



# **Design and Implementation of Novel Video Compression Technique Using Raspberry Pi**

Shiva Kumar N R<sup>1</sup>, Venkatesh Murthy N K<sup>2</sup>

M.Tech. Scholar, Dept. of EIE, Bangalore Institute of Technology, Bangalore, India<sup>1</sup>

Asst. Professor, Dept. of EIE, Bangalore Institute of Technology, Bangalore, India<sup>2</sup>

**ABSTRACT:** The tremendous advances in computer and communication technologies have led to development of numerous products and services which perks the digital media in everyday life. In case of digital video, the demand for resources such as processing power, bandwidth and power consumption is high. In order to minimize these resources, video data is compressed and transmitted. Now a day, video compression is carried out using motion-compensated block-based Discrete Cosine Transform (DCT) in MPEG and ITU-T video coding techniques. This design employs complex encoders and lightweight decoders suited for broadcasting applications. The applications such as sensor networks, wireless video surveillance cameras and mobile communication devices possess highly constrained resources. Hence, MPEG and ITU-T video coding standards are not suitable for implementation. Therefore, new novel encoding techniques are to be developed to fulfill the requirements of the applications by improve the compression efficiency of the system.

Hence, in this paper, new novel video compression techniques are proposed since it effectively compresses the video data by saving the bandwidth of the wireless sensor networks (WSN). The proposed video compression algorithms are implemented on raspberry pi platform using python language and communicated wirelessly to the base stations (laptop) using low power Zigbee radio trans-receivers. The performance of the implemented video compression techniques are studied by considering performance matrices such as compression ratio, space saving ratio, mean square error and peak signal to noise ratio.

**KEYWORDS:** Raspberry PI, Zigbee trans-receivers, Raspberry pi Camera Module, Python.

## **I. INTRODUCTION**

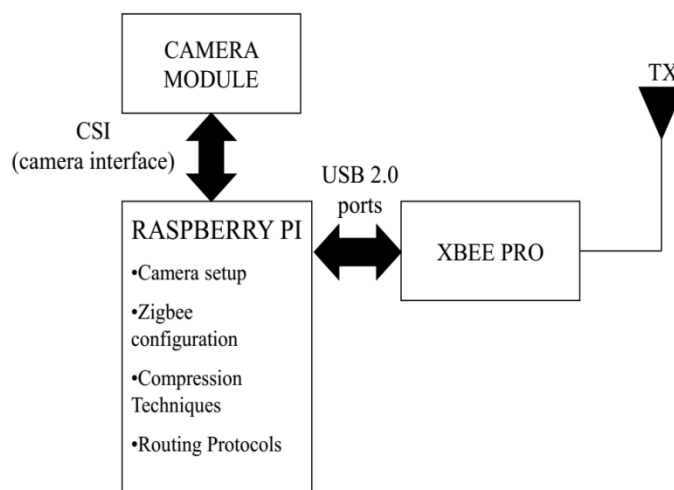
Video surveillance systems play an increasingly important role to maintain social security. It has been widely used in many fields, such as finance, public security, banking, and many other fields. Video surveillance can generally achieve close distance monitoring, by using the PC as a monitor host, monitor host connected to Raspberry pi camera module. The development of IT technology and electronics, especially the rapid development in CPU of the open systems such as Linux systems, video conferencing, remote monitoring, and mass data processing fields can apply this technology. This video monitoring system is designed to overcome the weak points of the Traditional video surveillance systems, such as complex structure, poor stability, expensive cost and high bandwidth. Meanwhile, it possess some advantages, for example, hardware and software can be cut, compact construct, portable, low power consumption, more suitable for low data rate Networks. In this system, we use the high-performance Raspberry PI as the microprocessor. Video data is captured from a Raspberry pi camera, compressed using DCT and DWT techniques in Python. The compressed video is transmitted using Zigbee and is received using another Zigbee which is connected to another Raspberry pi or PC to show the output as shown in Figure 1 and 2. Wireless video monitor system provide practical solution for wireless monitoring [1].

The video capture is the core part of the system. The camera capture video data, the data compressed, and transferred into a video stream format, this is Radio Frequency transmission. Zigbee radio transceiver module is used for wireless network module [2]. Data can be transmitted into the frame format and sent to the wireless network. The data arrival, the receiving monitor through high-speed network transmission, can also be a data center. The system structure is shown in Figure 1 and Figure 2. The video signal monitor of the entire system is under the control of Raspberry PI.

# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

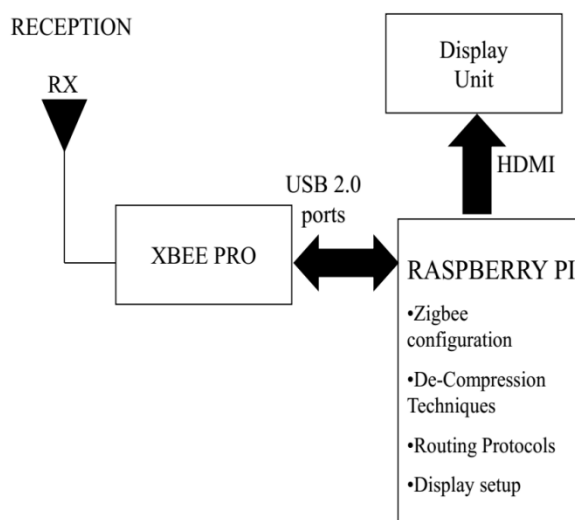
(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016



**Fig1. Video capture and Transmitter**

The above Figure 1 shows the transmission system.



**Fig2. Receiver end with display**

The above Figure 2 shows the Reception system. The transmission system captures the video and compresses the video and transmits it to the reception system through Zigbee. Whereas Reception system receives the video through another Zigbee and decompresses the compressed video and sends it to the display unit.

## II. THE SYSTEM DESIGN

### 1. Raspberry pi

The Raspberry PI B+ is a credit-card sized microcomputer and its Size is 8.6 X 5.4 X 1.7cm. It has ability to work as same as Pentium computers. Raspberry pi module offers outstanding features with its CPU core, a16/32 bit ARM11 RISC processor designed by ARM, Ltd. The Raspberry Pi Model B was utilized in the researcher's project and it has 512MHz SDRAM, and 700MHz CPU which allows a typical computing platform. The external ports of Raspberry pi module being used in this research are the CSI Camera port, HDMI port, Quad USB port and Ethernet port. CSI camera and HDMI ports are used for capturing video and Quad USB port was used to connect

# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

Raspberry pi to the Zigbee trans-receiver or to the networking device. Shown in Figure 3 is the picture of Raspberry pi.

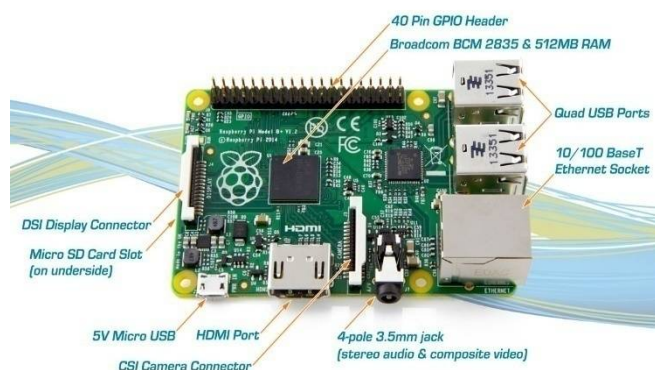


Fig3. Raspberry Pi B+

## 2. The Hardware design and Implementation.

We need 'Raspberry pi 2 Model B+' and preloaded SD card to it, USB keyboard, USB mouse and Monitor with HDMI/ DVI/ Composite/SCART input, and power supply as USB charger or a USB port from a powered USB Hub or another computer. For Internet connection, we'll need either an Ethernet/LAN cable (Model B). Raspberry pi works as the heart of the system. Camera module is connected to Raspberry pi to capture the image and video. A monitor connected to the Raspberry pi will display the image, video and python programming in which we write codes. The captured video is compressed by implementing compression techniques such as DCT and DWT. The compressed video is transmitted to another Raspberry pi or CPU through Zigbee trans-receivers. The received video is reconstructed and displayed in the monitor. Figure 4 shows the design and Implementation.

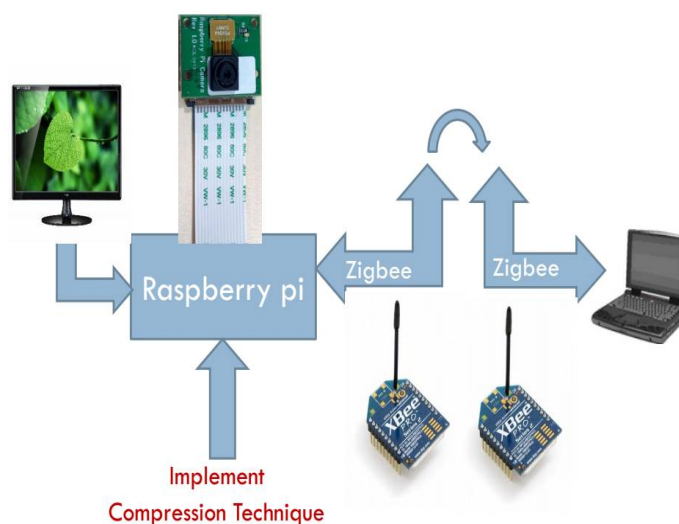


Fig4. System Design and Implementation

## 3. Software

The software used in this system is python 2.7.10 or 2.7.11. Python is a widely used high-level programming, general purpose, and interpreted, dynamic programming language. Its design philosophy emphasizes code readability, this enable us to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. Python is one of the main programming languages used to program Raspberry Pi.



## International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

### 4. Video Compression

Video clips are made of sequences of frames where each frame can be considered as an individual image. In this system implementation of the JPEG compression encoding scheme is used in the system based on the Discrete Cosine Transform (DCT). The original image is divided into 8×8 small video image after DC Transform. Among all the components the low frequency components are concentrated in the upper left corner, and high-frequency components are concentrated in the lower right corner. Then apply image compression to each and every frame. For the given set of video frame, First frame is extracted and image compression is applied to it. Similarly the second frame is extracted and image compression is applied.

This method is repeated for all the frames. Hence this method is called as frame based compression technique or spatial compression technique. Also spatial redundancy is eliminated in this method. Video compression is done by using DCT and DWT compression Techniques. After Quantified at high spatial frequencies a large numbers of consecutive zero will appear. The system uses Huffman variable length coding to encode the quantized DCT coefficients. So the both encoding and decoding are done by using Either DCT or DWT.

### 5. Setting Up Boot

We should boot Raspberry pi before using first time.

The boot partition steps must contain:

- i. Start by slotting your SD card into the SD card slot on the Raspberry Pi.
- ii. Plug your USB keyboard and mouse into the USB slots on the Raspberry Pi.
- iii. You should make sure that your monitor or TV is turned on, and that you have selected the right input.
- iv. Connect your HDMI cable from your Raspberry Pi to your monitor or TV.
- v. When you are happy that you have plugged in all the required cables and SD card, plug in the micro USB power supply. This action will turn on and boot your Raspberry Pi.
- vi. If this is the first time your Raspberry Pi and NOOBS SD card have been used, then you will have to select an operating system and configure it.

```
[ 10.879675] EXT4-fs (sda2): warning: mounting fs with errors, running e2fsck is recommended
[ 11.944667] EXT4-fs (sda2): recovery complete
[ 12.236880] EXT4-fs (sda2): mounted filesystem with ordered data mode. Opts: (null)
[ 12.251682] UFS: Mounted root (ext4 filesystem) on device 8:2.
[ 12.265387] Freeing init memory: 124K
[ 14.357999] bcm2708_spi bcm2708_spi.0: SPI Controller at 0x20204000 (irq 60)
[ 14.634941] bcm2708_i2c bcm2708_i2c.0: I2C0 Controller at 0x20205000 (irq 79)
[ 14.650315] Registered led device: led0
[ 14.652657] bcm2708_i2c bcm2708_i2c.1: I2C1 Controller at 0x20209000 (irq 79)
[ 14.914493] smsc95xx v1.0.4
[ 15.034594] input: Logitech USB Receiver as /devices/platform/bcm2708_usb/usb1/1-1/1-1.3/1-1.3:1.6
[ 15.212642] smsc95xx 1-1.1:1.0: eth0: register 'smsc95xx' at usb-bcm2708_usb-1.1, smsc95xx USB 2.0
[ 15.375477] generic-usb 0003:046d:C52E:0001: input,hidraw0: USB HID v1.11 Keyboard [Logitech USB
[ 15.492642] usbc0re: registered new interface driver smsc95xx
[ 15.659262] input: Logitech USB Receiver as /devices/platform/bcm2708_usb/usb1/1-1/1-1.3/1-1.3:1.6
[ 15.739923] generic-usb 0003:046d:C52E:0002: input,hiddev0,hidraw1: USB HID v1.11 Mouse [Logitech
[ 15.767989] usbc0re: registered new interface driver usbhid
[ 15.781524] usbhid: USB HID core driver
[ 17.805139] mousedev: PS/2 mouse device common for all mice
[ 21.785456] EXT4-fs (sda2): re-mounted. Opts: (null)
[ 24.075114] FAT-fs (mncbl0p1): utf8 is not a recommended IO charset for FAT filesystems, fileys
[ 28.142672] ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 28.931291] fuse init (API version 7.17)
[ 46.428987] sshd (741): /proc/741/oom_adj is deprecated, please use /proc/741/oom_score_adj instea
[ 108.664416] i2c /dev entries driver
[ 130.223666] rtc-pcf8563 0-0051: chip found, driver version 0.4.3
[ 130.225380] rtc-pcf8563 0-0051: low voltage detected, date/time is not reliable.
[ 130.225993] rtc-pcf8563 0-0051: rtc core: registered rtc-pcf8563 as rtc0
[ 130.226653] i2c i2c-0: new_device: instantiated device pcf8563 at 0x51
root@raspberrypi:~# hctlock
Fri 08 Jun 2012 09:18:03 BST -0.952736 seconds
root@raspberrypi:~# hctosys
root@raspberrypi:~# date
Fri Jun 8 09:18:35 BST 2012
root@raspberrypi:~# _
```

Fig.5 System Running



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

## III. EXPERIMENTAL RESULTS AND ANALYSIS

The system running is shown in Figure 5. In this method (Video compression using Raspberry pi), Bandwidth and power consumption is reduced. Transmission speed, Resolution and compression efficiency is increased than any other compression methods. Performance evolution of frame based DCT, frame based DWT and stream based DWT video compression algorithms are calculated using Python by considering different videos, each with a resolution of 144 X 176 pixels and with a frame rate of 10fps. The results after reconstructing a compressed video are Threshold=10; CR=2.7880; PSNR=35.8564. This result is different for different videos. The metrics considered for performance evolution are Compression ratio, spacing Ratio and Mean Square Error.

## IV. CONCLUSION

The Novel Video Compression and Wireless Transmission system is designed based on Raspberry pi 2 B+. The frame based and stream based video compression algorithms with DCT and DWT have been implemented using Python 2.7.10 and its performance metrics such as Compression ratio, Spacing ratio, Mean square error and Peak signal to noise ratio are calculated on the basis of video compression. The designed system is stable, reliable and clearer performance on video processing when compare to previous methods.

## REFERENCES

- [1] EURASIP Journal on Image and Video Processing | Full text | Distributed Video Coding: Trends and Perspectives.
- [2] Hot Spot Visual Aid Technique using ARM11 (Raspberry Pi) RAMALAKSHMI PALISETTY1, B. UPENDER R 2
- [3] Zho.ng Chongquan. Research and application of remote monitoring based on embedded web. Chinese Journal of Scientific Instrument, Vol. 27, pp. 575-577, June. 2006.
- [4] INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 1, ISSUE 10, NOVEMBER 2012 ISSN 2277-8616 114 IJSTR©2012 www.ijstr.org
- [5] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, High Efficiency Video Coding (HEVC) Text Specification Draft 9, document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct. 2012.
- [6] Video Codec for Audiovisual Services at px64 kbit/s, ITU-T Rec. H.261, version 1: Nov. 1990, version 2: Mar. 1993.
- [7] Jinxue Zhan, Ming Zhang. Research and design of embedded tank car monitoring system based on ARM9. 2009 International Symposium on Computational Intelligence and Design. ISCID2009. 2009(2):292-295
- [8] Peng Daogang, Zhang Hao, Zhang Kai, Li Hui. Research and development of the remote I/O data acquisition system based on embedded ARM platform. 2009 International Conference on Electronic Computer Technology I C E C T 2009, 341-344.
- [9] Chen Rongfang, Wang Haibin, Hu Zhenhua, Zheng Pin. Remote Monitor System of Power Network Based on Embedded Internet Technology. Microcomputer Information. vol. 24, no. 7-2, pp. 78-80, 2008
- [10] Zhang YuJie, Huai Weihua, Tian Ze. Design of Industrial Temperature Monitoring Equipment Based on Embedded Linux Operating System. Computer Engineering and Design, vol. 30, no. 17, pp. 3940-3942, 2009.
- [11] Micro2440 user manual-2010-3-8, Guangzhou: Guangzhou friendly arms of computer science and technology Co., LTD, 2010.
- [12] Zhu Yi, Zhao Jie. Design and Implementation of 802.11 Wireless Network Stack in Linux Kernel,” Computer Engineering, vol. 34, no. 1, pp. 140-144, 2008.
- [13] Li Shanping, Liu Wenfeng. Linux and embedded system [M]. Beijing: Tainghua University Publishing house, 2006.
- [14] Dong Xie. ARM-based embedded remote monitoring and control system design Gateway. Modern electronic technology, 2006, pp. 85-88.