Krzysztof SIECZKOWSKI, Tadeusz SONDEJ
ELECTRONIC DEPARTMENT, MILITARY UNIVERSITY OF TECHNOLOGY
2 gen. Sylvester Kaliski St., 00-908 Warsaw 49, Poland

Use of a Raspberry PI single-board computer for image acquisition and transmission

Abstract

The article presents an analysis of the use of a single-board computer Raspberry PI for video acquisition and transmission. The article focuses on requirements necessary for the recorded image to be used for face analysis to identify facial expressions and microexpressions. The quality of the recorded video frames was verified for different resolutions and fps using PSNR (Peak Signal-to-Noise Ratio). Tests of CPU cores usage were also carried out for simultaneous recording and transmission of different types of video streams. The results show that the size of the effective image area depends on the resolution of recorded video stream. Increasing the frame rate for the given video resolution has a significant impact on the value of PSNR. And the resultant CPU usage, for the available resolutions and frame rates of the video recorded, in most cases does not exceed 15% of the total computing power of the CPU.

Keywords: Raspberry Pi; single-board computer; video monitoring; image processing.

1. Introduction

Single-board computers (SBC) are being used more and more frequently. The size of SBC computers often does not exceed the size of a credit card. They offer both a relatively high computing power (in relation to the mid-range PCs), and their price usually does not exceed $100. Currently, on the market we can observe a significant increase in the interest in SBC computers by many manufacturers. They offer devices containing main multi-core CPUs in their structure. Furthermore, additional GPUs and RAM are often placed in one housing, together with the main processor, creating a powerful SoC (System-on-Chip). The use of SoC not only reduces the size of the entire device and increases reliability, but also significantly reduces power consumption.

The beginnings of the age of SBC computers can be traced to the 1990s, however, significant interest in SBC computers was observed in 2012, after the introduction of the first model of the Raspberry PI [8]. Even then, the Raspberry PI foundation offered a dedicated operating system, sample software and very good technical support for beginners. The success of the aforementioned SBC computer was also related to its exceptionally low price of $25 and the small size of the device. Today, Raspberry PI is offered in several basic versions and revisions of the electronic board. The main difference between the versions of the Raspberry PI computer is the SoC used. This system - depending on the computer version - includes a single-core ARM11 processor, a quad-core ARM Cortex-A7 processor or a quad-core ARM Cortex-A53 processor. The amount of RAM is also not constant and ranges from 256 MB to 1 GB. RAM in each case is shared with the Broadcom VideoCore IV GPU, offering computing power of 24 or 28.8 GFLOPS.

Currently there are many solutions of SBC computers similar to Raspberry PI on the market, e.g. computers called Banana PI, Odroid, BeagleBone, CubieBoard, Orange PI. They primarily differ by the SoC used, available memory and operating system.

The range of applications of SBC boards is enormous. It includes, for example, control systems [2], monitoring systems [4] and video processing systems [6], [5]. This article focuses on the use of Raspberry PI to record video for the purposes of facial analysis, e.g. to determine facial expressions and microexpressions. A sample solution for face detection is presented in [1], but it is not sufficient for rapid and advanced facial analysis.

2. Image acquisition using Raspberry PI

In SBC computers intended for image acquisition and transmission, key role is played by video camera and the interface to which it is connected. Currently, USB and CSI (Camera Serial Interface) are the most commonly used. Type of camera used also depends to a large extent on intended use of the target device. This article focuses on the acquisition of video for face analysis to identify facial expressions and microexpressions. For this reason, specific requirements regarding the quality of the recorded image must be met. Depending on the nature of the further processing, is often required to register only the specified area of the face, e.g. mouth, eyes. Also, the frame rate of the recorded image is not permanently defined. However, wanting to capture behavioral changes of the face, we have to record with a minimum speed of about 100 fps [3]. Depending on the area of the analyzed face, or frame in which the face is located, image must be recorded in the highest possible resolution. If additional parameters determining the choice of a video camera will be the size and price of the device, then the search should be narrowed to SBC computers with USB or CSI cameras connected. Currently, there are many USB cameras available on the market. Unfortunately, they are often offered for devices operating only under Windows. In the case of other systems, the availability of the required software is practically very small. There are also cameras offer support for Linux, but the parameters of the recorded image are often limited to specific values. Also, cameras of this type often have a limited ability to change the parameters of the recorded video sequences.

A SBC computer Raspberry PI version 2 model B was used for research, along with a wide-angle video camera NightVision H [9], connected via an CSI. A VideoCore GPU in the SoC of the Raspberry PI has hardware support for a decoder and video encoder. The use of multimedia components, including a CSI camera, encoder, video decoder, is implemented using the MMAL (Multi-Media Abstraction Layer) framework, which provides low-level software procedures allowing for easy use of these components. For example, during the recording - on the programmer's side - the callback function is only available upon registration of a new video frame. During the period which is a reversal of the frame rate, copy the current picture frame from the local memory of the camera (memory reserved by MMAL) to another storage location.

The NightVision H camera module used is equipped with two infrared illuminators (IR), allowing for image recording in the dark. The camera module includes a CCD (Charge Coupled Device) type OV5647 [7] with a resolution of 5 megapixels. It is thus possible to record a single frame of the image in native resolution (2592×1944, max. 15 fps), as well as recording Full HD video (1920×1080 max. 30 fps). The NightVision H module is equipped with a "fish eye" lens, allowing to increase the viewing angle (160° diagonal, 120° horizontal). However, a NightVision H camera coupled with a Raspberry PI allows for recording Full HD picture at the most. The rate of frames per second is closely associated with the selected video resolution, e.g. 1280×720 –max 60 fps, 640×480 – max 90 fps, 320×240 – max 120 fps.

In order to transmit image with parameters offered by the NightVision H camera, while maintaining 8-bit encoding of each color component (R, G, B) for each pixel, we must provide an adequate bandwidth. For Full HD picture at 30 fps, it amounts to approximately 1424 Mb/s, while for 320×240 resolution at 120 fps, the required bandwidth is about 211 Mb/s. With such a high demand for bandwidth, saving data in the form of an uncompressed RGB image is not possible in memory. Compression of image frames is therefore required to reduce the required bandwidth while maintaining acceptable image quality.

The VideoCore GPU used in the Raspberry PI includes hardware support for an encoder and video decoder. It is possible to encode an uncompressed video stream from a CSI camera to the lossy compression format H.264 or MJPEG. Since the assessment of the Raspberry PI computer has been carried out in regard to acquisition and transmission of video for further analysis of facial expressions and microexpressions, the MJPEG video format has been selected as format of the target video stream. MJPEG compression includes independent JPEG picture frames. This allows for analyzing an individual video frame without conducting a preliminary decoding of previously recorded frames. The hardware image encoder of the VideoCore processor can encode a video stream received from a camera to MJPEG format with a maximum rate of 25,000,000 b/s. Thus, depending on the number of fps for the given resolution, the average size of a single video frame is also different. The amount of data describing a single video frame of a certain size has a significant impact on the overall quality of the recorded video frame.

3. Measurement setup

Schematic diagram of the test stand is shown in Fig. 1. The test stand includes a Raspberry PI computer with NightVision H camera connected, mounted on a fixed tripod above the test image.

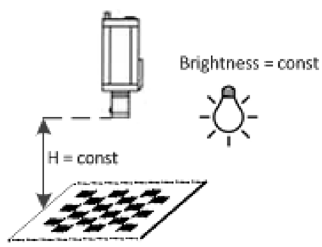

Fig. 1. Block diagram of the test stand

The distance between the camera and the recorded test image is constant and amounts to about 20 cm. The test image was illuminated by artificial light of a constant brightness.

A test of the effective area recorded by the camera was conducted for different resolutions and frame rates. A black-and-white checkerboard image was recorded. The distance of the recorded board from the image was also the same in each case. Video was recorded in 4:3 and 16:9 format. The resolution of recorded frames was within: 320×240 – 1280×960 and 640×360 – 1920×1080. For video sequences recorded in 16:9, there was a different effective area of the recorded image. For the highest resolution (1920×1080 and 1600×900), the effective area of the recorded image was smaller compared to lower resolutions. The camera recording a video sequence in Full HD maintained a smaller angle with respect to lower resolutions. In the case of recording video sequences in 4:3, the effective area of the recorded image did not change. This effect is illustrated in Fig. 2.

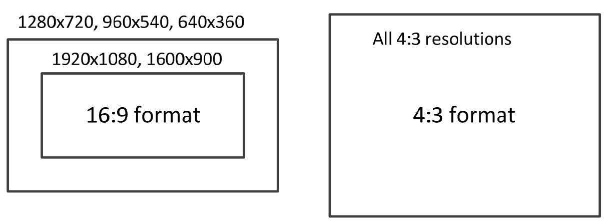

Fig. 2. The impact of frame rate (fps) on the quality of the recorded video stream

Recording uncompressed video frames, due to the required bandwidth in Raspberry PI, is not possible. There is thus no need for lossy compression, which is supported by the GPU. A test was conducted of the impact of the size and frame rate of the recorded video sequence on the quality of the resulting stream. PSNR (Peak Signal-to-Noise Ratio) was used as a measure of the quality of the video stream, defined with the formula: $PSNR=10\log_{10}(peakval^2/MSE)$. The value of *peakval* for monochrome image represented using an 8-bit word is 255. The MSE parameter is the mean square error calculated with respect to each pixel in the given image frame. A reference standard was used in order to determine the mean square error, i.e. one image frame recorded without compression. A second image frame was also required - after JPEG compression. The image reference standard and the MJPEG stream were recorded under the same conditions - independently for each resolution and frame rate.

4. Results

Two types of images were recorded. The first with a relatively regular structure - black-and-white checkerboard (BW) and the second - color, showing several Raspberry PI 2 boards arranged next to each other (RPI). The measurement was made for automatic camera settings. Tab. 1 shows the effect of resolution and frame rate on the quality of the recorded video stream.

Tab. 1. The effect of resolution and frame rate on the quality of the recorded video stream

Video stream	Size of RAW image, B	Average size of JPEG image, B	Compression ratio	PSNR BW, dB	PSNR RPI, dB
480p30	921 600	58 485	15.80	25.17	18.99
480p60	921 600	.47 806	19.30	15.12	16.53
480p90	921 600	32 251	28.60	14.88	18.28
720p30	2 764 800	96 408	28.70	25.92	19.77
720p60	2 764 800	50 122	55.20	6.33	13.71
1080p15	6 220 800	199 131	31.24	25.89	11.71
1080p30	6 220 800	94 502	65.83	24.95	13.51

A different average size of a single JPEG frame was obtained depending on the resolution and frame rate of the video stream. This results in a different value of the compression ratio. Within the specified resolution, an increase of the number of frames reduces the value of the PSNR. This is related to the average number of data attributable to the image description, as low-frequency information (the color of a single macroblock) is described first, then the data on higher frequencies (more details).

A significant impact of the number of fps on the value of the PSNR was observed for the lowest resolutions.

The Raspberry PI computer used, because of its multi-core processor (model 2 and 3), can perform complex operations. A test of processor usage in the SoC was conducted during image acquisition for different resolutions and frame rates. This test was carried out in the case of recording the video stream onto an internal memory card µSD along with simultaneous transmission via Ethernet with an active UDP protocol. To measure system usage, values updated by the Linux kernel in the */proc/stat* file were used. The stat file contains the time of operation of all processor cores, counted from the boot, for a given activity, also measuring the idle time of a given core. Since version 2.6.33 of the Linux kernel, the stat file contains 10 columns defining times for the different processor states. Calculation of usage is done using the quotient of the sum of operating times of a given core performing certain tasks $t_{activity}$ (without idle time t_{idle}) and the sum of the operating times of a given core including idle time (1).

$$CPU(n)_{load} = \frac{t_{activity}}{t_{activity}+t_{idle}} \quad (1)$$

Division of dedicated time of a given core for certain types of activities allows pinpointing the cause of CPU usage. Results of processor usage tests while saving to volatile memory and with simultaneous transmission of recorded frames are shown in Fig. 3.

The main processor used in Raspberry PI 2 is a 4-core system based on ARM Cortex-A7 cores operate at a frequency of 900 MHz. The main tasks in the test program were allocated to the respective processor cores. Core CPU1 and CPU2 has been designed to handle the test program and the possibility of communication with the user. In addition, core CPU1 performed the tasks associated with Linux. Core CPU3 received data from dedicated camera memory. Data transmission via Ethernet (UDP) was additionally implemented with the help of core CPU3.

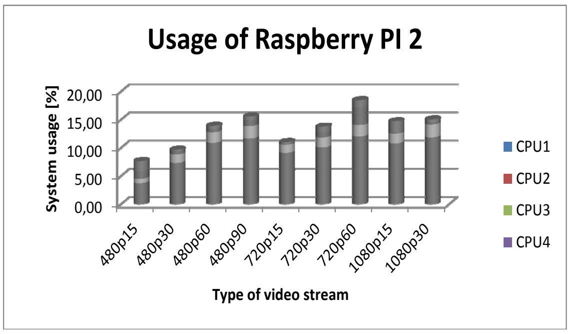

Fig. 3. Impact of the type of recorded video stream on the usage of individual cores of the SoC in Raspberry PI 2

Core CPU4 was dedicated only to writing data in non-volatile memory of the device. Load on core CPU4 resulted almost exclusively from the waiting time for access to I/O resources resulting from the write speed on the memory card. System usage in the graph presented has been standardized to 100% of the CPU's total computing power, so the maximum load of a single core is 25% of CPU resources. The results indicate that the biggest usage of the respective core is connected with the recording data in read-only memory. The usage of core CPU4 - depending on the type of stream - is in the range of from about 4 to 12% of the total computing power (16 to 48% of core CPU4 resources). The impact of handling image frames and transmission through the network interface on the total computing power usage ranges from about 1% to slightly more than 2%. (4 to about 8% of core CPU3 resources). In most cases - to transmit any video stream - the overall usage of the system was less than 15.5%. The exception is the video stream with a resolution of 720p at 60 fps, where we've seen a significant increase in CPU2 usage.

5. Conclusions

Raspberry PI can be used as a device for image recording and acquisition, for example, for face analysis. It is possible to record the image in a wide range of resolutions and frame rates. It was observed, however, that depending on the selected resolution in 16:9 format, the area of the recorded image changes. This area did not change for 4:3 format. This implies the need for additional manual adjustment of the distance after changing image resolution. In many cases, an increase in frame rate in a given resolution adversely affects the PSNR. The greatest differences were observed for 480p. In Raspberry PI, saving data of the recorded image frames has the biggest impact on the usage of the entire system, absorbing from 16 to 48% of the computing power of a given core. Saving image frames with the simultaneous transmission via the UDP interface - depending on the parameters of the video stream - absorbs from 4 to about 8% of the resources of the given core in the SoC. The total CPU usage (all cores) in most cases does not exceed 15.5%.

This work was supported by the Polish National Centre for Research and Development under contract no PBS3/B9/29/2015.

6. References

[1] Janard K., Marurngsith W.: Accelerating Realtime Face Detection on a Raspberry Pi Telepresence Robot. International Conference on the Innovative Computing Technology, pp. 136-141, 2015.

[2] Michalak S.: Mikrokomputer Raspberry Pi jako sterownik systemu pomiarowego. Pomiary Automatyka Kontrola, vol. 60, nr 8, str. 649-651, 2014.

[3] Pfister T., Xiaobai Li, Guoying Zhao, Pietikäinen M.: Recognising spontaneous facial micro-expressions. IEEE International Conference on Computer Vision (ICCV), pp. 1449-1456, 2011.

[4] Prinz A.C.B., Taank V.K., Voegeli V., Walters E.L.: A novel nest-monitoring camera system using a Raspberry Pi microcomputer. Journal of Field Ornithol., vol. 87, no. 4, pp. 427-435, 2016.

[5] Sahitya S., Lokesha H., Sudha L.K.: Real Time Application of Raspberry Pi in Compression of Images, IEEE Inter. Conf. On Recent Trends In Electr., Infor. & Comm. Tech., pp. 1047-1050, 2016.

[6] Szabo R., Gontean A.: Industrial Robotic Automation with Raspberry PI using Image Processing. 21ST International Conference On Applied Electronics, pp. 265-268, 2016.

[7] OV5647 Product Brief http://www.ovt.com/uploads/parts/OV5647.pdf, 2017.

[8] Raspberry Pi Foundation https://www.raspberrypi.org, 2017

[9] Waveshare Electronics http://www.waveshare.com/rpi-camera-h.htm, 2017.

Received: 16.02.2017 *Paper reviewed* *Accepted: 04.04.2017*

Krzysztof SIECZKOWSKI, MSc

Born 20th July 1990 in Łowicz. Received his BSc and MSc degrees in electronics and telecommunications from Military University of Technology, Poland, in 2014 and 2015, respectively. He is currently working toward the PhD degree in electronic engineering. His main scientific interests are measurement and analysis of biometric signals, especially detection strokes in blood pressure and energy-efficient microprocessor systems.

e-mail: krzysztof.sieczkowski@wat.edu.pl

Tadeusz SONDEJ, PhD

He received the MSc degree in electronic engineering and the PhD degree in applied sciences from the Military University of Technology (MUT), Warsaw, Poland, in 1997 and 2003 respectively. Since 1998, he has been with the MUT, where he has been working on design and programming of embedded systems. His current interests are in the field of design, optimization and programming of System-on-Chip based digital systems, especially for biomedical applications.

e-mail: tadeusz.sondej@wat.edu.pl