An Improved Wavelet Filtering Algorithm and Its FPGA Implementation

Lan Yang

School of Information Engineering

Chang' an University   Xi' an, China

E-mail: yanglan.1985@163.com

Xiang-mo Zhao   Fei Hui   Xin Shi

Chang' an University   Xi' an, China

E-mail: {xmzhao, feihui , shixin} @chd.edu.cn

Abstract—**The de-noising of sensor data has become an important to research. Since the traditional de-noising method can't achieve successful de-noising effect and the software-only method never meets a high real time capability. In this paper, we illustrate a novel threshold function based on the wavelet hard and soft threshold function. It is unlike ordinary function, which has overcome the defect such as the discontinuity of hard threshold function and an invariable dispersion between the estimated wavelet coefficients and the decomposed coefficients of soft threshold function. Moreover, we consider the hardware implementation of wavelet threshold filter on FPGA which adopt the pleated sheet structure of multiplier and fit to frame data. A detailed description of the simulation and implementation is given. Finally, the experiment result on-board is shown that our hardware implementation can meet the requirement of real-time signal processing.**

Index Terms—**FPGA implementation, Wavelet Threshold Filtering, Digital Signal Process**

I.   INTRODUCTION

Nowadays, wavelet de-noising algorithm is wildly used in many fields. At the same time, wavelet threshold de-noising algorithm has been widely adopted as a successful method, which has several merits such as simple calculation, well-efficient and more suitable for low SNR. There is a lot of literatures about wavelet threshold de-noising and excellent reviews can be found in [1]-[3].

In our project, the huge amount of vehicle sensor data is vulnerable to radio frequency radiation interference or other factors. Therefore the noise-containing signals should be filtered to gain the real signals. Since the software-only can't meet a high real time capability, more and more experts consider the hardware circuit to implement. The FPGA is a very useful device for realize a specific digital electronic circuit in diverse industrial fields [4]. From G. Knowles firstly put forward the method of VLSI architecture to now, There are several successful implementation of ASIC such as in [5]-[8]. For example, wavelet codec based on FPGA which designed by Nevin A Elghamery and S.E.-D.Habib, wavelet transform coprocessor based on FPGA which given by M.Nibouche,A. Bouridane and so on.

Most of these projects demonstrated above are successful in some conditions, but it is pertinence for some products but lacking of application. There are two major problems under discussion. The first question is how to choose an appropriate threshold according to the signal intensity for the over-reservations or over-kill phenomenon [9]. Besides that, how to establish a relational module in the internal dimension and between

the dimensions is also important [10].

For the purpose of solving these problems, in this paper, we have proposed an improved de-noising algorithm and implement it on FPGA. We would like to emphasize features of the research as follows.

1) Summarize the wavelet transform and its hardware implementation.

2) Discuss the merits and defects of the ordinary de-noising threshold function and establish a novel mathematical model.

3) Modularize design a FPGA wavelet de-noising filter and contrast the simulation result between MatLab and FPGA.

4) Analyze the result of on-board implementation in Xilinx XC5VLX110T FPGA.

II. *Wavelet Transform and Threshold De-noising Algorithm*

In engineering, a one-dimensional model of signal with additive noises can be shown in as follows [11]:

$$f(t) = f_s(t) + f_n(t). \tag{1}$$

Where, $f(t)$ denotes noise-containing signal, $f_s(t)$ denotes the real signal, and $f_n(t)$ is white Gaussian noise with a normal distribution [12]. By f(t) function for discrete sampling are n points discrete signal f(n), when n = 0, 1, 2…, N-1. The wavelet transform can be shown in as follows:

$$Wf(j,k) = 2^{-j/2} \sum_{n=0}^{N-1} f(n)\phi(2^{-j}-k), j,k \in z \tag{2}$$

Where, W(j,k) is wavelet transform coefficients. In practical application, the calculation formula (2) is very cumbersome and wavelet base function   (t) generally no explicit expression. Thus, recursive wavelet transform method is shown in as follows:

$$sf(j+1,k) = sf(j,k) * h(j,k) \tag{3}$$

$$wf(j+1,k) = sf(j,k) * g(j,k) \tag{4}$$

Where, h and g are the low-pass and high pass filters of scaling function and wavelet base function. $S(0,k)$ is the original signal, $S(j,k)$ for the scale coefficient, $W(j,k)$ for the wavelet coefficient. The corresponding reconstruction formula is shown as following.

$$sf(j-1,k) = sf(j,k) * \hat{h}(j,k) \tag{5}$$

$$sf(j-1,k) = sf(j,k) * \hat{h}(j,k) + Wf(j,k) \hat{g}(j,k) \tag{6}$$

Where, $\hat{h}$ and $\hat{g}$ are the reconstruction low pass filter and high pass filter.

In this paper, $W_{j,k} = Wf(j,k)$, Because the wavelet transform is a linear transform, After $f(t) = s(t) + n(t)$ been discrete wavelet transformed, obtaining the wavelet coefficients $W_{j,k}$ which still consists of two parts, one is wavelet coefficients $Ws(j,k)$ of the signal s(k), denoted by $U_{j,k}$, another is wavelet coefficients $Wn(j,k)$ of noise n(k), denoted by $V_{j,k}$.

Donoho and Johnstone have proposed wavelet threshold de-noising method. The basic method is shown as following. $W_{j,k}$ is discarded when $W_{j,k}$ less than some critical thresholds. There, $W_{j,k}$ is mainly caused by noise, and $W_{j,k}$ is preserved (hard threshold methods) or shrink to zero by a fixed amount (soft threshold method) when $W_{j,k}$ is greater than or equal to this critical threshold. There the wavelet coefficients are mainly caused by the signal. Then it uses the new wavelet coefficients to reconstruct wavelet filters and obtain the de-noising signal.

.Generally, the useful real signals usually behave in the form of low-frequency signals or certain relatively stable signal, while noise signal are usually in the form

of high-frequency signals. The objective is to suppress the additive noise $f_n(t)$ from the signal $f(t)$. The process of wavelet threshold de-noising contains three steps as Fig. 1 and a detail description is stated bellow.

Figure 1. Wavelet de-noising process

1) Calculate the orthogonal decomposition of $f(t)$, choose an appropriate wavelet and the layers of wavelet decomposition and obtain the approximation and detail coefficient.

2) Choose an appropriate threshold function and Quantizing operation for each layer of the wavelet coefficient.

3) Reconstruct the wavelet coefficient and recovery the estimated value of the real signal $f_s(t)$.

Obviously, the selection of threshold directly influences the effectiveness of the de-noising algorithm. For any threshold, two kinds of threshold function can be used: hard and soft threshold functions. Their mathematical expressions are as follows [13]:

$$w_h(j,k) = \begin{cases} w(j,k) & ,|w(j,k)| \geq \lambda \\ 0 & ,|w(j,k)| < \lambda \end{cases} \quad (7)$$

$$w_s(j,k) = \begin{cases} sign(w(j,k))(|w(j.k)|-\lambda), & |w(j,k)| \geq \lambda \\ 0 & ,|w(j.k)| < \lambda \end{cases} \quad (8)$$
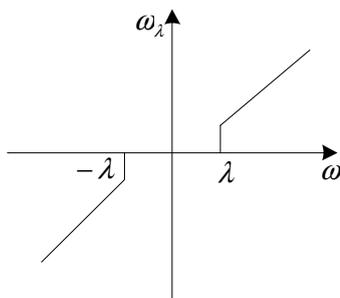
Figure 2. Hard Threshold Function
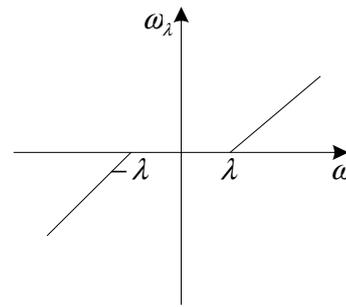
Figure 3 Soft Threshold Function

Where, $w(j,k)$ is wavelet coefficient from the decomposed noise-containing signal. The variable λ is the selected threshold. The function of $w_h(j,k)$ and $w_s(j,k)$ is the wavelet coefficient processed with hard and soft threshold function. Sign() is signum function, is the noise threshold. There are four main methods to determine the threshold: The Stein SURE (rigrsure), heuristic threshold selection (heursure), common threshold $\sqrt{2\log()}$ (hard threshold and soft threshold function).

As we see, on the one hand, too high a threshold would result in wavelet decomposition coefficients reset as zero and destroy the local signal. On the other hand, too low a threshold can't achieve the expected de-noising effect. For example, the hard threshold function is easily to lead in the problem of discontinuity in an interval between λ and $-\lambda$. The soft threshold function is easily to lead in invariable dispersion between $|w(j,k)|$ and $w_s(j,k)$ while wavelet coefficient is larger than a threshold.

III. Improved threshold Algorithm

In order to make up for the defect in these methods above-mentioned, we hold up an improved threshold function. It is shown in formula (9).

$$\hat{w}(j,k) = \begin{cases} sign(w(j,k))(|w(j,k)| - \dfrac{1.5\lambda \times k}{\exp(\frac{|w(j,k)|-1.5\lambda}{N} \times 1.5\lambda)}), & |w(j,k)| \geq 1.5\lambda \\ (\dfrac{-(w(j,k)-0.5\lambda)^3}{2\lambda^2} + \dfrac{2(w(j,k)-0.5\lambda)^2}{\lambda})(1-k), & 0.5\lambda < w(j,k) < 1.5\lambda \\ (\dfrac{-(w(j,k)+0.5\lambda)^3}{2\lambda^2} - \dfrac{2(w(j,k)+0.5\lambda)^2}{\lambda})(1-k), & -1.5\lambda < w(j,k) < -0.5\lambda \\ 0 & ,|w(j,k)| \leq 0.5\lambda \end{cases} \quad (9)$$

Where, weight factor $k \in [0,1]$. The values of k is relevant to the threshold. That is, when the threshold of noising is larger, the value of k should set small, or set larger. It ensure that the function $\hat{w}(j,k)$ is continuous when $|w(j,k)| = 0.5\lambda$ or 1.5λ. With the $|w(j,k)|$ increasing, the aberration between $\hat{w}(j,k)$ and $|w(j,k)|$ decrease. That is to say, while $|w(j,k)|$ approach infinity, $\hat{w}(j,k)/w(j,k)$ approach to 1. As a result, it can cut down the constant error in soft threshold function.

Simulation analysis on MatLab

In order to illustrate the validity and advantage of new threshold function in de-noising algorithm, we use the traditional soft threshold function and the new threshold function on MATLAB de-noising simulation.

The input noise-containing data for synthesis simulation are got from MatLab. For example, Fig.4 shows noise-containing data which consists of a built-in function SIN with zero mean and SNR 3db of white noise source for 1024 sample points. We store it as a text file and read it into ISE using the Verilog HDL statement as input data. The Fig. 5 shows the de-noising signal on MatLab with soft threshold function. The Fig. 6 shows the de-noising signal on MatLab with new threshold function.
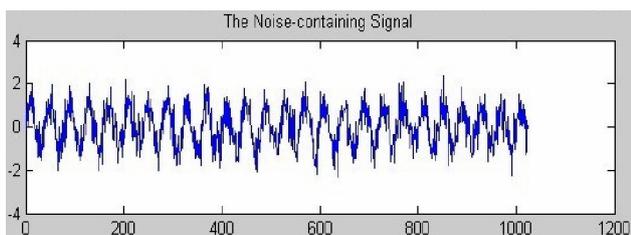
Figure 4. Noise-containing signal
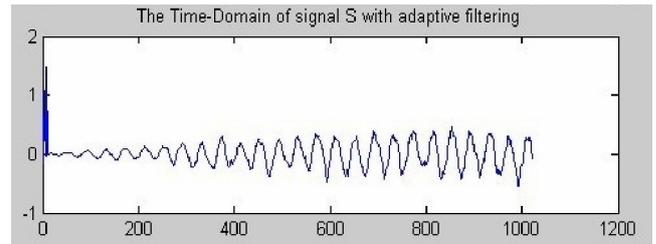
Figure 5. Soft threshold function simulation result
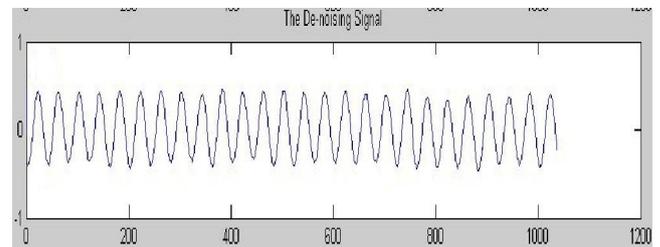
Figure 6 New threshold function simulation result

Where, the x-coordinate represent signal data and the y- coordinate represent sample points. By contrast with the software method in MatLab, it shows the improved algorithm and its hardware implementation achieve a better de-noising effect. However, the question is that coefficient form MatLab is generally real number and the hardware memory can only store integer. Consequently, the data should be taken the integer value for left shifting with n bits to be stored.

In this algorithm, the filtering result is almost same as the desired signal with rapid convergence. It greatly reduces steady-state error and improves the convergence precision. This algorithm can well remove the signal noise and the output signal can converge to the input signal well.

IV. FPGA HARDWARE IMPLEMENTATION

A. *Modularize Designing*

The purpose of the work is to design an IP core which can handle the combinational of sequential logic functions to carry out our improved algorithm. It is based on Verilog HDL behavior description, all of the system components have been described with structural architecture. Its functional diagram is show in Fig. 7. The system is divided into four parts. It contains input module, memory module, filter module, coefficient processing module and output module. Where, the wavelet filter module is divided into filter module and coefficients processing module. Firstly, we configure the system with db4 wavelet and 2 layers for wavelet

transform.
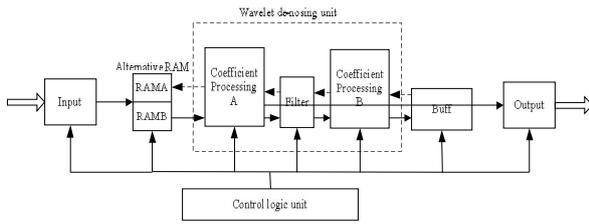
Figure 7. Architecture of the de-nosing system

Where, the parameter RAM stores parameters which are used to configure system. The internal RAM stores intermediate data in signal processing. Wavelet filter process is as follow description. It receives the coefficients which are used to configure parameters into the parameters RAM, and then it starts the data filtering.

Decomposition process is described as following. When the internal signal of reset and the clock are effective, the input data is transferred to the RAMA or RAMB. Then the coefficient processing-A received and deliver data to filter for wavelet decomposing. After that, the coefficient processing-B received the data, which acquire the detail coefficient with down-sampling operation and process it with threshold function. Finally, the coefficient processing-B writes the data into the data buffer. That is decomposing and threshold processing of layer 1.

Layer 2 process is as following. The coefficient processing-B reads the detail coefficient from the buffer and transfers it to the filter for completing the second wavelet decomposing. The coefficient from the filter is delivered to the coefficient processing-A for completing the down- sampling and threshold processing. At last, the data is written to the RAMA or RAMB. Decomposing and threshold processing of layer 2 is completed. Compare with the decomposing, the transmission mode of the data in reconstruction process is in the reverse direction. Once wavelet reconstruction completed, the data is output which is controlled by the coefficient processing module.

Input module is used to receive data, including the data need to filter and configuration parameters. The configuration parameters include the filter coefficients, decomposition levels, the signal length and threshold. The input module implements in a second election between RAMA and RAMB. Filter coefficients,

decomposition levels, the signal length and threshold are written in given address space of parameters RAM in accordance to be called back during the data processing. Output module implements output control logic for judging whether the current system completed the last reconstruction. If it is, it outputs the data directly and processing of a frame data completed by wavelet threshold de-noising.

It requires a total of 4 blocks: RAMA, RAMB, the internal RAM and the parameter RAM. RAMA and RAMB used to cache input data and intermediate data of filtering process. The internal RAM for caching filtering the intermediate data. Parameters RAM stores system configuration parameters, filter coefficient, decomposition level, signal length and threshold. Parameter RAM stores data are used to configure the system. Provisions of the various parameters stored address in the parameter RAM, so that when the data are called back, it can directly read the specified address.

Wavelet filtering module completes wavelet decomposition, coefficients processing and wavelet reconstruction. Wavelet decomposition and reconstruction is essentially a convolution operation. So, the wavelet filter module can be divided into filter modules and Coefficient processing module.

The filter structure is shown in Fig. 8. Where, the implementation of the filter adopted the double multiplier module which transferred the data to high-pass filter and low-pass filter to calculate [13]. As the results, it improved parallel efficiency and shorten the running time one time.
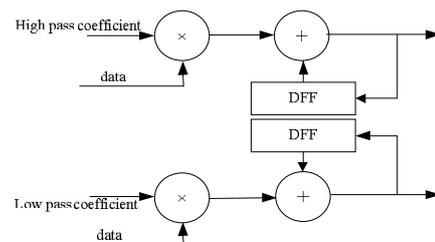
Figure 8. Filter with double multiplier

Coefficient processing structure is shown as in Fig. 9. It is implemented with FSM for four states. Its main task is stated below.

1) In decomposing stage, it performs down-sampling operation to decompose coefficient and de-noised

the detail coefficient with threshold function.

2) In reconstruction stage, it performs up-sampling the recombined approximation and detail coefficient.
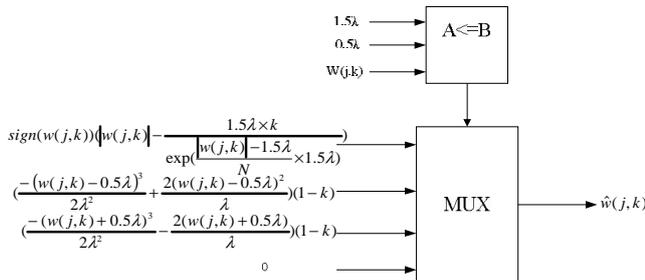
Figure 9. Coefficient processing structure

The coefficient processing-A used in conjunction with coefficient processing-B. Coefficient processing unit is divided into four states: state0, state1, state2, state3. Where, state0 started to read data from RAM and sent to the filter module in wavelet decomposition. State1 receives filters module processed data and after the data drop-sampling and threshold processing to write in RAM in wavelet decomposition. State2 reads data from RAM and after sampling input to filter module in wavelet reconstruction; State3 receives filter module data and after adding the output data of high filter and low filter write into RAM module.

B. Simulation and Verification

The implementation of the improved de-noising algorithm is performed on Xilinx XC5VLX110T-FF1136 FPGA (speed grade-3) with 550MHz of main block RAM. The XC5VLX110T consists of 160*54 arrays of Configurable Logic Blocks (CLB) [14].

We write the configuration parameters, which include frame data length, filter coefficients, wavelet transform layers and the threshold value. The data frame length is 128. Length is 7. Wavelet transform layer is 2. The first layer decomposition of wavelet coefficients processing threshold is 4500, and the second layer threshold is 4600. We chose of "db4" wavelet function.

Experiment signals with 1024 byte a frame are input as the sensor data. We set the order of the filter as 8. We verified the synthesis, simulation, placing and routing process. The buffers implement efficiently with the alternative RAM distributed along each CLB. The simulation result on FPGA with improved de-noising algorithm is shown in Fig. 10 which display on Modelsim. Where the first signal is the contain noise

signal, the second is the true signal and the last is the filtering result signal. The implementation performance is shown in Table .
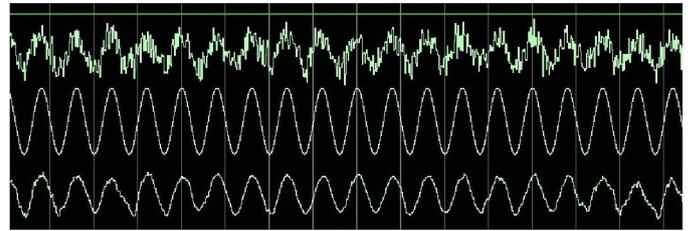
Figure 10. FPGA simulation result with new algorithm on Model si m

TABLE .

IMPLEMENTATION PERFORMANCE

	Available	Used	Percentage
Logic Elements	69120	27451	39%
Memory Bits	28485	1043	3%
Dedicated Logic Registers	69120	2080	3%
Embedded Multiplier	64	32	50%
Maximum Clock	383.65 MHz		
Data Throughout	1.049 Gbit/s		

V. RESULT DISCUSSION

The DSP48E is embedded in Xilinx FPGA which complete a multiply operation in a clock cycle for 550MHz, therefore, all of the multiply-accumulate operation at most requires 89us. The data reading require 1.86us. As it is apparent from Table 1, filter runs at a maximum clock speed 383.656MHz easily. It can meet the requirement of real-time signal processing while the frequency of sensor data is 1MHz. What's more, the method we presented performed well and can be applied to different FPGAs.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an improved de-noising algorithm for simulation and implementation. The new threshold function is based on the hard threshold and software threshold function. We have experimented with both synthetic data and real sensor data. The simulation results designed on ISE achieve obvious effect. Implementation using the FPGA is shown that our sensor signal can filter the noising signal caused by the electromagnetic radiation, and meet the real-time

demands, which at the same time demonstrate the better effectiveness. Therefore, it was relatively predominant to the software-only implementation.

However, there still some problems should be focused on. For our future work, we will mainly focus on two aspects:

1) Increase the bit number of operators to increase the operation exactitude.
2) Set more appropriate pipeline to gain operation speed.

Acknowledgment

REFERENCES

[1] Donoho D I. De-noising by soft-threshold. IEEE Trans. Inform.Theory,1995,41(3):613-627.

[2] Donoho DL Johnstone I M Adapting to Unknown Smoothness Via Wavelet Shrinkage[J] Journal of American Stat Assoc 1995 12(90) 1200-1224.

[3] G. Strang and T. Nguyen, "Wavelets and filter banks," Wellesley- Cambridge Press, 1997.

[4] Engin Avci, Comparison of wavelet families for texture classification by using wavelet packet entropy adaptive network based fuzzy inference system, Applied Soft Computing, v.8n.1,p.225-231,January, 2008 .

[5] Nevin A Elghamery, S.E.-D.Habib. An efficient FPGA implementation of a wavelet coder/decoder. Microelectronics, 2000. ICM 2000. Proceedings of the 12th International Conference on Oct.2000.269-272.

[6] M.Nibouche, A.Bouridane. An FPGA-based wavelet transforms coprocessor. Image processing, 2001. Prceedings.2001 International Conference on Oct.2001.3.194-197.

[7] T.T.Cai,B.W.Silverman.Incorporating information on neighbouring coefficients into wavelet estimation.Sankhya: The Indian Journal of Statistics,63(B),Pt.2,2001:127-148.

[8] Tian Xin, Tan Yihua, Tian Jinwen,New design for 9/7-tap wavelet filter and its VLSI implementation.Jisuanji Xuebao/Chinese Journal of Computers.Vol.31,no.3,pp.411-418.Mar.2008.

[9] M. Nibouche, O. Nibouche, A. Bouridane, "Design and implementation of wavelet based system," Proc. of the 10th IEEE Conf. on Elect., Circuits and Systems, vol. 2, pp. 14-17, 2003.

[10] J. Carletta, G. Giakos N. Patnekar, L. Fraiwan and F. Krach, Design of a field programmable gate array-based platform for real-time denoising of optical imaging signals using wavelet transforms ,Elsevier, vol.36, 2004 289-296.

[11] Peng Jiang, Qingbo Huang, Yaguang Kong1, and Li Chai,Research on a   Denoising Method Based on Wavelet Packet Shrinkage for Pulp Thickness Signals.The First International IEEE conference of Multi-Symposiums on   Computer   and   Computational   Sciences (IMSCCS'06).2006.

[12] L. Brechet, M. Lucas, C. Doncarli, and D. Farina, "Compression of   Biomedical Signals With Mother Wavelet Optimization and Best-Basis Wavelet Packet Selection," IEEE Trans on Biomedical Engineeiring, Vol. 54, NO. 12, p.p. 2186- 2192, Dec 2007.

[13] Donoho DL   Johnstone I M   Ideal Spatial Adaptation Via Wavelet   Shrinkage[J]   Biometrika   1994   81(12) 425-455.

[14] "Virtex-5 Family Series Field Programmable Gate Arrays", Xilinx 2009.

Lan Yang was born in XingPing city, China, in August 1985. She received a B.E degree in Computer Science and Technology from the Xianyang Normal college, China, in 2008. And she will receive the Master of engineering degree in Technology of Computer Application from the Chang'an University, China, in 2011.

Currently, she is recommended to working towards a Ph.D degree at Traffic Information Engineering and Control, System Development Lab, Chang'an university. Her areas of interest include Research of Embedded Control System, Digital Signal Processing and FPGA implementation.

Xiang-mo Zhao was born in ChongQing city, China, in August 1966. He received a B.E. degree in Technology of Wireless from The University of ChongQing, China, in 1987. He received the Master of engineering degree and Ph.D degree in Traffic Information Engineering and Control from the University of Chang'an, China, in 2006.

Currently, he is a Professor, tutor of Ph.D. students and the dean in the School of Information Engineering of Chang'an University. His area of interest include Distribute Computer Network Control Technology and Application, Traffic Information Technology and ITS.

Fei Hui was born in HuiBei city, China, in February 1982. He received a B.E. degree in Computer Science and Technology from The Northwestern Polytechnical University, China, in 2003. He received the Master of engineering degree and Ph.D degree in Computer System Architecture from Xi'an Institute of micro-electronics Technology, China, in 2006.

Currently, he is a Vice-Professor in the School of Information Engineering of Chang'an University and the leader of System Development Lab. His area of interest include Information Processing, Design for reliability of Embedded System and Architecture Study of Parallel Computer.

Xin Shi was born in AnYang city, China, in January 1987. He received a B.E degree in Computer Science and Technology from the Chang'an University, China, in 2008. And he will receive the Master of engineering degree in Technology of Computer Application from the Chang'an University, China, in 2011.

Currently, he is recommended to working towards a Ph.D degree at Traffic Information Engineering and Control, System Development Lab, Chang'an university. His areas of interest include Wireless Sensor Networks, Digital Image Processing.