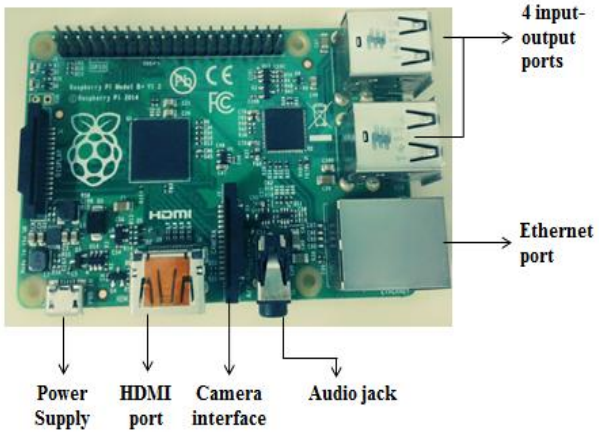

Fig 2. Raspberry Pi 2 board

978-1-5090-0774-5/16/$31.00 © 2016 IEEE

Raspberry Pi 2 model is the revised version of previous Pi models. It centralizes the embedded system module. Fig.2 shows Raspberry Pi 2 model consists of 4 input and output ports. More USB provides the overcurrent behavior and better hotplug. Micro SD version is changed to nicer push-push. Power consumption is reduced by using switches. Hence, power consumption is very less compared to other version i.e. 0.5W -1W. It has 1GB of Random Access Memory (RAM). It has a Quad-core ARM Cortex A7 CPU Processor with clock speed of 900MHz.ARMv7 allows the processor to run LINUX, snappy UBUNTU, and higher windows version i.e. Windows 10.Raspberry Pi processor requires its own Operating System (OS). Raspbian or NOOBS are the different version of OS which can be implemented [7].

B. Software Analysis

Software analysis deals with 2 different stages of analysis of software. First, let us discuss about the OS used. Raspbian is the OS which is used in Raspberry Pi 2, the OS is dumped in the SD card and then SD card is inserted into Pi board. After inserting the card, connect the necessary hardware components and provide power supply. Therefore, OS will be installed and it starts booting. Once the booting is finished it asks to login using default username and password. Once the board is setup, check out with network setting and network configuration to access internet. So that it will work with real time application.

Secondly, software analysis deals with the Open CV platform which supports different programming languages like c/c++, python, Java, android etc. It also helps to access MATLAB modules. We are using the Open CV 2.8.10 version. Open CV helps the Python coding much simpler by providing predefined modules which saves the processing time and memory [6].

IV. METHODOLOGY

C. Implementation of Algorithm

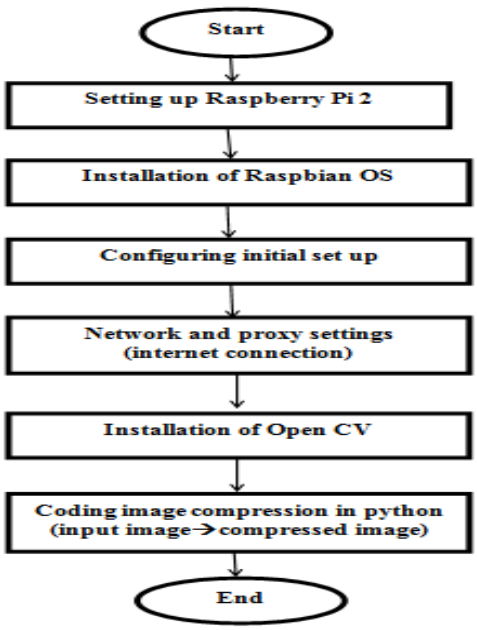

Fig 3: Flow Chart for Implementation of Algorithm

Implementation of software on hardware gives the complete process of image compression on Raspberry Pi 2. Fig.3 shows flow chart which defines the flow of the process takes place for implementation. As it is shown in flow chart first installation of the OS then it is followed by configuring the network and proxy setting for internet connection. Installation of Open CV provides platform for coding the image compression algorithm in Python. Thus, the image is compressed to increase storage, to store large number of images.

D. Compression Algorithm

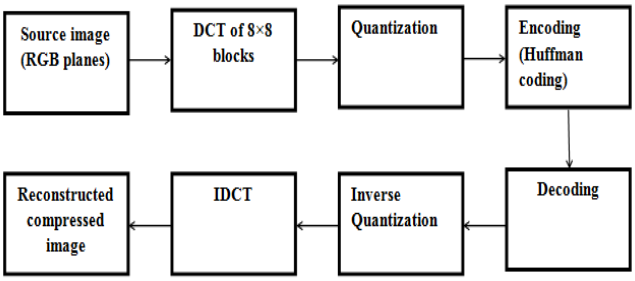

Fig 4: Basic block diagram for image compression

The compression algorithm concentrate on most widely used JPEG standard. The main reason to choose this method as it is used for non-analytical application of signal processing. Thus it helps to implement in Raspberry Pi for MAV application. JPEG images are obtained by discrete cosine transform. Main concern of DCT compression is to remove the statistical redundancy without affecting human visual perception. Statistical redundant pixels are the pixels having no information or repeated values of neighborhood pixel due to correlation of color panels [5].

Now, Fig.4 discusses the principle of DCT image compression. The source image will have three panels they are red (R), green (G), blue (B) these panels are divided into 8×8 blocks. DCT transforms the image into different frequencies i.e. lower frequencies and higher frequencies. Lower frequencies have good visual perception hence these frequencies should be retained and higher frequencies are removed. Higher frequency levels can be removed using the DCT equation given

$$D(s,t) = \frac{2}{N} C(s) C(t) \sum_{a=0}^{7} \sum_{b=0}^{7} p(a,b) \cos[\theta_1 s] \cos[\theta_2 t] \quad (1)$$

Where $\theta_1 = \frac{(2a+1)\pi}{2N}$, $\theta_2 = \frac{(2b+1)\pi}{2N}$,

p(a,b) represents the pixel value at spatial domain a and b, D(s,t) represent the transformed pixels into DCT co–efficient values, N matrix block range e.g. N=8 for 8×8 blocks.

Further these DC co-efficient values are quantized where major part of compression takes place. Quantization is rounding the co-efficient values to the nearest step size and normalizing it with standard quantization matrix. Usually the standard quantization matrix at quality level 50 is preferred to obtain the good visual quality image with better compression ratio.

Huffman coding is seen as variable length coding. This calculation encodes with reference of likelihood of event of comparable frequencies. The worth with more events is spoken to utilizing less number of bits and furthers the less happening values spoke to utilizing more bits.

Inverse of the above process gives the reconstructed image with less file size. The bits are decoded then inversely quantized and the image is obtained back using the inverse discrete cosine transform (IDCT) equation given below:

$$p(a,b) = \frac{2}{N}\sum_{s=0}^{7}\sum_{t=0}^{7} C(s)C(t)\, D(s,t)\cos[\theta_1 s]\cos[\theta_2 t] \qquad (2)$$

E. Quality Measures

There are two main quality measures that have to be considered during compression of an image. Quality factor and the compression ratio are two important measurements.

Quality factor: Quality of the image is maintained by quantization matrix. If the level of standard quantization matrix is at 50 then it is having the good visual perception. If the quality level is more than 50 then quality of the image will have best visual view and quality level is less than 50 then the image will have block effects.

Compression ratio: The ratio of uncompressed image (c1) to the compressed image (c2) gives the compression ratio

$$\text{Compression ratio} = \frac{c1}{c2} \qquad (3)$$

Where c1 is file size of uncompressed image and c2 is file size of compressed image.

Hence it is necessary to consider high quality images with very good compression ratio hence it is preferred to choose standard quantization matrix of level 50.Pecentage of image compressed is calculated using (1-compression ratio)100 [1][2].

V. RESULTS AND DISCUSSION

The image captured by the MAVs through the Raspberry Pi is compressed using the compression algorithm in developmental environment of Python in Open CV platform. The results are shown in Fig.5 with Dc co-efficient of RGB planes.

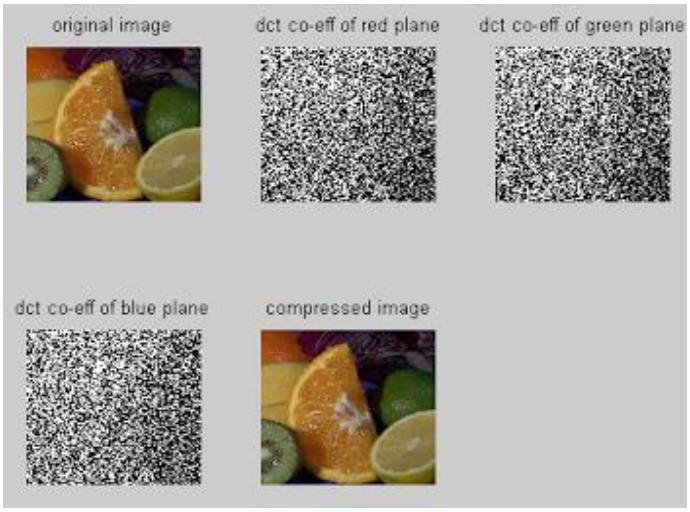

Fig 5: Outcomes of proposed compression algorithm

The image is compressed at different quality level using standard quantization matrix which gives high quality images with good compression ratio. The results after the compression algorithm is shown below

Fig 6: Compressed image at quality level 50

Fig 7: Compressed image at quality level 10

Fig 8: Compressed image at quality level 90

Consider an example of fruit image, original size of the image is 91 KB and the figures show compression at different quality level. Observing the results we can conclude as quality level increases compression ratio decrease. The compression ratio is converted to percentage as it represents percentage of image compressed. Fig.6 shows image at the quality level 50 file size is reduced to 67KB and compression is 26.3%, Fig.7 shows image at quality level 10 reduce file size to 40KB and compression is 56.04% and Fig.8 shows image at quality level 90 compression is 12% with reduced file size 80KB.

TABLE I. COMPARISON USING DIFFERENT QUALITY MEASURES

Sl.No	Quality Measures		
	Quantization Level	*File size*	*Compression percentage*
1	50	67	26.3
2	10	40	56.04
3	90	80	12

a.

VI. CONCLUSION

In this paper we have proposed a method for implementation of image compression on raspberry pi board using JPEG compression algorithm in python environment. As open CV has predefined modules which makes the coding of the algorithm simpler. The quality of the image can be maintained using different compression ratio and quantization level. Further algorithm can be designed for more advanced JPEG 2000 standards using Pywavelets [3] but there is restriction of using this method because compression of data needs lot of CPU power hence it limits implementation in Raspberry Pi. JPEG standard gives the good quality image with more compression ratio which helps in preserving large number of images.

REFERENCES

[1] NavpreetSaroya, PrabhpreetKaur."Analysis of image compression algorithm using DCT transforms." IJARCSSE ,Volume 4,Issue 2, Febraury 2014.ISSN:2277 128X.

[2] A.M.Raid, W.M Khedr, M. A. El-dosuky and WesamAhmed."JPEG image compression using discete cosine transform- A survey".IJCSES,Volume 5,No 2, April 2014.

[3] Himanshu M. Parmaur ."Comparison of DCT and Wavelet based Image compression techniques".2014 IJEDR,Volume 2,Issue 1,ISSN: 2321-9939

[4] NirbayKashyap,Dr .Shailendra Narayan Singh ."Review of Image compression and comparison of its algorithms". IJAIEM ,volume 2,Issue 12,December 2013.ISSN 2319-4847.

[5] Pratishtha Gupta, G.N Purohit,Varsh Bansal ,"A survey on image compression techniques".IJARCCE, volume 3,Issue 8,August 2014. ISSN 2278-1021.

[6] Open Cv tutorials documentation by Alexander Mordvintsev & Abid K, September-2015.

[7] www.raspberrypi.org