H.264-BASED WIRELESS SURVEILLANCE SENSORS IN APPLICATION TO TARGET IDENTIFICATION AND TRACKING

By

WEI ZHAO * JEFFREY FAN ** ASAD DAVARI ***

*,** Department of Electrical and Computer Engineering, Florida International University, Miami, FL, USA.
*** Department of Electrical and Computer Engineering, West Virginia University Institute of Technology, WV, USA

ABSTRACT

In this paper, the author propose a hardware-based micro structure inside the architecture of H.264 encoder, called Vector Bank (VB), which can dramatically reduce the bandwidth, memory, and computation time of a multi-camera video surveillance system. VB is a memory-based structure that contains the vector information for every single Macro Block (MB) of the video frame from the Motion Estimation module of the H.264 encoder. After extraction of the several vectors from several frames, a dedicated Digital Signal Processor (DSP) can be assigned to analyze and predict the trajectories of the objects in motion. With the application of 2-D edge detectors, such as Sobel or Laplacian of Gaussian (LoG) operators, the objects in motion can be isolated from the background and thus be easily identified. In addition, a middleware-based technique derived from Directional Discrete Cosine Transform (DDCT) is introduced to improve the image quality without sacrificing the system performance. We propose four new modes of operations in DDCT for better image compression ratio. The experimental results show that we can improve the image quality of the targets, isolate them from the background, and track them easily in both linear and exponential trajectories of the motion.

Keywords: H.264, Vector Bank, Surveillance Network, Target Tracking.

INTRODUCTION

People have been using visual sensors to setup surveillance systems for more than ten years. In the beginning, most were analog, and finally became digital, which are better with noise control and signal processing. However, just because we have digital visual sensors does not mean that we have a complete digital visual surveillance system. In this paper, the author propose a hardware-based digital surveillance system. It needs several related but separate technologies to make our proposed structure feasible. In the next few subsections, the author discusses these technologies, followed by the discussion of the analog video surveillance system in comparison to the digital platform that the author propose in this paper.

1. Analog Video Surveillance System

1.1 Digital Visual Surveillance System

People use Visual Surveillance Systems as human eyes to predict and prevent threats. The Digital Visual Surveillance System in this paper consists of four parts: digital visual sensors, digital video codec chips, digital video data network, and digital video playback/storage system.

1.2 H.264 Video Codec Core

The first technology that we need to talk about is the video codec. Without video codec, a typical 480p (pixel) raw color video (about TV size) at 30fps (frames per second) will need a transfer bandwidth or storage rate around 28MB/s, which will be about 100 times larger in comparison to the H.264 version.

$$(640x480)_{pixel/frame} X3_{Byte/pixel} X30_{frame/sec} = 27.648 MB/s \quad (1)$$

Figure1 shows the module connection of the H.264 codec core [6]. The current frame (F_n) and reference

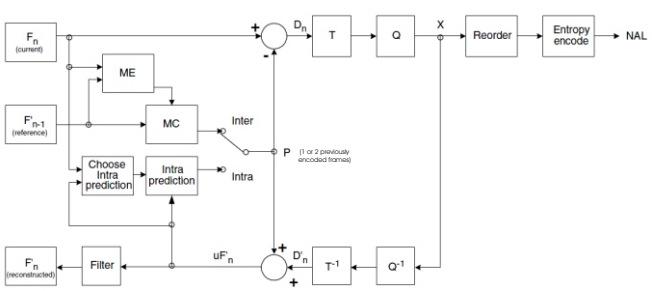

Figure 1. H.264 Codec Core [6]

frame(s) (F_{n-1}) are all going to be processed in a very important module called Motion Estimation (ME). In analysis of the relationship between the current frame and reference frame(s), the ME module can eliminate the time redundancy of current frames and with the help of the Motion Compensation (MC) module, the RGB signal can be changed into 2-D signals, i.e. Motion Vectors (MV) and Residues. Afterwards, the Transformation module (T) uses Discrete Cosine Transform to reorder the frequency response of MVs and Residues, followed by the quantization (Q) module, which intends to remove the high frequency component to eliminate the frequency redundancy of the MVs and Residues.

1.3 H.264/AVC System-on-a-Chip (SoC) Design

According to the instruction profiling of HDTV1024P (High Definition TV with resolution of 2048×1024, 30fps) specification [2], the H.264/AVC decoding process requires 83 Giga-Instructions Per Second (GIPS) of computation and 70 Giga-Bytes Per Second (GBPS) memory access rates. In H.264/AVC encoder, up to 3,600 GIPS and 5,570 GBPS are required according to HDTV720P (1280 × 720, 30fps) specification [2].

Apparently, although many excellent works on H.264 integer motion estimation schemes have been proposed [3–5], there still exists a wide gap in development for software to match up with the efficiency. The software approach is always the better solution (in consideration of the cost), in particular when ideally there is no computing time limitation. However, realistically there exist applications with time constraints, such as real time processing. Just like our case, a visual surveillance sensor network needs to respond in real time, and can only achieve this based on the hardware solution.

Figure 2 is a sample demonstration of H.264 based System-on-a-Chip (SoC) architecture. It contains one CPU as the management module of the entire system, one arbiter as the management module of the data/control bus, one Direct Memory Access Controller (DMAC), one main memory, one H.264 codec core, and several I/O peripheral device interfaces such as the sensor interface and USB/Ethernet/Wi-Fi interface. The data flow is

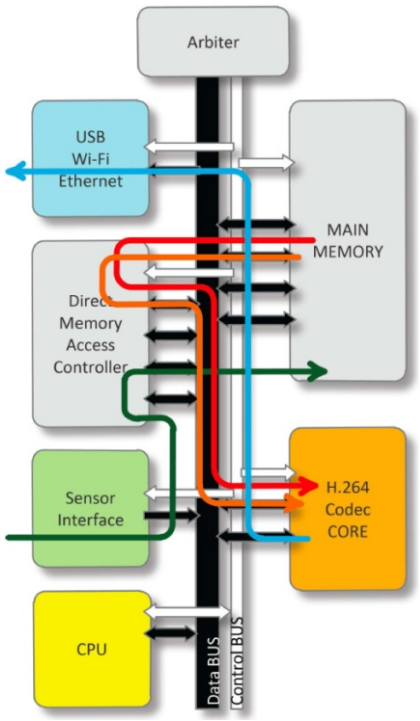

Figure 2. General H.264 Based System-on-a-Chip (SoC) Design

described below:

1) Starts from the green line which inputs the raw video stream data into our main memory through DMAC.
2) Then it turns into a red line and orange line, which represents the "current frame" and "reference frame" data respectively, and goes through the DMAC again to the H.264 codec core.
3) Finally, the blue line which is compressed data, gets released from the H.264 and travels along the bus to the USB/Wi-Fi/Ethernet system output interface and out puts from the system.

For a typical industry model, generally it consists of a Power Management Unit (PMU) and possibly several more I/O interfaces such as a LCD output interface, keyboard input interface, and so on. As for our research purpose, we use this simplified SoC model to build our system.

1.4 Motion Detection and Edge Detection

Motion object detection and tracking techniques have been studied for years and served in many areas of interest such as authentication systems, machine-human interfaces, and the most common, video surveillance. There are a number of different techniques used in

motion tracking fields today [7-10]. Most utilize the frame differences to detect object movement.

Edge Detection [12-14] is another very popular and important technology in Computer Vision. It is well developed and widely used in Digital Image Processing (DSP). Edge Detection also has many members of algorithms which can be categorized as first-derivative operators and second-derivative operators. First-derivative operators, such as Roberts, Prewitt, and Sobel, can detect the edge of an image in one dimension (either horizontal or vertical), while second-derivative operators, such as the Laplacian operator can detect in both dimensions at the same time [11]. In this paper, the author uses the two most famous, which are "Sobel" and "Laplacian of Gaussian" (LoG) operators.

1.5 Analog Video Surveillance System Structure

Figure 3 shows a typical analog video surveillance system. Every single camera (no matter digital or analog sensors) needs to use an independent channel to send the captured video data back to the "central room". The central room must have the ability to playback and store all of the transmitted data. There must be a security guard (some times more than one) who needs to monitor the screen to perceive any potential threats or unwanted activities.

In an analog vision system, the biggest constraints are the bandwidth of all the channels and the processing ability of the central server. Simply changing the video data to digital (by using digital sensors) cannot solve these problems completely. Digital raw data requires much more bandwidth than analog, but if data is compressed by on-camera video coding chips, it will dramatically increase the overhead of the server with high computing power in the central room. By adding a new hardware element, this paper proposes a video surveillance system that is much more efficient and secure.

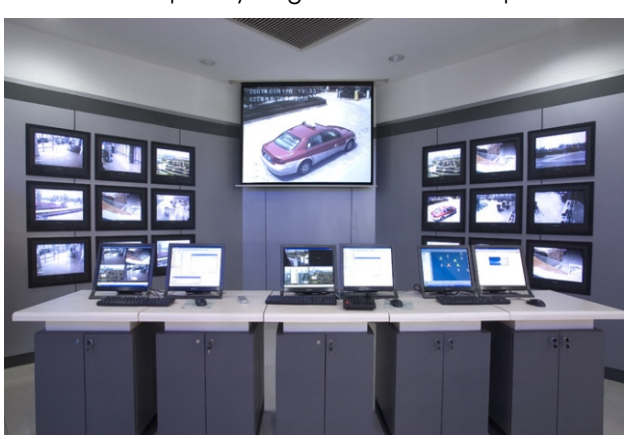

Figure 3. H3C Vision Surveillance System [25]

The rest of the paper is organized as follows. Directional DCT (DDCT) implementation based on H.264 core architecture will be introduced in Section 2. Vector Bank structure and implementation will be illustrated in Section 3. Edge Detection operations for Motion Detection will be reviewed in Section 4, and finally, a digital video surveillance system in comparison to an analog system will be discussed in Section 5.

2 Directional DCT Implementation

2.1 Directional DCT (DDCT) introduction

Many image and video coding algorithms have been developed over the past 30 years, such as sub-band/wavelet coding [15, 16], transform coding [17], vector quantization [18], and predictive coding [19, 20]. Among these coding technologies, the block-based transform approach has been recognized as the most successful, both for videos and images.

The Directional Discrete Cosine Transform (DDCT) [21] is a modified 2-Dimension Discrete Cosine Transform (DCT) [22, 23] technique to optimize the block based transform platform.

2.2 DDCT Algorithm

The DDCT is a block-based transform algorithm which is based on 2-D DCT image processing. In general, by applying the 2-D DCT to an image block, a frequency distribution map can be generated which contains both the low frequency component to be kept and the high frequency component to be eliminated, but only in the horizontal and vertical directions. By adding several different modes, the original 8x8 pixel block could be processed with 2-D DCT in several different directions.

Figure 4 shows an 8x8 pixel block. Originally, 2-D DCT will process DCT twice, horizontally and vertically but in this case, as you can see in these figures, the first DCT direction is not horizontal or vertical. This figure shows you

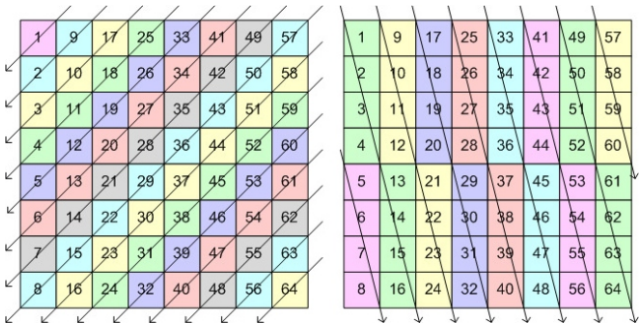

Figure 4. Mode 2 and Mode 5 of Directional DCT

mode 2 and mode 5 of DDCT. As you might have already noticed in this figure, the length (pixel counts) for each DCT is not the same.

DCT outputs the DC value and low frequency values in the front, followed by the high frequency values. For example, in mode 2, after the first DCT process, the 64 pixel block changes as shown in Figure 5.

In Figure 5, the darker side represents the low frequency component of the first DCT results of mode 2. The second DCT should process all of the resulting data in a perpendicular direction, shown in Figure 6 as the final result of DDCT mode 2.

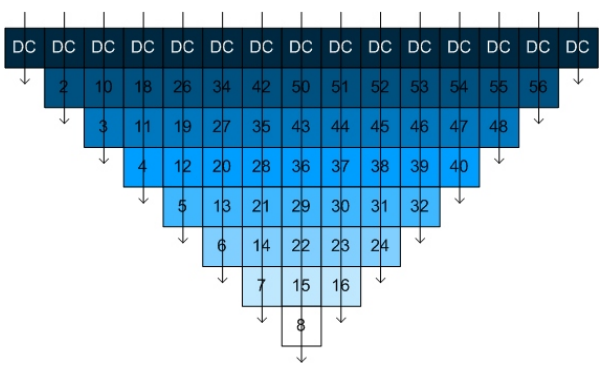

Figure 5. Results after First DCT Mode 2 Processing

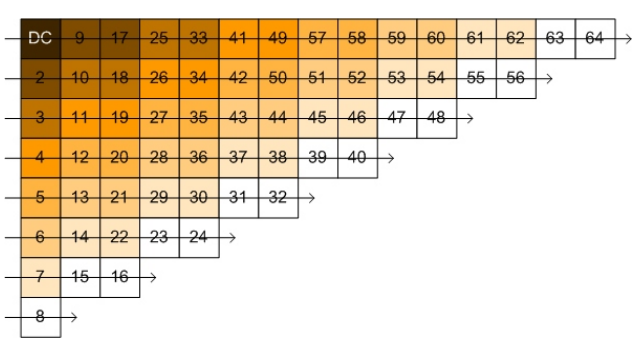

Figure 6. Final Result of DDCT Mode 2

Just like the original 2-D DCT, Figure 6 shows that all the corresponding components will be processed in the same column, which indicates that all DC values will be processed together in order to obtain the DC value of the whole map, 2nd derivative frequency values, 3rd derivative frequency values, and so on. But because the DC value of Figure 5 is the average results from the different number of pixels, we could get a so-called mean weighting defect [24] if we only do the DCT without any proper corrections.

There are generally two ways to deal with the mean weighting defect [21]. In their research, the author uses the first and also known as the simplest way to modify the weighting factors. If we define a as the original *NXN* block and A as the conventional 2-D DCT result, the conventional 2-D

DCT's coefficient block can be expressed as:

$$A = [A_{u,v}]_{N\times N} = C_{N\times N} \cdot a \cdot C^{T}_{N\times N} \quad (2)$$

Where

$$C_{N\times N} = [C_{i,j}]_{N\times N}, c_{i,j} = \alpha_i Cos\left(\frac{(2j+1)}{2N} i\pi\right)$$

$$\alpha_i = \begin{cases} \sqrt{1/N}, i = 0 \\ \sqrt{2/N}, i \neq 0 \end{cases} \quad (3)$$

By changing the weighting factor ai we can use

$$\hat{\alpha}_i = \begin{cases} 1/N_k, & i = 0 \\ \sqrt{2}/N_k, & i \neq 0 \end{cases} \quad k = 0,1,\ldots,2N-2 \quad (4)$$

to correct the DC value of mode 2.

For the rest of DDCT modes, we repeat the process, but in different directions, as shown in mode 5 in Figure 7.

After DDCT obtains all the result candidates from 7 internal modes, the selection has been made dependent on the

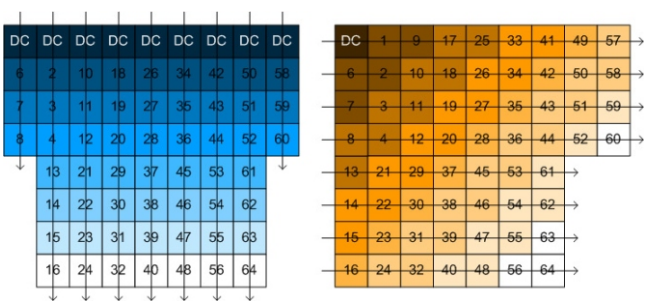

Figure 7. DDCT Mode 5 Processing

"zero" counts. Generally, the image with more high frequency zeros after quantization means lower correlation from pixel to pixel in this direction, which theoretically will lead to a more efficient compression ratio.

2.3 Simulation Results of 8-Mode DDCT

First, the author talks about the eight modes contained in the DDCT algorithm. Actually, there are only seven modes instead of eight which can be divided into 3 groups. The first group only contains mode 0 which is identical to the conventional DCT. The second group contains mode 2 and mode 3 which are diagonal modes. Finally, the third group is made up of mode 4, 5, 6, and 7 which are four symmetric semi-diagonal modes. All modes are shown in Figure 8.

H.264 codec system uses 16x16 pixel block (so-called Macro Block, MB) as the basic component of video frames. Different from the 8x8 based DDCT instance listed above from[21], we had to build our own simulation models and analysis.

Figure 9 shows some of the DDCT modes. We have 3

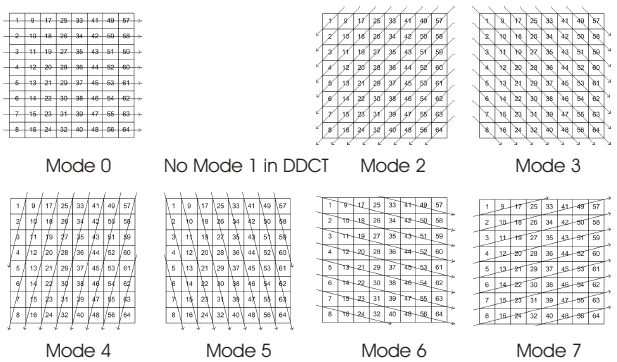

Figure 8. DDCT Modes

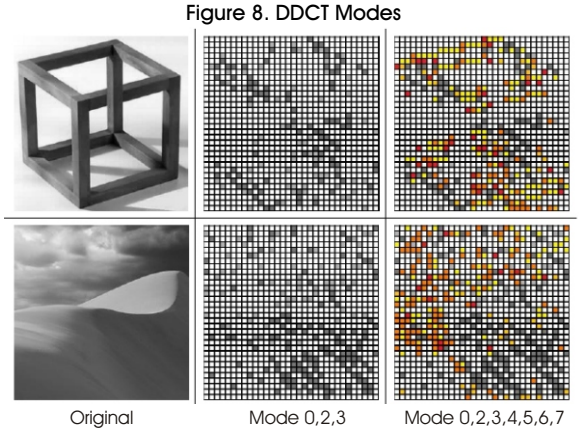

Figure 9. DDCT Modes Representing Results

columns shown in Figure 9. The left column is the original picture. The image is divided into 16x16 pixel Macro Blocks (MBs). The middle or right column is the final mode mapping DDCT result. Each tiny little square represents a 16x16 MB, and the color illustrates the mode that particular MBs are chosen from based on the 7 modes listed in Figure 8.

The reader may also notice that the middle column has only white and grayscale colors compared to the right column. That is because we are testing our DDCT in three different mode styles: first in conventional DCT mode (mode 0) only, and then, conventional DCT plus diagonal DCT (mode 0, 2 and 3), and finally, all the modes (mode 0, 2-8). We want to see if different mode combinations would affect one another, and which group is going to affect the compression ratio and/or the image quality the most.

Finally, after a number of different tests, simulations generated good results. The increase in candidate mode numbers will decrease the bitrates of our output data, while the PSNR and MSE could still remain the same.

The red, green, and black lines in Figure 10 indicate different combination of modes in DDCT. In comparison to the bottom blue line which represents the conventional DCT, they are more efficient (a better PSNR with lower bitrates).

3. Vector Bank structure and Implementation

3.1 Vector Bank introduction

Figure 11 gives us a general view of motion vectors in

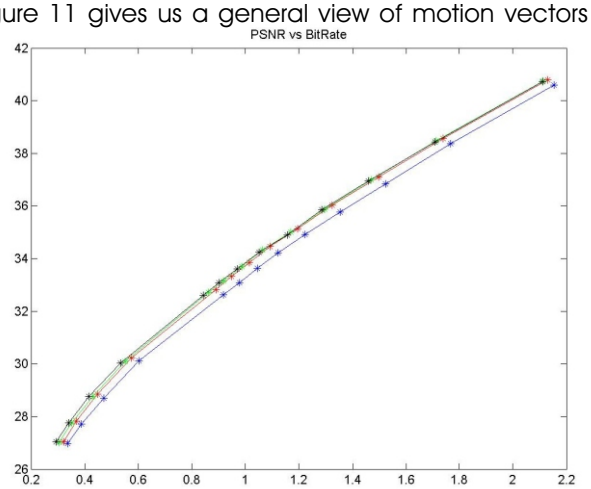

Figure 10. PSNR vs. Bitrates Test Results

Figure 11. Typical Vector Plane of Portion of Frame

some part of the reference frame. A Macro-Block (MB) with 16x16 square color pixels of the original frame represents a single block. In reality, the H.264 ME may generate MBs with motion vectors of even smaller MBs such as 16x8, 8x8, or even 4x4 square color pixels. In this paper, the author assume those MBs are all 16x16. Firstly, it simplifies our design and secondly, it matches with many other codec core designs that don't have various size MBs.

After the author got the vector data from every MB that H.264 actually processes, as Figure 12 shows, we extract those vectors to a memory based module called Vector Bank (VB). VB is a simple structure with multiple applications such as an information source. We can output them back to the ME module for better estimation decisions made in the next round. We also could output them as the guide for "focusing point" in some video capturing. In this paper, for surveillance vision camera network use, they focus more at the output for motion analysis.

3.2 H.264 SoC Based Vector Bank Implementation

Vector Bank gathers the vector information MB by MB when processing. After a frame is processed, the information for a whole frame can be saved onto a vector plane and sent to a DSP for motion analysis.

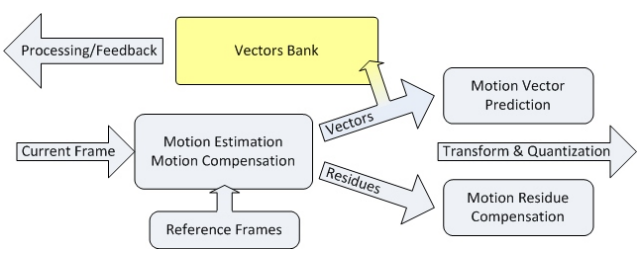

Figure 12. Vector Bank Implementation in H.264 Codec Core

As the reader can see in Figure 13, the author of this paper added the VB and DSP block into our original H.264 based SoC architecture, so that the vector information could gather by Vector Bank under the H.264 codec core, and after several frames of capturing, the DSP can generally use several different algorithms to analyze the vector data and then output them to the peripheral interfaces, such as Wi-Fi or Ethernet to pass the motion information to other systems.

As the reader can see in Figure 13, the author have added several modules for the H.264 SoC platform. The implementation is going to be inside the video codec chip where no extra hardware will be added to the whole system.

4. Edge Detection Based Motion Detection

4.1 Vector Based Edge Detection

4.1.1 Sobel Operator

The Sobel operator is a first-derivative edge detection

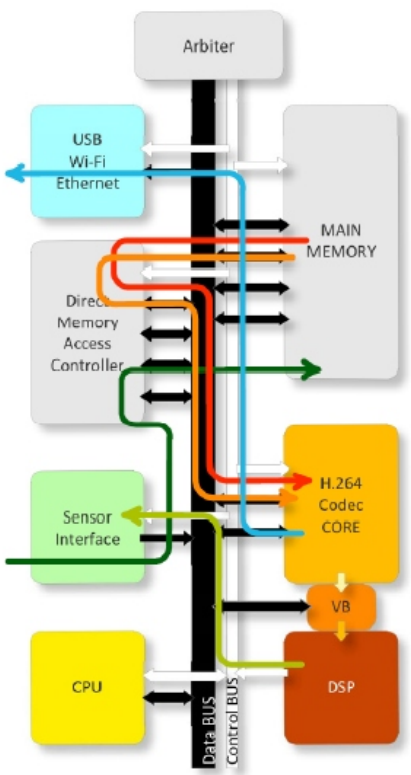

Figure 13. SoC Architecture with Vector Bank and DSP Build with H.264 Core

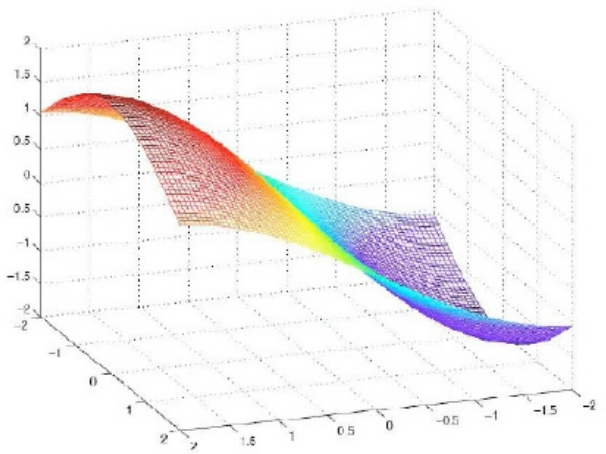

Figure 14. The Sobel Operator 3D Plot in MatLab

operator. It is simple and easy to realize in most cases. In order to detect a 2-D image edge, we need to run Sobel twice with different directions. A typical Sobel bi-directional kernel (also shown in Figure14).

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (5)$$

and

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \quad (6)$$

G_x and G_y can be combined together to get the absolute magnitude of the gradient:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (7)$$

For the fast computation, the magnitude can also be approximated as:

$$|G| = G_x + G_y \quad (8)$$

Thus, the approximate kernel for the 2-D Sobel detection operator is:

$$|G| = |(z_1 + 2 \times z_2 + z_3) - (z_7 + 2 \times z_8 + z_9)| + |(z_3 + 2 \times z_6 + z_9) - (z_1 + 2 \times z_4 + z_7)| \quad (9)$$

4.1.2 Laplacian of Gaussian (LoG) Operator

The Laplacian of Gaussian (LoG) Operator is a second-derivative edge operator. The 2-D function can be shown as:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (10)$$

The typical Gaussian Kernel with width s is:

$$G_\sigma = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (11)$$

Thus, the Laplacian of Gaussian will be:

$$\nabla^2 G_\sigma = \frac{\partial^2 G_\sigma}{\partial x^2} + \frac{\partial^2 G_\sigma}{\partial y^2} \quad (12)$$

Variable x and y are equal in this equation. We determine the x part first:

$$\frac{\partial^2 G_\sigma}{\partial x^2} = \frac{x^2 - \sigma^2}{\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (13)$$

Let E9 " F9 M9, and put x, y together back to the equation:

$$\nabla^2 G_\sigma = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} = \frac{r^2 - 2\sigma^2}{\sigma^4} e^{-\frac{r^2}{2\sigma^2}} \quad (14)$$

4.2 Simulation Result for VB with Edge Detection

As we can see in Figure 16, the upper-left corner picture is one of the video frames that we took in our library. First, they processed it using LoG with different coefficients. With different coefficients, we obtained different outputs.

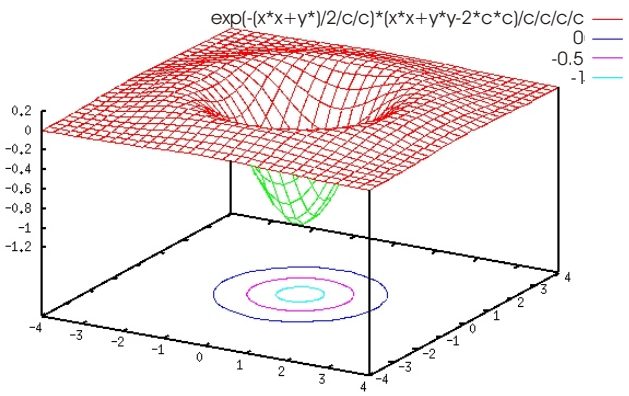

Figure 15. The Laplacian of Gaussian 3D Plot in Matlab

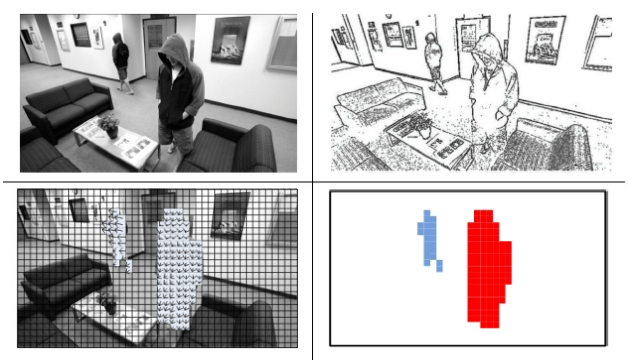

Figure 16. Motion Pictures Edge Detections with/without Vector Bank

The upper-right is one of the best results that have obtained. As the reader can see, the boundary is very complicated and most of the backgrounds are included in the computation but with our Vector Bank, the whole frame actually was divided into a lot of MBs. Each MB has a vector, and only those MBs who sense movements can have a non-zero vector – as we illustrated in the down-left corner figure. And the motion-background split is extremely easy and efficient with the help of VB, and the result is in the last figure – the down-right one.

5. Proposed Digital Video Surveillance System

With the substitution of DDCT for the DCT algorithm, the H.264 core, VB memory bank attached on the side of the H.264 core, and the edge detection algorithm programmed in the DSP ROM file, an H.264 SoC with the ability of motion detection has been proposed which is very important in this paper. So, what is the difference between the analog surveillance system structure and the proposed digital surveillance system? Let's take a look at Figure 17.

As Figure17 shows, the most common parts of analog systems are:

1) dedicated channel for each camera captured video, fixed camera mapping

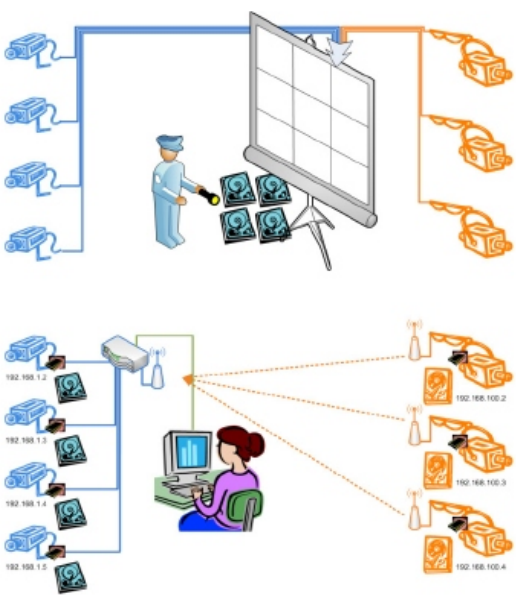

Figure 17. Analog Surveillance Systems vs. Proposed Digital Surveillance System

2) need high capacity and high speed hard drive to store everything

3) need one or more responsible guard (human) to sit in the control room

Some surveillance systems already have the camera in digital which means that:

1) they have a shared channel for video streaming

2) the channel bandwidth can be adjusted for each camera

3) they still need a guard to sit in the center of the room to watch the video, plus the need of a high capacity and high speed disk array to store tons of videos.

With the proposed H.264 SoC architecture, the camera itself has the ability to detect intrusion and response to illegal motion, which means that we do not need a guard to analyze those videos and the system becomes:

1) shared data channel just like common digital systems where cameras could be added or eliminated without routing concern

2) video need not be transferred unless potential intrusion is detected

3) video could be stored under every surveillance camera, but doesn't have to transferred for such a long distance, which means it is more reliable and unbreakable

4) does not need to have a guard to sit in front of the screen 24 hours a day

In conclusion, the proposed digital surveillance system

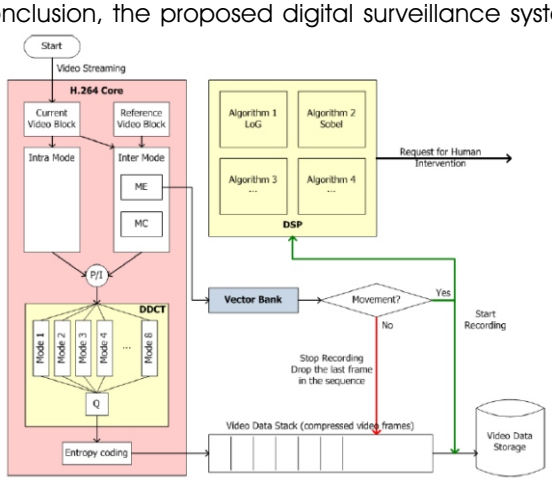

Figure 18. Algorithm of Proposing Vector Bank Based Digital Surveillance System

introduces the DDCT, Vector Bank, and Motion Detection DSP into the H.264 SoC design as Figure 18 shows. Finally, with all the presented technology, the digital surveillance network system becomes possible and efficient.

6. Future Works

In this paper, the author present several techniques for an H.264 SoC based video codec surveillance network system. They believe that there will be tons of possibilities if the surveillance system could be digitalized and SoC chip distributed. However, even the experimental result looks promising, but Vector Bank and Edge detection still cannot replace the human-guarded surveillance, because we are still missing most of the features, such as face detection, real-life measurement, or even following the target. However, the digital world is unlimited. The computer is more reliable than human-being in particular ways. The next step for us would be the object detection and multiple cameras based on 3-D measurement for objects.

References

[1]. Ahmed, N. Natarajan, T. and Rao, K. R. (1974), Discrete cosine transform, *IEEE Trans. Computer*, Vol. 23, No. 1, pp.90-93, January.

[2]. Chen C.Y. Chien S.Y. Huang Y.W. Chen T.C. Wang T.C. Chen L.G. (2005), "Analysis and Architecture Design of Variable Block Size Motion Estimation for H.264/AVC," *IEEE Transactions on Circuits and Systems I*, No.99.

[3]. Chen, T.C. Lian, C.J. Liang, G. (2006), Hardware architecture design of an h.264/avc video codec.

[4]. Clark, R. J. (1985), Transform Coding of Images. London, U.K.: Academic.

[5]. Cucchiara, R. Grana, C. Piccardi, M. and Prati, A. (2000), Statistic and knowledge-based moving object detection in traffic scenes, *IEEE Proceedings. Intelligent Transportation Systems*, pp.27-32.

[6]. Cumani, A. (1991), Edge Detection in Multispectral Images, CVGIP: *Graphical Models and Image Processing*. Vol. 53, pp.40-51.

[7]. Gersho, A. and Gray, R. M. (1991), Vector Quantization and Signal Compression. Boston, MA: Kluwer.

[8]. Gonzalez, R. C. and Woods, R. E. (2001), *Digital image Processing*, Vol. 10, No. 2, pp.585611.

[9]. H3C surveillance system at experience center, (2009), Retrieved from H3C website: http://www.h3c. com /portal/About_H3C/Photos/, 12th November 2009.

[10]. He, Zhihai, S. M. A (2001), unified rate-distortion analysis framework for transform coding, *Circuits and Systems for Video Technology*, Vol. 11.

[11]. Iain E. G. Richardson. (2003), H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia. John Wiley & Sons, Ltd.

[12]. Jayant, N. S. and Noll, P. (1984), Digital Coding of Waveforms. Englewood Cliffs, NJ: Prentice-Hall.

[13]. Jung, Y.K. Lee, K.W. and Ho, Y.S. (2001), Content-based event retrieval using semantic scene interpretation for automated traffic surveillance, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 2, pp.151-163.

[14]. Kauff, P. and Schuur, K., (1998), Shape-adaptive DCT with block-based DC separation and ΔDC correction, *IEEE Trans. Circuits Syst. Video Technology*, Vol. 8, No. 3, pp.237-242.

[15]. Li, D. (2000), Moving objects detection by block comparison, Electronics, Circuits and Systems, Vol. 1, pp.341-344.

[16]. Marr, D. and Hildreth, E. (1980), Theory of Edge Detection, *Proceedings of the Royal Society of London*, B207, pp.187-217.

[17]. Montoliu, R. and Pla, F. (2001), Multiple parametric motion model estimation and segmentation, ICIP 2001, Vol. 2, pp.933-936.

[18]. Novak, C.L. and Shafer, S.A. (1987), Color Edge Detection, *Proceedings DARPA Image Understanding Workshop*, Vol. I, pp.35-37, Los Angeles, CA, USA.

[19]. Rabiner, L. R. and Schafer, R. W. (1978), Digital Processing of Speech Signals. Englewood Cliffs, NJ: Prentice-Hall.

[20]. Rao, K. R. and Yip, P. (1990), Discrete Cosine Transform-Algorithms, Advantages, Applications. London, U.K.

[21]. Tourapis, H.-Y.C. Tourapis, A.M. (2003), Fast motion estimation within the h.264 codec, Conf. Multimedia and Expo, ICME '03, Vol. 3, pp.III 517-520.

[22]. Vaidyanathan, P. P. (1993), Multirate Systems and Filter Banks. Englewood Cliffs, NJ: Prentice-Hall.

[23]. Vetterli, M. and Kovacevic, J., (1995), Wavelets and Subband Coding. Englewood Cliffs, NJ: Prentice-Hall.

[24]. Wiegand, T. (2003), Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264--ISO/IEC 14496-10 AVC), in Joint Video Team (JVT) of ISO/ICE MPEG and ITU-T VCEG, VT-G050

[25]. Zeng, B. and Fu, J. (2008), Directional Discrete Cosine Transforms A New Framework of Image coding, *IEEE Transactions on Circuit and Systems for Video Technology*, Vol. 18, No. 3.

ABOUT THE AUTHORS

Wei Zhao received his Master of Science degree in EE from Beihang University, and Bachelor's degree in EE from Zhejiang University, China in 2006 and 2005 respectively. After graduation, he worked at Vimicro Corp. as an IC Verification Engineer, designing part of a verification work flow for a multi-media SoC chip. Verification bring him a lot interest in the field of computer architecture, and he started working on his Ph.D. degree at Electrical and Computer Engineering Department of Florida International University, Miami, FL, in September 2007. His technical interests lie in the field of H.264 architecture design and implementation with optimization.

Dr. Jeffrey Fan is currently an Assistant Professor in Electrical and Computer Engineering at Florida International University. His research interests include very-large-scaled-integrated (VLSI) circuit simulation, modeling, optimization, bio-electronics, embedded real-time operating systems in application to robotic control, and wireless communications in sensor networks. Prior to his academic career, Fan served as Vice President of Vivavr Technology, Inc., and General Manager/co-founder of Musica Technologies, Inc. From 1988 to 2002, he held various senior technical positions in California at Western Digital, Emulex Corporation, Adaptec Inc., and Toshiba America. His product line of research and development includes Virtual Reality (VR) 3-D animation, MP3 players, hard drives, fibre channel adapters, SCSI/ATAPI adapters, RAID disk array, PCMCIA cards and laser printer controllers.
Dr. Fan received his Ph.D. degree in electrical engineering at University of California, Riverside in 2007, and the Master of Science degree in electrical engineering from State University of New York at Buffalo in 1987. He also holds Bachelor of Science degree in electronics engineering from National Chiao Tung University in Taiwan, R.O.C. He has served as a steering committee member of SSST, a technical program committee member for ICESS, CAMAD, ISQED, ISCAS, and an invited tutorial speaker for ASICON'07. He is a Senior Member of IEEE.

Dr. Asad Davari is currently working as a Full Professor in Department of Electrical and Computer Engineering in WVU Tech. He received his Bachelor of Science degree in 1980, the Master of Science degree in 1981, and Ph. D from University of Alabama in Huntsville in 1985. He teaches and conducts research in the area of Control Theory and Applications as well as energy-related problems. Davari is a senior member of IEEE, a member of Eta Kappa Nu and IEEE WV Section Chair 2009. His research was recognized by West Virginia Governor with a Certificate of Achievement in Scientific Research on February 9, 2004.