

HW-SW Framework for Multimedia Applications on MPSoC: Practice and Experience

Liu Yan

School of Computer and Communication, Hunan University, Changsha, P.R.China
Email: qmttz@163.com

Li Renfa , Xu Cheng and Yu Fei

School of Computer and Communication, Hunan University, Changsha, P.R.China
Email: lirenfa@vip.sina.com, cheng_xu@yeah.net, hunanyufei@126.com

Abstract—Constructing an intelligence surveillance system using embedded video server requires a sophisticated hardware/software framework for this system, and it should consider the performance, cost and energy/power consumption constrains. This paper discusses the design and implementation of an intelligence surveillance system which uses embedded multimedia server as core computing platform. We have carefully designed the embedded server based on MPSoC platform for the surveillance system whose key functionalities include heterogeneous multiprocessors environment set up, key algorithm porting and development framework design. The heterogeneous multiprocessors environment setup provides memory/communication management among multiprocessors to enhance the programmability of platform. Using realize/optimize the video encoding algorithm and porting to the TMS320DM6446 chip to guarantee the availability of media streaming system based on network. The hardware/software development framework provides an adequate mechanism which models signal processing in video surveillance system as process network and provides integrated homogeneous abstract programming environment. The proposed system has been implemented on an MPSoC platform and it is fully operational. The design principles of the whole system are validated with real experiments, and the performance characteristics are analyzed.

Index Terms—surveillance system, embedded system, H.264

I. INTRODUCTION

Most surveillance system currently in use or under construction use analog signal as their primary transfer mode. Modern video surveillance system using multimedia network, human intelligence and computer vision technology needs to offer a broad range of functions at low cost with low energy consumption. However, the digital surveillance system usually requires high-speed broadband wireless network, custom multimedia servers and high definition video for the user's demands. Therefore the high speed and low cost is in constant demand as surveillance systems are

widespread. One approach to support the intelligent surveillance system ability requirement is to use embedded multimedia server as the system architecture's core device. The embedded multimedia server typically consists of hardware system and software system and hard to develop. It has a flexible system configuration and low cost, but the demand of computing ability increases significantly due to the complex video processing algorithm. To obtain high definition video in the system, the embedded servers usually select Multiprocessor System-on-Chip as core process unit. MPSoC are often heterogeneous multiprocessors, made up of several types of processors. Multiprocessing is very common in embedded computing systems because it allows us to meet our performance, cost, and energy/power consumption goals [1].

Rapidly development a digital network intelligent surveillance system poses many technical challenges [2]. First, the handling of multimedia contents using network should employ the powerful computing platform consists of DRAM, CPU, and DSP unit and so on. It is difficult to program on the platform due to the mixture architecture. Therefore, configuration of tool chain in Ti DavinCi system is needed, and software architecture must be carefully design and allocate to ARM or DSP subsystem separately. In this paper, we concentrate on the software modular allocation and communication realization details in heterogeneous MPSoC. Secondly, the video encode algorithm should be redesigned and optimized effectively, and port to the TMS320DM6446 board designed by ourselves. H.264/AVC algorithm is encapsulated by xDM standard to running on the Ti DavinCi architecture. Furthermore, motion search as the component with highest computational complexity in H.264, has a significant impact on the performance of the entire algorithm. An adaptive fast motion estimation algorithm was presented in this paper to improve the efficiency of the search process. Thirdly, increase of the complexity in surveillance systems makes it mandatory the need of new approaches to help the designer to manage such system, and the election of the appropriate design flow is crucial to achieve a satisfactory result. It is very important to

Corresponding author: Liu Yan.

provide an integrated development environment supporting rapid development and complexity component management. The criterion of a new IDE should be supporting shared memory/message passing mechanism, and uniform on-line abstract evaluate framework.

Research results are found in the literature regarding the use of multiprocessor platform device for a network multimedia system computing device. There are so many platforms have been designed. The Philips Nexperia [3] is a multiprocessor system-on-chip designed for digital video and television applications. ST Nomadik is designed for mobile multimedia systems such as PDA, and car entertainment system. The Texas Instruments OMAP family of multiprocessors was designed for mobile multimedia applications. Although these platforms provide powerful computing resources, they do not provide comfortable and portable development environment. Motion Estimation (ME) is a core part of most modern video coding standard, and it directly affects the compression efficiency and visual quality of a video. Many fast algorithms have been proposed to reduce the number of search points in ME, such as TSS [4], NTSS [5], PMVFAST and MVFAST [6]. However, the both of these algorithms did not consider the change of the applications environment such as surveillance system. Video surveillance networks are a subclass of sensor network, but there are significant differences in the constraints to which they are subject, and hence in the required middleware support. This middleware provides support for both computational and communication aspects of automated video surveillance networks. Future work will include further refinement of the middleware and its underlying architecture, supporting changes in the surveillance network software so as to accommodate changing requirements.

Most of these works have studied the use of embedded computing platform as a media server of surveillance system. The existing works only consider some of above issues, and does not look at them in a system-wide perspective. In this paper, we study and discuss the technical issues on developing digital surveillance system based on MPSoC platform. We discuss the system architecture, key algorithm optimization, and supporting software environment. The proposed system has been implemented on a TMS320DM6446 platform, and its performance is analyzed. The paper is structured as follows. Section 2 describes the system structure and supporting software. Section 3 and section 4 detail the key algorithm optimization and porting, development environment, respectively. Section 5 discusses the experimental results. Section 6 concludes the paper.

II. SYSTEM ARCHITECTURE

A. Surveillance system

The main task of proposed system is to capture the real-time audio/video data, then to compress and transmit it, and store it to local disk if necessary. The architecture of the proposed real-time intelligent video surveillance

system is shown in Figure 1. The multimedia data can be stored in the local disk of embedded video server or other media server. Multiple cameras are positioned at different place to monitor object area. The embedded video server provides the capabilities of compression, storage and transmission of multimedia data, and the control of other devices. The management server accepts the requests from the PC monitor or embedded device and designates them to the special video channel. The key parts of our proposed system are embedded video server and software framework designed for real-time multimedia.

B. Software layer

The software system consists of three parts: one for the embedded system, one for the management server and the other for the client. The most important part is

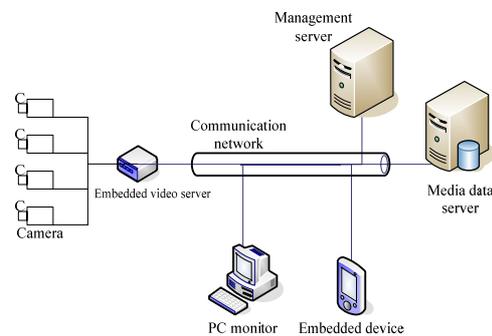


Figure 1. Architecture of surveillance system.

software system running onto the embedded video server based on Ti DavinCi technology, showed in Figure 2. Linux OS was ported to dual-core architecture, provides control-intensive tasks such as TCP/IP application, device driver, et al. The computation-intensive task such as H.264 algorithm was running onto DSP core. Code Engine and DSP Link provide interactive channel between ARM and DSP. DSP/BIOS is a small core

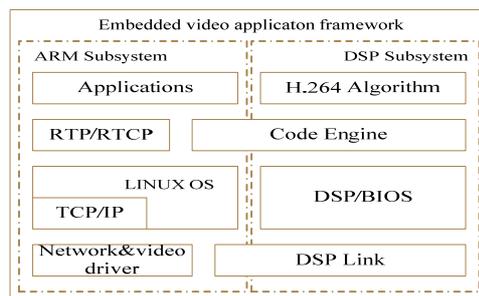


Figure 2. Software of system.

operating system running on the DSP core to provide basic operations. The management server software provides the main form of user interaction with the proposed system such as transmitting control, authentication reorganization.

C. Communication between ARM and DSP

The processing elements in a multiprocessor perform are the basic computations of the whole system. TMS320DM6446 includes a RISC processor - an ARM9 - along with a DSP - a Ti C55x. A shared memory

interface allows the two processors to communicate

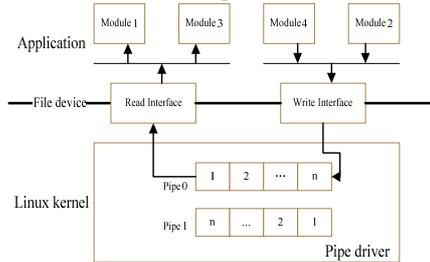


Figure 3. Module Link diagram.

efficiently. Furthermore, we have carefully designed Module Link to support transfer data/control signal among subprogram running on ARM processors, showed in Figure 3. Communication services are constructed by the low layer named pipes of OS. The drivers of pipes, which had been compiled into kernel modules, could be loaded automatically by the system or manually by the administrator. And all these drivers afford standardized read/write interfaces to applications. Inside the driver service, there are multi-pipes. Each of them are mapping with a memory segment allocated by the kernel. Applications can distinguish the pipes from the device file names, that is to say, the only thing should be done in the application layer is that make sure which devices should be used for the data reading and writing. And the driver service also secures the data reading by its P, V operations. A read port and a write port makes up of the two terminals of a pipe. The read-write ports are shared for several process modules. The maintenance of a waiting queue in the driver service allows application layer to inquire about the data condition by the system call - select. The processes will be suspended and the control of CPU will be abandoned if there is no data available. And all these behaviors will bring on the system efficiency.

D. Memory allocation design

The multimedia stream used in surveillance system is usually multi-channel video/audio signal. In our project,

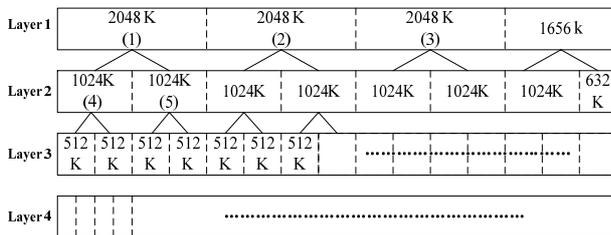


Figure 4. layer of virtual memory.

4-channel CIF format (352*288 pixels) video signal must encode or decode immediately. So we have to allocate four 202752 byte memories for data buffering (UYVY format), and Ti DavinCi platform can not support the requirements. Furthermore, the memory manage mechanism still has several defects. In general Static memory management used in it led to many memory fragments that are unused and the memory utilization too lower. Additionally, there is no mechanism to support memory allocation self checking. In other words, if there are not suitable area for allocation, system would collapse.

Buddy algorithm was adapted to manage dynamic memory allocation in our system. We have carefully designed layered dynamic virtual memory pool, showed in Figure 4.

III. ALGORITHM PORTING AND OPTIMIZATION

A. Algorithm package using xDM

The range of bit rates and picture sizes supported by H.264/AVC is correspondingly broad, addressing video coding capabilities ranging from mobile and dial-up devices, through to HDTV, and beyond. H.264/AVC standard is suitable for surveillance application system.

In our project, to running H.264 on the TMS320DM6446 platform, the algorithm interface must be standardized by xDM. As a result, every xDM-compliant algorithm is created and deleted with exactly the same function calls. This means that all xDM-compliant algorithms can use the same skeleton function for create and delete. Furthermore, some design details should be considered. The first is data coherency problem. The ARM processor contains a memory management unit (MMU) which is used to translate physical memory addresses into virtual addresses. The DSP core on the chip does not have an MMU. There is a driver module that is used to allocate physically contiguous buffers on the ARM and buffers that will be used by the DSP need to be allocated using this module; the second memory issue is cache coherency. Both the ARM and DSP cores use cache to improve the efficiency of using external memory. However, the ARM and DSP cores are both unaware of reads and writes performed by the other core, so the programmer must be aware of it. Following is packaged H.264 codec function.

```

VIDENC_create(Engine_Handle, String ,IVIDENC_Params *)
VIDENC_control(VIDENC_Handle,VIDENC_Cmd,VIDENC_Dyna
micParams *param, VIDENC_Status *)
VIDENC_process(VIDENC_Handle , XDM_BufDesc
*,XDM_BufDesc *, IVIDENC_InArgs *, IVIDENC_OutArgs *)
Void VIDENC_delete(VIDENC_Handle)
    
```

B. Adaptive motion estimation algorithm

H.264 algorithm codes have been analyzed carefully using CCS profile tools to guide us optimize goal in the

TABLE I. PROFILING RESULTL

Modular	Computing Time(%)
Motion Search	35 – 45
DCT	20 – 25
Macroblock Rebuild	5 – 10
Entropy encoding	5 – 10
Deblock & Interpolation	10 - 20

future. The profile result of our program is showed in Table I.

The CCS compiler can optimize program in C code level by several parameter options to improve circulation efficiency. The assemble level optimization used in the

paper is a linear assemble model. This means that program should be recoded according VLIW and pipeline architecture. The programmer can ignore some information about register used by instructions, and parallelism of instructions. Furthermore, we can use some special instructions to further improve code efficiency provide by C64x serials DSP such as AVGU4, MIN2, MAX2, SPACKU4, PACK2, etc.

Motion estimation is an important part for the computing intensive H.264 algorithm. The motion estimation is normally computing the motion vector, and decreases the timing redundancy. This section describes the design principles of proposed adaptive motion estimation algorithm. The system reduces the motion estimation time by following three steps.

Step 1, pre-judgment of the skipped macroblock. A macroblock for which no data is coded other than an indication that the macroblock is to be decoded as "skipped". In the fact, there are lots of full-zero blocks in the skipped macroblock after DCT transfer. Many skipped macroblocks can be found in the video frame in the surveillance applications due to the object do not move frequently. So, if we can pre-judgment of the skipped macroblock, the Mode Select, Motion Predication, and DCT can be ignored to improve codec efficient visibly. Following is the judgmental formulation of the skipped macroblock.

$$\sum_{x=0}^3 \sum_{y=0}^3 |f(x, y)| \leq \frac{\sqrt{2}(2^{q_{bit}} - f)}{2 * QE[q_{rem}][0][0]}$$

Step 2, select multi reference frame prediction center. An efficient way of decrease computing time of Motion Estimation is to find the most valuable frame prediction

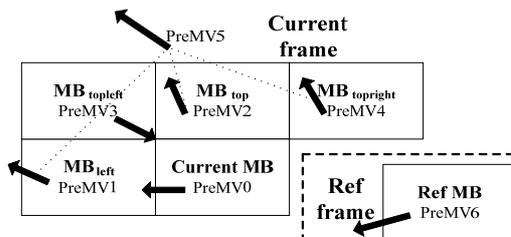


Figure 5. Prediction vector diagram.

center to start search procedure. There are seven motion prediction vector for current macroblock in our system, named PreMV0~PreMV6, showed in Figure 5. PreMV0 is equal to current MB central MV value. PreMV1, PreMV2, PreMV3, and PreMV4 are equal to the MV value of left, top, topleft, and topright direction MB respectively. PreMV5 is median {PreMV1, PreMV2, PreMV3, PreMV4}, and PreMV6 is reference frame MV value. After computing SAD of above, the minimums value one is selected as prediction center.

Finally, system will select appropriate motion search process according to the SAD value of frame.

IV. DEVELOPMENT ENVIRONMENT FRAMEWORK

A. Development framework for surveillance system

The digital surveillance system is the design of an embedded computer system integrated computer vision, digital video processing, artificial intelligence, and control theory. Design efforts on software part are getting much heavier than on hardware part as the system becoming more and more complicated. In this paper, the surveillance system is described by device topologic diagram, and algorithm diagram. The first one describes devices connective relationship, and the dataflow or interface of key algorithm are presented in the other one. The development framework proposed in this paper

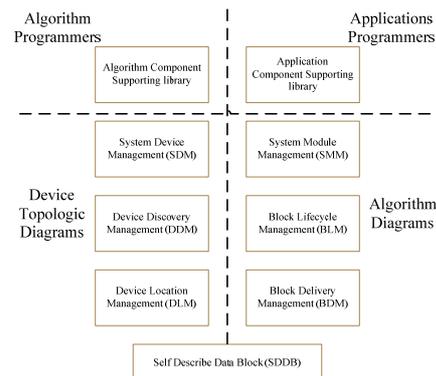


Figure 6. Development framework for surveillance.

provides algorithm interface standards, and application coding standards or guidelines to improve system performance and robustness, and to accelerate the development process. Figure 6 shows the block diagram of the open framework for digital video surveillance system which is a message-oriented architecture. The self description data block (SDDB) is core part of the whole architecture, providing message passing services for the components used. It consists of block information such as data type, destination address, and data package. The block lifecycle management (BLM) monitors the procedure of create/process/transfer, and allocates memory used by SDDB instance. The block delivery management (BDM) maintains the position details of each component. The supporting library which is key part of whole development framework implements the public code of framework services, so application programmer can pay more attention to the top-level software's development.

B. Basic module and IDE

The IDE of digital surveillance system is tool chains for system rapid develop, consist of graphic processing flow constructor, code auto generator, application services wizard, and code parser for modules. Users can describe the surveillance system behavior graphically, and can generate mainly framework code automatically by using the graphic processing flow constructor and code auto generator; The application services wizard can scan the code, and then generate others files needed by framework. Above IDE of digital surveillance system is realized by Eclipse platform showed in Figure 7.

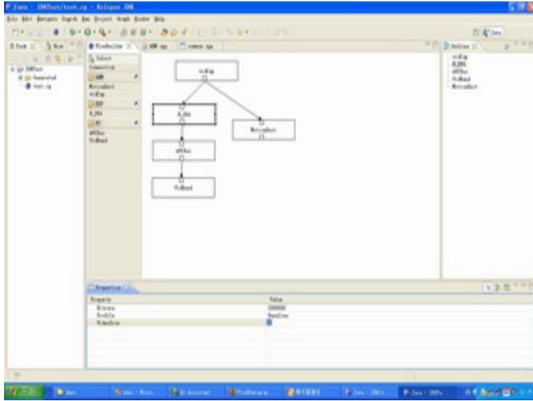


Figure 7. IDE based on Eclipse tools.

V. EXPERIMENTS AND RESULTS

The intelligent surveillance system has been implemented, and the embedded multimedia server has been realized on the TMS320DM6446 platform. The whole system has been evaluated on real-world surveillance scenes and consistent good results have been observed, in terms of high accuracy and robustness to noise influences.

A. Experimental setup

The client software system mainly written in C++ runs on a 2.4GHz PC. The embedded server has been developed on a Linux system based on self design Ti TMS320DM6446 platform, equipped with some video cameras. The camera installed in object area captures the video data, and compressed in H.264 format in embedded video server board. Then the H.264 video data can be accessed by client using wireless network or Ethernet, or stored in the local disk of embedded video server. The

TABLE II. KEY FUNCTIONS CYCLES

Names	Un optimize	C code optimize	Assembly optimize
Sad16x16	4197	190	84
Sad8x8	1260	60	35
Dct4x4	969	66	32
Idct4x4	993	70	36
Quant4x4	870	75	37

TABLE III. VIDEO TESTING RESULTS

CIF video	rate (frame/sec)	bitrate (kbps)	PSNR (db)	DSP use (%)
Coastguard.cif	24.7	300.6	32.03	36
Foreman.cif	25.2	299.9	35.22	35
News.cif	27.5	300.2	40.44	34
Akyio.cif	27.0	298.1	45.08	35

key objective of the experiments is to evaluate the performance characteristics of the implemented server experimentally. The first experiment is to analyze the algorithm optimization results in our platform. The

second experiment analyzes the proposed adaptive motion estimation algorithm in our real application environment. And the last experiment shows our basic development flow, and application scenario.

B. Algorithm optimization results

This experiment analyzes the optimization results of H.264 algorithm when it has ported to DM6446 chip. In the algorithm optimization process, we may use several methods to improve H.264 algorithm running speed, such as C code level optimization, linear assembly optimization, and instruction level optimization. Performances of several key functions in the H.264 have been evaluated.

TABLE IV. PERFORMANCE COMPARE USING PRE-JUDGMENT METHOD

CIF video	Org-PSNR (db)	PSNR (db)	Org-rate (f/s)	rate (f/s)
Coastguard.cif	31.29	31.21	14.81	16.02
Foreman.cif	34.86	34.73	14.72	16.21
News.cif	40.16	40.02	15.44	18.04
Akyio.cif	44.91	44.81	15.85	17.66

Table II shows the CPU cycles of key functions of H.264 after C code and assembly optimize process. Obviously, un-optimize code is very inefficiently. For example, function sad16x16 obtains more than 95% improvement after using C code level optimization. When we use assembly level techniques, the CPU cycles of sad16x16 can further improve about 50%. Table III shows results of the real CIF format testing sequence in our real system. Because computing platform used in embedded server is heterogeneous dual-core architecture, we can learn utilization of DSP core in our system from Table III column 5.

C. Adaptive motion estimation evaluate

This experiment analyzes performance of proposed

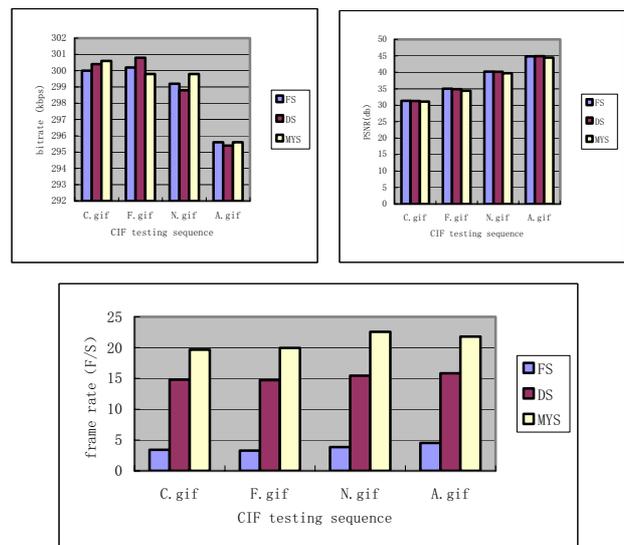


Figure 8. Algorithm evaluation results.

adaptive motion estimation algorithm. There are four

testing video sequence, such as coastguard.cif, foreman.cif, news.cif and akyio.cif, the same as above experiment. Table VI gives the results of testing sequence after using pre-judgment skipped macroblock method. It is apparent that for these testing sequences, the PSNR is decreased, and the frame rate of video is improved about 2%.

To evaluate proposed adaptive motion estimation algorithm (MYS), it is compare with fully search algorithm (FS) and diamond search algorithm (DS). Figure 8 shows the evaluation results. There are three evaluation indexes for algorithm compare--PSNR, and bit rate, and frame rate. See from experimental results of Figure 8, we can know that, after adopting the improved adaptive estimation algorithm, compared with diamond searching algorithm, the signal-to-noise ratio of image peak value has a bit coming down, less than 0.5db, but above 63% of calculate time has been saved, and at the same time, the frame rate (CIF series) has been advanced 5-7 F/S, which has made a great advance on the road leading to real time code. Simultaneously, compared with full searching algorithm, the adaptive estimation algorithm has saved 99% of the calculate time, and the advancements on other performances are more obviously.

D. System integration

The system has been evaluated on real-world surveillance scenes and consistent good results have been observed, in terms of high accuracy and robustness to noise influences. We can obtain 30 fps in CIF format and 25 fps in D1 format, which satisfies the demand of commercial intelligent surveillance system applications. The video coding rate is lower than 300 kbps, so the compressed video data can be transmitted by communication data network such as ADSL.

Figure 9 shows the user screen of application software on PC. In fact we can set one channel to nine channels by software. There is some alarm zones connected to the object areas on the bottom of the GUI. In the right of screen, some attributes of system can be set on this panel including video channel, network delay tolerance, pan tilt zoom camera control, and other functions. The camera installed in object area captures the video data, and



Figure 9. 4-channel video surveillance system.

compressed in H.264 format in embedded video server board. Then the H.264 video data can be accessed by client using wireless network or Ethernet, or stored in the local disk of embedded video server. There are some intelligent functions such as motion tracking and alarm,

face recognition can also be run based on our system. During the process of testing, all functions mentioned in system design are usable and stable.

VI. CONCLUSIONS

This paper has presented the design and implementation of a digital surveillance system, which use heterogeneous multiprocessor system-on-chip as the core computing platform of embedded multimedia server. We have proposed an effective development framework based MPSoC for rapidly develop intelligent surveillance system. Some realization details about framework are presented including system architecture, and key software supporting middleware. Memory allocation technique presented here is very important for dual-core communication in Ti DaVinci platform. To obtain efficiently H.264 algorithm, some optimization skills have been used in our project, such as adaptive motion estimation, C code and linear assembly optimize. Finally, a real surveillance system was built using our development framework, and its performance was analyzed.

The experience discussed in this paper is based on a real implementation of the system. Developing a intelligent surveillance system requires sophisticated system design principles, software architecture and implementation techniques. Our experience could be meaningful to practitioners who are considering developing a similar system. There are several interesting issues to be studied further including program model for heterogeneous multiprocessor system-on-chip, intelligent motion detect and track algorithm, and some architecture level research topics. There are also parts of our future research.

ACKNOWLEDGMENT

This paper is supported in part by the National Nature Science Foundation of China under Grant 60673061 and Hi-Tech Research and Development Program of China under Grant 2007AA01Z104.

REFERENCES

- [1] W. Wolf, *High-Performance Embedded Computing*, 1st ed., Elsevier: Singapore, 2007, pp.267-268.
- [2] M. Valera Espina and S. A. Velastin, "Intelligent distributed surveillance systems: A review," *IEE Proceedings - Vision, Image and Signal Processing*, vol.2, pp. 192-204, April 2005.
- [3] Santanu Dutta, Rune Jensen, and Alf Riechmann, "Viper: a multiprocessor SOC for advanced set-top box and digital TV systems," *IEEE Design & Test of Computers*, vol.18, pp. 21-31, Sept 2001.
- [4] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *Proc. Nat. Telecommun. Conf.*, New Orleans, LA, pp. G5.3.1-G5.3.5, Dec 1981.
- [5] R. Li, B. Zeng, and M.L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438-42, Aug 1994.

- [6] K.K. Ma and P.I. Hosur, "Performance Report of Motion Vector Field Adaptive Search Technique (MVFAST)," in *ISO/IEC JTC1/SC29/WG11 MPEG99/m5851*, Noordwijkerhout, NL, Mar 2000.

Liu Yan received the B.S. and M.S. degree in computer science from Hunan University in P.R.China 2002, 2005 respectively. He is currently working toward the Ph.D. degree at the Hunan University. His current research interests are embedded system design, and computer architecture.

Li Renfa was born in 1956. He received B.S. degree and M.S degree from Tianjin University in P. R. China in 1982 and

1985 respectively. He received PhD degree in Mechanical-electronic Engineering from Huazhong University of Science and Technology in P. R. China.

Currently he is professor in College of Computer and Communication, Hunan University, P. R. China. His current research interests include embedded system, computer architecture and wireless sensor network.

Xu Cheng was born in 1962. He received B.S. degree and M.S degree from Hunan University in P. R. China in 1983 and 1986 respectively. He received PhD degree in Mechanical Manufacture and Automation Wuhan University of Technology in P. R. China in 2006.

Currently he is professor in College of Computer and Communication, Hunan University, P. R. China. His current research interests include embedded system, network multimedia applications and control theory.