Video Capturing And Long-Distance Display System Based On Improved H.264

Jian-fei Mao, Jie Zhang, Rong-hua Liang
Department of Computer Science and Technology
Zhejiang University of Technology
Hangzhou, China
e-mail: zhangjie433@gmail.com

Zhi-rong Liao
Department of Computer Science and Technology
Zhejiang Changzheng Vocational and Technical College
Hangzhou, China

***Abstract*—A feasible and novel design and implementation scheme of embedded video surveillance system, which is based on improved H.264 and Wi-Fi, is proposed in this paper. A video server, built on an ARM11 platform with a wide range of peripheral interface, is responsible for capturing image data using Video for Linux Two API, provided by the kernel, with a USB camera. The compressed video data encoded by X264 library is transmitted to clients through RTP protocol and Wi-Fi wireless network. Based on in-depth examination about UMHexagonS, the fast motion estimation algorithm for H.264/AVC, two improvements are proposed and implemented in the X264 used in this system to further improve coding efficiency. On the client side, the encoded data is firstly saved for future use and then decoded using FFmpeg to get the raw data, which should be conducted format conversion in order to be displayed using GTK on the screen. The experimental results indicate that the embedded system runs stably, have higher speed and is cost-effective and small in size, all these merits making it significant in practical use.**

Keywords-improved UMHexagonS; Wi-Fi; image acqisition; H.264; x264; FFmpeg

I. INTRODUCTION

With the rapid development of science and technology, people's living standard has been greatly improved. The security issue has attracted more and more attention. Because of its intuitionistic, convenient and rich-information feature, the video surveillance system has been widely used in many places, such as school, bar, personal home [1]. During the past decade, ARM and Wi-Fi technology has achieved vast improvement, making it feasible and more convenient to implement a novel embedded wireless video surveillance system on an ARM platform, which supports network, Wi-Fi wireless communication, formidable image processing capability, and so on. Because of massive data needs to be communicated through network, video encoding is essential and incorporated into the video surveillance system. It uses the newest video compression standard-H.264/AVC, which not only enhances the network ability, but also significantly improves the coding efficiency, while getting higher subjective and objective quality under the same rate [2]. In this paper, a feasible scheme of embedded video surveillance based on improved H.264 and Wi-Fi is proposed and implemented, and two improvements on UMHexagonS algorithm are implemented using X264 encoding library for further performance gain.

In section 2, the structure of software system is introduced. Two improvements on UMHexagonS algorithm are proposed and discussed in section 3. Modules of server and client side are covered in the section 4, followed by the conclusion in section 5.

II. THE SOFTWARE SYSTEM ARCHITECTURE

The development board used for the video surveillance system is based on S3C6410 microprocessor chip, which is very powerful and popularly used in embedded applications. The board produced by Hua Tian Zheng Technology Company has integrated many kinds of functional modules, including Wi-Fi, LCD, Nand flash, etc., making it for users to develop their own applications easily and quickly.

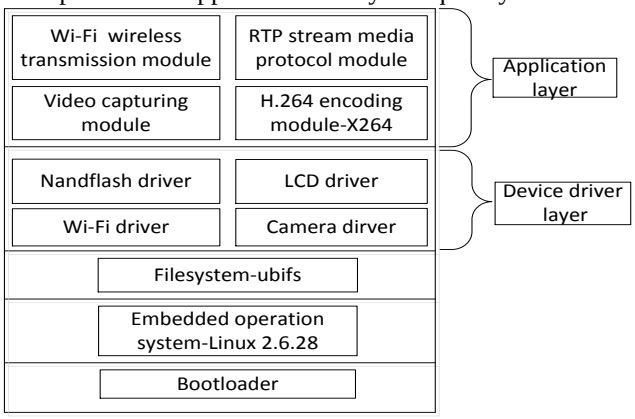

Figure 1. the software system architecture

When the system starts on power, it first loads the Bootloader snippet into memory to initialize hardware devices, create map table of memory space, load and install linux system kernel, then the overall system starts, and load some essential driver programs, such as nandflash driver, LCD driver, Wi-Fi drive, etc., as the Fig.1shows.

III. THE IMPROVED UMHEXAGONS (IUMHEXAGONS) ALGORITHM

It is popularly known that motion estimation is the most time-consuming part in H.264/AVC video encoding. In order to find the best matching block using the minimal time, many fast block-matching algorithms with different searching patterns and strategies have been proposed, including three step search (TSS)[3], new three-step search (NTSS)[4], 2-D logarithmic search (2-D LOGS)[5], four-step search (FSS)[6], diamond search (DS)[7], hexagon-based search (HEXBS)[8], block-based gradient descend search (BBGDS)[9], predictive motion vector field adaptive search

technique (PMVFAST)[10], etc. Otherwise, the hybrid unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) algorithm, which was proposed in [11], is a very good one to greatly reducing the number of points to be evaluated. Based on UMHexagonS algorithm, this paper proposes two improvements, which the experimental results show it can further decrease computational complexity. Let's first examine the UMHexagonS algorithm.

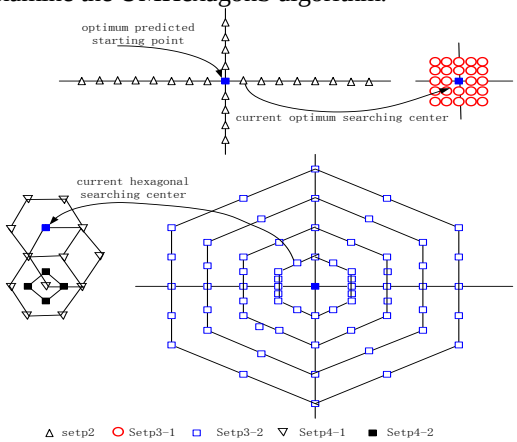

Figure 2. search pattern of UMHexagonS algorithm

The details of UMHexagonS algorithm can be summarized as follows [11][12]:

- Step 1: Initial search point prediction.
  Five efficient Motion Vector (MV) prediction modes are used to identify the starting search point, i.e., the Median Prediction (MP), the UpLayer Prediction (UP), the Corresponding-block Prediction (CP), the Neighboring Reference-picture Prediction (NRP) and the Zero-valued Motion Vector Prediction (ZMVP) [13]. The motion vector with the smallest distortion is chosen to be the starting point to perform the next step.
- Step 2: Unsymmetrical-cross search
  As Fig.2 step2 shows, an unsymmetrical-cross with the horizontal search range equals W and vertical search range equals W/2 is used.
- Step 3: Uneven Multi-Hexagon-grid Search
  A full search with search range equals 2 is performed, as Fig.2 step3-1 indicates, followed by a multi-hexagon-grid search, as Fig.2 step3-2 shows.
- Step 4: Extended Hexagon based search (EHS)
  The large and small-sized hexagon search pattern is iteratively used until the current optimum matching point is located in the center, respectively.

A. Two improvements about UMHexagonS algorithm

From the above description about UMHexagonS algorithm, we can see it also has relatively high complexity in both computation and time although it has better performance than others. Here, we propose two improvements on the original UMHexagonS algorithm in order to further speed up the searching process.

Firstly, let's dig into the step 3 of UMHexagonS algorithm. For every video sequences, the original UMHexagonS algorithm has to evaluate all the 64 points covered by the four Sixteen Points Hexagon Pattern (16-HP), just as Fig.2 step 3-2 shows, which is very often useless and time-consuming because of the characteristic of center-biased motion vector distribution. Here, we introduce the EARLY_TERMINATION concept into step 3. When coming to step 3, the inner two 16-HP should be evaluated according to an evaluation metric and a current optimum point with minimum distortion can be located. Then we make a decision whether to continue or switch to step 4 according to the following conditional inequality.

$$cur\_opti\_poi\_cost * 0.8 < min\_cost$$

Where cur_opti_poi_cost represents the cost of the current optimum point, min_cost the minimal cost defined. If the inequality is true, then the outer two 16-HP searching is discarded immediately and switching to the step 4; otherwise, go on the remainder searching of step 3.

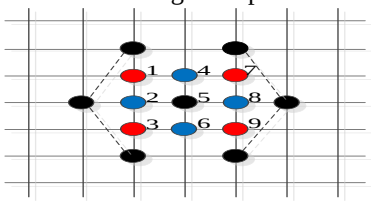

Figure 3. Inner points within hexagonal search pattern

Secondly, let's discuss the inner search within the large hexagon of step 4 of UMHexagonS algorithm, as Fig.3 shows. The original UMHexagonS algorithm has omitted points 1, 3, 7 and 9, while a full search would increase additional overhead. Given the monotonic distortion characteristic on the localized area around the global minimum, iterative small-sized hexagon search, which would decrease the overall performance due to added candidate points, is not necessarily in most cases. Based on the idea from [14], a new grouping method is proposed.

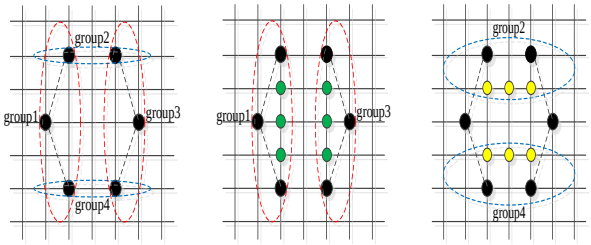

Figure 4. Grouping method and inner points distribution of hexagonal search pattern

The six checked endpoints of the large hexagon are divided into four groups, as Fig.4 shows. A group distortion is defined for each group by summing the distortions of all the points within the group. Note that when it comes to group 1 and 3, which three points within them, the group distortion value should be multiplied by 2/3, because only two points are within group 2 and 4, which is obvious from Fig.4. Three inner points closest to the group with the smallest group

distortion will be evaluated to find the global optimum point, i.e. the final solution of motion vector which points to the best matching block.

Here the IUMHexagonS algorithm is implemented using C programming language in H.264/AVC reference software 17.2. Three standard video sequences, "salesman", "Container" and "silent", are used for the test with search window size of ±15. The experimental result shows compared to the original UMHexagonS algorithm, the IUMHexagonS algorithm can reduce the computational complexity in motion estimation and give significant speed improvement rate in terms of search points per block while maintaining the almost same good video equality.

Moreover, the IUMHexagonS algorithm has been implemented in X264, which is used to encode video data on the serve side.

IV. THE DESCRIPTION OF SERVER AND CLIENT'S JOBS

A. *Video Capturing Module*

On the serve side, video data acquisition is implemented by use of Video Linux For Two(V4L2) interface, which is provided by the Linux kernel as API for user to develop video and audio applications. It involves many essential operations, including opening and closing device, setting format parameters, capturing and processing image information, on video device, such as USB camera in this paper. Vide devices belong to character device, corresponding to /dev/videox(x=0, 1 …) in Linux system.

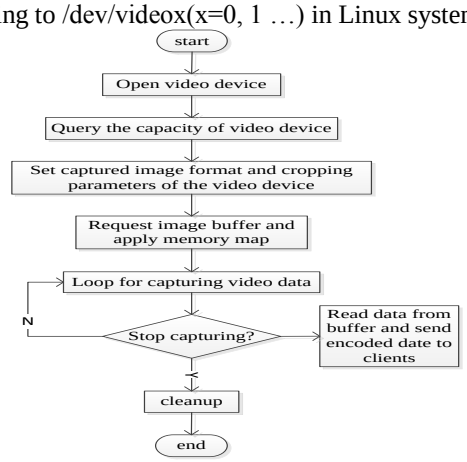

Figure 5. Video capturing flow diagram

The process of video data acquisition using V4L2 by the serve can be summarized as follows, as Fig.5 depicts.

- Opening the device using open("/dev/video0", O_RDWR), which would return a file descriptor on success; otherwise -1 on error with errno set to indicate the error.
- Querying capabilities of device by ioctl(fd, VIDIOC_QUERYCAP, &cap), which would return a structure v4l2_capability for judging whether the device supports capturing operations and streaming input and output.
- Enuming all the available captured image format through ioctl(fd, VIDIOC_ENUM_FMT, &fmtdesc), but the USB camera can only support format YUV422. Then, V4L2_PIX_FMT_YUYV should be set. What's more, the type of data stream is set to be V4L2_BUF_TYPE_VIDEO_CAPTURE with image width and height set to 320 and 240, respectively.
- Setting reqbuffers.count to the number of buffer used for storing captured image, and the type must be the same as stream type set last step. Also, other than reading and writing directly, memory map mechanism should be used in order to gain higher speed.
- Using VIDIOC_STREAMON and VIDIOC_STREAMOFF to start and stop image acquisition, and ioctl(fd, VIDIOC_QBUF, &type) should be used to put requested buffers into queue firstly. The captured data should be encoded using X264 of last section and transmitted to client through RTP.
- When the video capturing is done, don't forget to make necessary cleanup job, which can free unused system resources and mapped memory. Finally, closing the video device is essential.

B. *Wi-Fi Wireless Module*

Modern wireless communication technology has many different kinds of standards, including the 3rd generation (3G), Wi-Fi, Bluetooth, Zigbee, etc., each one with different protocols and speed rate.

TABLE I. WIRELESS COMMUNICATION TECHNOLOGY COMPARISON

type	3G	Wi-Fi	Bluetooth	Zigbee
protocol	WCDMA CDMA2000 TD-SCDMA	IEEE 802.11 a/b/g	IEEE 802.15.1	IEEE 802.15.4
maximum speed rate	2Mbps	11~54Mbps	1Mbps	10Mbps

From the Table 1, we can see that compared to other wireless technology, Wi-Fi has the maximum communication speed, support more protocols, easy to installation and setup, and has lower cost. Therefore, the system makes use of Wi-Fi wireless technology to assess to the Internet, for video data transmission. As the development board has already integrated the Wi-Fi module, a few easy steps, i.e., using command insmod to load the Wi-Fi module into kernel (Two essential files helper_sd.bin and sd8686.bin can be downloaded in Marvel official website.); using command ifconfig eth1 up to open network interface card and then command iwlist eth1 scanning to search all available wireless network; setting up the address and subnet mask of eth1; using command iwconfig eth1 essid wireless_network_name to set current network; through iwconfig eth1 key : youe_key to set your access key, should be taken to activate it.

C. *RTP Protocol Module*

The Real-time Transport Protocol (RTP) [15], which lies in the application layer of Internet protocol suite, defines a standardized packet format for delivering audio and video over IP networks. RTP is often used in conjunction with

the RTP Control Protocol (RTCP), which is used to aid synchronization of multiple streams and monitor transmission statistics and QoS. The well-known open source project, JRTPLIB [16], which is an object-oriented RTP library written in C++, is used in the video surveillance system for data transmission. Due to the maximum payload of 1500 bytes, the ceiling size of RTP packet is set to be 1400 bytes. Then, if the encoded video data is larger than 1400 bytes, it should be divided into many fractions with each less than 1400 bytes. The specific transmission process is depicted by the following picture.

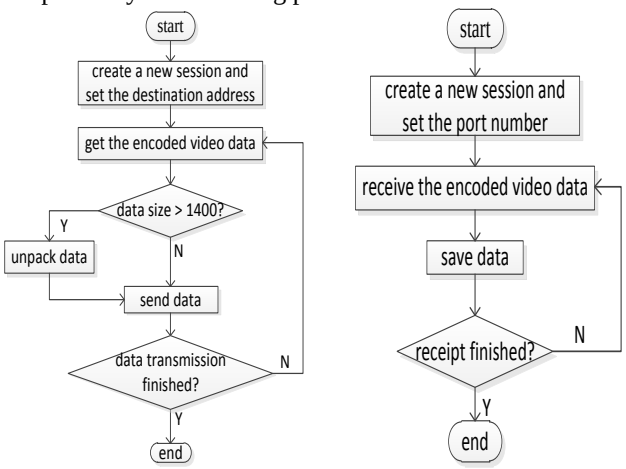

Figure 6. Transmission and receipt process diagrams

The details of transmission process can be described as follows, as Fig. 6 shows.

- Create a new RTP session and set the destination address. A RTP instance can be got by invoking method create(); then, call AddDestination() to acquire IP address and port number of the destination.
- Use Get_Data() to get video data.
- Send data via SendPacket().

The details of receipt process can be described as follows.

- Invoke create() to get a RTP instance and set the same port number as the transmission endpoint.
- Receive video data by invoking method PollData() of class RTPSession.
- Save the received data.
- Make a judgment about whether the data is finished, if no, switch to (2); otherwise exit.

D. Image Format Conversion And Display

When a client has already get video data from the server, the image format is yuv, which should be firstly converted into rgb in order to display. The conversion process can be easily done using the following formula.

$$\begin{bmatrix} B \\ G \\ R \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 2.018 \\ 1.164 & -0.813 & -0.391 \\ 1.164 & 1.596 & 0 \end{bmatrix} \begin{bmatrix} Y-16 \\ V-128 \\ U-128 \end{bmatrix} \quad (1)$$

The converted video data is displayed on the screen using GTK+ GUI library, which is a suite of cross-platform graphics development toolkit, and distributed under the GNU LGPL. GTK+ is an event driven toolkit, and the main theory is signal callback mechanism. Through g_signal_connect() to bind a specific signal with a specific function, the specific signal would be emitted immediately the specific event occurs. GTK+ can capture the signal quickly and invoke the binded function. The widget drawingarea and signal expose_event are used by the client to display video data received from the server. When a frame is successfully received, the client performs the format conversion by yuv_to_rgb() function, then calling gtk_widget_queue_draw() to let drawingare omit signal expose_event, which is captured by the main function and followed by gdk_draw_rgb_image() to display it.

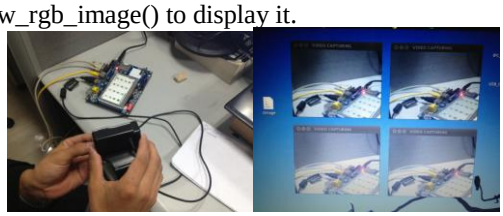

Figure 7. Experimental results

V. CONCLUSION

In this paper, a feasible scheme of embedded video surveillance system is proposed and implemented using ARM11 platform and H.264 compression standard. Two improvements on UMHexagonS algorithm are introduced to further enhance coding performance and implemented in X264 library used to encode video data in the system. Moreover, Wi-Fi wireless technology is used, making it more comfortable and convenient. The experimental result shows the overall system is stable and has great performance. This system can be used in many fields in the future, such as bank, road, school, bars, etc., having a broad application prospect.

ACKNOWLEDGMENT

This work was supported by: (1) Natural Science Foundation of Zhejiang province *(No. Y1090335)*; (2) National Science Foundation (*No. 61070114)*; (3) Natural Science Foundation of Zhejiang province *(No. LY13F010010)*; (4) Open Topic of Key Laboratory of Visual Media Intelligent Processing Technology Research of Zhejiang Province *(No. 2013050).*

REFERENCES

[1] Jun-Wei Gao, Ke-Bin Jia, "Embedded Video Surveillance System Based on H.264[C]", International Conference on Multimedia Information Networking and Security, Hubei, China, 2009, pp. 282-286.

[2] Ke Li, Ke-Bin Jia, Jing Xie, and Yan Wang, "Design and Optimization of H.264 Video Encoder on DSP Platform [C]", Second International Conference on Innovative Computing, Kumamoto, Japan, 2007, pp. 541-541.

[3] T. Koga, K. Iinuma, A. Hirano, Y. Lijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in

Proceedings Nat. Telecommunications Conf. 81, New Orleans, LA, pp.G5.3.1-G5.3.5, Nov. 1981.

[4] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation,"IEEE Trans. Circuits Syst. Video Technol.,vol. 4, pp. 438–442, Aug. 1994.

[5] J.Jain, A.Jain, "Displacement measurement and its application in interframe image coding," IEEE Transactions on Communications, vol.COM-29, pp.1799-1806, Dec. 1981.

[6] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation,"IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313–317, June 1996.

S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," IEEE Trans. Image Processing, vol. 9, pp.287–290, Feb. 2000.

[7] Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Patten for Fast Block Motion Estimation", IEEE Trans. on CSVT, pp.349-355, Vol.12, No.5, May, 2002.

[8] L. K. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding,"IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 419–423, Aug. 1996.

[9] A. M. Tourapis, O. C. Au, and M. L. Liou,"Predictive motion vector field adaptive search technique (PMVFAST) enhancing block-based mo-tion estimation,"inProc. SPIE Conf. Visual Communication and Image Processing, Jan. 2001, pp. 883–892.

[10] Z. Chen, Y.He, and J. Xu, "Hybrid unsymmetrical cross multihexagon-grid search strategy for integer pel motion estimation in H.264," in Proc.PCS, pp. 17-22, Apr. 2003.

[11] Zhibo Chen, Peng Zhou, Yun He, "Fast Integer Pel and Fractional Pel Motion estimation in for JVT", JVT-F017r1.doc, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 6th meeting, Awaji, Island, JP, 5-13 December, 2002

[12] Z.B. Chen, P. Zhou, and Y. He, "Fast Motion Estimation for JVT", JVT-G016, Thailand, March 2003.

[13] Zhu C. Lin X. Chau L. PO L.M. Enhanced hexagonal search for fast block motion estimation [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2004, 14(10): 1210-1214.

[14] http://en.wikipedia.org/wiki/Real-time_Transport_Protocol

[15] http://research.edm.uhasselt.be/~jori/page/index.php?n=CS.Jrtplib