OPTIMIZATION ALGORITHM FOR EMBEDDED LINUX REMOTE VIDEO MONITORING SYSTEM ORIENTED TO THE INTERNET OF THINGS (IOT)

WENBO FU*
School of Mathematics and Computer Science, Shanxi Datong University
Datong, China

DEBNATH NARAYAN
Department of Computer Science, Winona State University
Winona, MN 55987, USA

ABSTRACT. At present, the remote video monitoring system has the problem of weak anti-interference ability and poor response of the system. Therefore, the video image is not clear. On the basis of the Internet of things (IOT), a design method of embedded Linux remote video monitoring system is proposed. The method is based on ARM+Linux development platform, the 301V USB camera of Vimicro is used to collect images, to make preprocessing, and improve the system’s response. The embedded Linux operating system is used to realize the functions of data acquisition and transmission of video image. The fractal wavelet of multivariate statistical model is used to denoise the video image so as to improve the anti-interference of the system. The experimental results show that the method has strong anti-interference ability and good response to the system.

1. Introduction. Remote monitoring is the function of local computer to monitor and control remote terminal through network system, to complete the state monitoring of distributed control network, and the diagnosis and maintenance of equipment and so on [18]. The communication media, computer software and hardware system, which can realize remote monitoring are called remote monitoring system [11]. Embedded system is a special computer system, which is centered on application and based on computer technology; software and hardware can be tailored. It is a special computer system which is suitable for the requirement of application system for function, reliability, cost, volume, power consumption and so on [23]. Remote video monitoring system based on embedded technology can effectively integrate embedded technology and video processing technology, and solve the problem in traditional video monitoring system based on PC [5,13]. Embedded video monitoring system is a newly rising video monitoring system with the core of digital video compression technology. It is a new high-tech product [10], which is an organic combination of embedded technology, computer technology and video processing technology. The embedded remote video monitoring system shows strong vitality with its superior performance [16,17].

The embedded remote video monitoring system has many advantages.
(1) Convenient for computer processing. As the image video is digitized, it can be compressed, analyzed, stored and displayed by the fast processing capability of the computer [7].

(2) Suitable for long distance transmission. Digital information has strong anti-interference ability and is not susceptible to attenuation of transmission line signal. It can monitor the scene in real time thousands of miles away, and achieve distributed monitoring [8, 22]. The whole video monitoring network can be integrated with the enterprise computer network. The video signal of the front end of the system can directly enter the Ethernet and Intranet network. Authorized users can monitor the scene in anywhere through Web browsers [9, 14]. It can access the Internet locally, and can also be remotely netted through a modem. In this way, multiple terminals at different locations are monitored at the same time, and the monitoring range for each monitoring terminal can be set [6].

(3) To improve the quality of the image and the efficiency of monitoring. By using computers, it can process noise and sharpening of indistinct images. We can see clear high-quality images by adjusting the size of the images and using the high resolution of the display [15, 21]. In addition, multiple video images can be watched on a display at the same time.

Embedded remote video monitoring technology is based on embedded Linux development and image processing technology. It has high requirements for stability, real-time and compatibility. It represents the development direction of modern embedded video technology [19, 20]. When the current method is used to design a remote video monitoring system, there is a problem of weak anti-interference ability and poor response of the system [2]. To this end, a design method of embedded Linux remote video monitoring system oriented to the Internet of things is proposed.

2. Design of embedded Linux remote video monitoring system.

2.1. The overall framework of embedded Linux remote video monitoring system. In this paper, a universal and high-performance embedded video monitoring system based on embedded technology is proposed. It is an embedded system that realizes data acquisition and data transmission through the network to the background server. The system is based on ARM+Linux as the core development platform. The system uses TCP/IP protocol technology to complete the network communication, and the front end of embedded video collects image and video data. After processing, it is transmitted to the background server through the network, and the client can achieve real-time monitoring. The system uses B/S architecture as a whole. The client can receive pictures through the browser with JAVA plug-in and realize real-time monitoring. Figure 1 is the overall framework of the embedded Linux remote video monitoring system.

The embedded Linux remote video monitoring system is divided into the following four parts:

1) The acquisition of front-end by camera. In the embedded Linux remote video monitoring system, the front-end is acquired by 30IV camera of Vimicro. The camera has the hardware compression function for JPEG image, it is mainly used to complete the function of video and image data acquisition [3, 24].

2) Embedded system platform: the embedded platform is the Linux operation system, which specifically implements the functions of image and video data
collection and transmission, and is applied in embedded application software development and programming.

(3) Embedded web server: this system uses Boa as an embedded web server, which only needs to be properly configured and transplanted into the whole system. As a compact and efficient web server, Boa supports CGI, which is an important basis for the system client to monitor real-time video by the browser.

(4) Client: in the same LAN, any authorized client, such as PC, mobile device and so on, can be accessed and managed. The client part mainly realizes the real-time reception of image data and displays it on the browser.

2.2. **Linux device driver.** In the embedded Linux remote video monitoring system, the Linux device driver is very important, and it is the interface between the kernel and the hardware [4, 12]. The Linux device driver abstracts the specific hardware equipment, and uses the standard system call interface to complete the specific operation of the hardware, such as writing, control, etc. The device driver implements the call function in the system. The architecture of Linux device driver of the embedded Linux remote video monitoring system is shown in Figure 2.

Linux device driver is mainly to complete the following functions: the hardware initialization and release; data transmission from the Linux kernel to the hardware to read; getting the communicate data between application program and hardware file; detecting and processing hardware errors. The device driver has the following features:

(1) Kernel code: when the driver is wrong, it can cause the system to be unable to start or the system will collapse directly.

(2) Kernel interface: providing a standard interface for the Linux kernel or its subsystem.

(3) Mechanism and service of Kernel.

(4) Loadable: most device drivers can be manually loaded into the kernel when needed and uninstall from the kernel when they are not needed.

(5) It can be set up when the system is compiled to implement the custom Linux kernel [1].
2.3. **USB camera’s driven loading.** The USB driver is mainly composed of the USB master controller, the USB core driver and the USB device driver. The hierarchical structure of the USB subsystem is shown in Figure 3.

In the embedded Linux remote video monitoring system, Vimicro 301V is used as the USB camera, of which the dynamic loading mode is used for driver module, to complete configuration, and then re-compile and generate kernel file to be downloaded to the target board. The loading work driven by the USB camera in this system is completed, and the images collected by the USB camera are preprocessed.

Preprocessing is an operation on an image at the lowest level of abstraction, at this time, the input and output of the processing are grayscale images. These images are similar to the original data taken by the sensor, usually a gray image represented by the value matrix of the image function. Preprocessing does not
increase the amount of information of the image. If the information is measured by the entropy, preprocessing generally reduces the amount of information in the image. Preprocessing is very useful in many cases, which helps to suppress special image processing or to analyze information that is unrelated to tasks. The purpose of preprocessing is to improve image quality, suppress unnecessary deformation or enhance important image features for subsequent processing, and improve the speed of image data transmission.

The image collected in embedded Linux remote video monitoring system can be defined as a two-dimensional function $f(x, y)$, where $x$ and $y$ represent the location of a coordinate point in XY of two-dimensional space, and the magnitude $F_{xy}$ of any pair of spatial coordinates $(x, y)$ is called the intensity or grayscale of the point image.

For the digital image $F_{xy}$ with the field of view $M \times N$, the mathematical model is as follows:

$$F_{xy} = \begin{bmatrix} F_{xy}(1, 1) & \cdots & F_{xy}(1, j) & \cdots & F_{xy}(1, N) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ F_{xy}(i, 1) & \cdots & F_{xy}(i, j) & \cdots & F_{xy}(i, N) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ F_{xy}(M, 1) & \cdots & F_{xy}(M, j) & \cdots & F_{xy}(M, N) \end{bmatrix} \quad (1)$$

Where, $F_{xy}(i, j) \in (1, 2^L)$ represents a gray level data which coordinate is $(i, j)$ pixel after digitization.

In the embedded Linux remote video monitoring system, nonlinear transmission function is adopted to process the image. The range of nonlinear transmission function processing information is large, which can solve the problem of small signal on both sides and the low gain region extended to both sides to stimulate large signal.

The tangent direction of gray value of the pixel point $(i, j)$ in the arbitrary coordinate $(i, j)$, the range of the value is $[0, \pi]$. In general, the continuous tangent direction is quantized with equal intervals, and the continuous direction is quantized to become 8 discrete equal spaced directions. The values set of the gray tangent direction of the quantized pixel point $(i, j)$ is as follows:

$$Q = \left\{ 0, \frac{\pi}{8}, \frac{3\pi}{8}, \frac{5\pi}{8}, \frac{3\pi}{4}, \frac{7\pi}{8} \right\} \quad (2)$$

A window with a pixel $f(i, j)$ as the center and a size of $n \times n$. In order to determine the direction of gray tangent of $f(i, j)$, the mean variance of the difference between the gray value of each point in the direction line and the difference of the mean value of the line’s gray level is calculated. The smaller the calculation results are, the less obvious change of the gray level in the direction line does. According to the definition of tangent in mathematics, the tangent direction of the gray value of the current point $f(i, j)$ should be closest to the direction of the minimum value of the mean square deviation, so the tangent direction $D_{\tan}$ of the gray value is defined as:

$$C^{D_{\tan}}_{ij} = \min[C^d_{ij}], d = 0, 1, \ldots, 7 \quad (3)$$

In the formula, $d$ is the direction number, and $C^d_{ij}$ is the mean variance between the gray value of each point in the direction line $d$ and the gray mean value of the gray line.
In order to be more accurately defined, the direction $d'$ vertical to $d$ should be considered:

$$d' = (d + 4) \mod 8$$  \hspace{1cm} (4)$$

The gray value tangent direction of the current point $f(i, j)$ can also be defined as: if $\prod_{d=0}^{7} C_{ij}^d \neq 0$, then:

$$C_{ij}^{D_{tan}} / C_{ij}^{D'_{tan}} = \min\{C_{ij}^d / C_{ij}^{d'}, d = 0, 1, \ldots, 7\}$$  \hspace{1cm} (5)$$

To sum up, if $\prod_{d=0}^{7} C_{ij}^d \neq 0$, the formula (5) is used to determine the tangent direction of the current point, otherwise the formula (3) is used to determine the tangent direction of the gray value of $f(i, j)$.

Image preprocessing is image compression. The analog data of the CMOS image sensor is 10bit through the A/D converter. The data of the embedded Linux remote video monitoring is 8bit. In order to ensure the matching of data bits, the data of 10bit is transformed to 8bit.

There are two basic types of transformation, one is to take $2 \sim 9$ bit directly, which is lower than that of 0, which is higher than that of 255. This method is a simple linear processing, which is used to calculate the gray level of the image with the pixel of $x$ after transformation.

Another one is hyperbolic transformation, which is an ideal nonlinear transmitted transformation which accords with image characteristics. According to the minimum variance principle of the gray value between each point and the gray value of the line, it can determine the tangent direction of gray value. The following is the formula for hyperbolic transformation:

$$y = \begin{cases} 
0 & 0 \leq x < 32 \\
0.722 \ln(x) - 304.443 & 96 \leq x < 1023 \\
80 + \frac{x}{4} & 512 \leq x \geq 256 \\
6 + \frac{x}{2} & 256 \leq x \geq 128 \\
x - 32 & 128 \leq x \geq 32 \\
134 + \frac{x}{8} & 1024 \leq x \geq 512 \\
70 + \frac{x}{4} & 512 \leq x \geq 256 \\
32 & 32 \leq x \geq 0 
\end{cases}$$  \hspace{1cm} (6)$$

Considering the rule of data bit conversion and the characteristics of shift register, the approximate formula is obtained by calculation.

$$y = \begin{cases} 
134 + \frac{x}{8} & 1024 \leq x \geq 512 \\
70 + \frac{x}{4} & 512 \leq x \geq 256 \\
6 + \frac{x}{2} & 256 \leq x \geq 128 \\
x - 32 & 128 \leq x \geq 32 \\
0 & 32 \leq x \geq 0 
\end{cases}$$  \hspace{1cm} (7)$$

The function is a simple selector, and then the displacement operation can be achieved.

2.4. **Specific implementation of video acquisition.** The Linux system treats all devices as files for unified operation, and the files corresponding to the device are called device files. The Linux device driver abstracts the specific hardware devices, and completes the specific operation of hardware device, such as reading, writing and controlling by the standard system calling interface API. Figure 4 is a flow chart of video capture.

There is noise in the collected remote video, and the fractal wavelet of multivariate statistical model is used to denoise the collected images. The specific algorithm is as follows:
$I_A$ is a natural image without noise, and $I_B$ is a noisy image. The relationship between them can be expressed as a formula (8).

$$I_B = I_A + \sigma^2 C$$

Where,  \( C \) means zero mean Gauss white noise,  \( C \sim N(0,1) \); \( \sigma^2 \) is the noise variance.

After the multi-resolution fractal wavelet decomposition of the noise image $I_B$, the $i$ th horizontal wavelet coefficient $y_{hi,j}$, the vertical wavelet coefficients $y_{vi,j}$ and the diagonal wavelet coefficients $y_{di,j}$ in $j$th layer are obtained. By the linear relation of the wavelet transform, it can be obtained:

$$\begin{cases} 
  y_{hi,j} = x_{hi,j} + \sigma^2 z_{hi,j} \\
  y_{vi,j} = x_{vi,j} + \sigma^2 z_{vi,j} \\
  y_{di,j} = x_{di,j} + \sigma^2 z_{di,j} 
\end{cases}$$

Where, $x_{hi,j}^h, x_{vi,j}^v$ and $x_{di,j}^d$ represent the horizontal, vertical and diagonal wavelet coefficients of image $I_A$, respectively. $z_{hi,j}^h, z_{vi,j}^v$ and $z_{di,j}^d$ denote the horizontal, vertical and diagonal wavelet coefficients of noise $C$, respectively.

Set $x$ be a $d$ dimensional wavelet coefficient vector, $x = (x_1, \ldots, x_d)^T$. $x_1$ is the wavelet coefficient that must be considered during denoising, and $(x_2, \ldots, x_d)$ is the correlated wavelet coefficients that can be considered during denoising, such as neighborhood and father & son wavelet coefficients. In order to simplify the formula, the single subscript wavelet coefficients $x_k, y_k$, and $z_k$ are used to replace $(x_{hi,j}^h, x_{vi,j}^v, x_{di,j}^d), (y_{hi,j}^h, y_{vi,j}^v, y_{di,j}^d)$ and $(z_{hi,j}^h, z_{vi,j}^v, z_{di,j}^d)$ respectively. The wavelet coefficient vectors of the noise image and noise are $y$ and $z$, respectively.

$$y = x + \sigma^2 z$$
In the calculation process, the focus of attention is the estimated value of the unknown wavelet coefficient vector \( \hat{x} \). The estimation value calculation of \( \hat{x} \) is dependent on the wavelet coefficient vector \( y \) corresponding to the noise image \( I_B \). In this paper, the maximization probability \( p(x|y) \) of the maximum posterior probability operator’s is used to estimate \( \hat{x} \), which can be calculated by the formula (11).

Since \( p(y) \) in the formula (11) is only a known constant, it does not affect the calculation result. When the error of probability is the minimum, the best value of \( \hat{x} \) can be estimated by \( p(x|y) \) and the mean square \( p(x) \) of the distribution function of the statistical sample.

\[
\hat{x} = \arg \max_{x \in \mathbb{R}} \ln p(x|y) = \arg \max_{x \in \mathbb{R}} \ln p(y|x)p(x)/p(y)
\]

\[
= \arg \max_{x \in \mathbb{R}} [\ln p(y|x) + \ln p(x) - \ln p(y)] \arg \max_{x \in \mathbb{R}} [\ln p(y|x) + \ln p(x)]
\]

(11)

Since each vector of Gauss’s noise is distributed independently, \( p(y|x) \) satisfies the multivariate Gauss distribution \( N(0, \sum_z \sigma^2 I) \), so \( \ln p(y|x) \) can be calculated by the formula (12).

\[
\ln p(y|x) = \ln \left( \frac{1}{(2\pi)^{d/2} |\sum_z|^{1/2}} \exp \left( -\frac{(y-x)^T \sum_z^{-1} (y-x)}{2} \right) \right)
\]

\[
= -\frac{d}{2} \ln(2\pi)^{d/2} - \frac{(y-x)^T (y-x)}{2\sigma^2}
\]

(12)

Set up a suitable statistical model for \( p(x) \). The wavelet coefficients of the sample images are detected, and they are found to be approximate to the Gauss distribution. The best model of Gauss mixed model is established. The MGGD model of the extended GGD model can be expressed as:

\[
p(x) = \varsigma \exp \left\{ -\left[ \frac{(x-\mu)^T \sum^{-1} (x-\mu)}{\alpha} \right]^\beta \right\}
\]

(13)

Among them, \( \alpha \) and \( \beta \) are spherical parameters of the model, and \( \varsigma \) is the normalization constant of \( \alpha, \beta \) and covariance matrix \( \sum_z \). \( p(x) \) in the formula (11) is defined as an unknown function \( f(x) \), obtained by formula (11) and (12).

\[
\hat{x} = \arg \max_{x \in \mathbb{R}} \left[ -\frac{d}{2} \ln(2\pi\sigma^2) - \frac{(y-x)^T (y-x)}{2\sigma^2} + f(x) \right] = \arg \max_{x \in \mathbb{R}} F(x)
\]

(14)

Among them, \( F(x) \) is part of the square brackets. It is assumed that \( F(x) \) is continuous and differentiable, if \( \hat{x} \) meets \( F(\hat{x}) > \lim_{x_1 \to \infty} F(x) \), then the maximization of \( F(x) \) can be replaced by formula (15) to calculate.

\[
\nabla F(\hat{x}) = \frac{\partial F(x)}{(x)} = 0
\]

(15)
Formula (15) and (14) can be used to reduce.

\[
\begin{cases}
\nabla F(\hat{x}) = -\frac{\bar{x} - y}{\sigma^2} + \nabla f(\hat{x}) = 0 \\
\hat{x} = y + \sigma^2 \nabla f(\hat{x})
\end{cases}
\]  

(16)

Supposing \( \mu = 0 \), the MGGD model is used to get a more explicit calculation of the formula (16):

\[
\nabla f(x) = -\frac{2\beta}{\alpha^2} \left( x^T \sum_{x} x \right)^{-1} \sum_{x} x
\]  

(17)

The formula (16) and (17) can be obtained:

\[
\hat{x} = \sum_{x} y \left[ \sum_{x} + \frac{2\sigma^2\beta}{\alpha^2} \left( \hat{x}^T \sum_{x} \hat{x} \right)^{-1} \right] (I)
\]  

(18)

In order to solve the problem that is not solved in formula (18), the purpose of solving this problem can be achieved by defining \( \alpha, \beta \) and covariance matrix \( \sum_{x} \) as special values or values.

Multivariate statistical models are used to analyze the distribution of wavelet coefficients after fractal wavelet transform, and make full use of the local features of the images to improve the filtering effect of the filter, and the fine structure and edge information of the images are preserved.

3. Experimental results and analysis. The hardware platform of the embedded Linux remote video monitoring system for the Internet of things is illustrated in Figure 5.

After the completion of the platform, the video monitoring system’s near end and the effect of remote video monitoring are tested and analyzed, which prove the overall performance of the embedded Linux remote video monitoring system. The embedded Linux remote video monitoring system, remote video monitoring system based on LabVIEW and the remote video monitoring system based on GPRS are used to test. In the embedded Linux remote video monitoring system, remote video

![Figure 5. hardware development platform](image-url)
monitoring system based on LabVIEW and remote video monitoring system based on GPRS, interference signals are added, and the anti-interference performance of the three different methods are compared. The test results are as shown in Figure 6.

Figure 6 (a) is a test result of a remote video monitoring system based on GPRS. In the remote video monitoring system based on GPRS, interference signals are added. Comparing the signal frequency of the system with adding interference signals and without adding interference signals, we can see that in the remote video monitoring system based on GPRS after adding interference signals, the fluctuation of signal frequency is large, which proves that GPRS based remote video monitoring system has poor anti-interference ability. Figure 6 (b) is a test result for a remote video monitoring system based on LabVIEW. Interference signals are added to the remote video monitoring system based on LabVIEW. Comparing the signal frequency of the system with adding interference signals and without adding interference signals, we can see that in the remote video monitoring system based on LabVIEW after adding interference signals, the fluctuation of signal frequency is large, which proves that LabVIEW based remote video monitoring system has poor anti-interference ability. Remote video monitoring system based on GPRS and LabVIEW are influenced by the great interference signal, and the collected video is blurred or spotted. Figure 6 (c) is the test result obtained by using an
embedded Linux remote video monitoring system. Comparing the signal frequency of the system with adding interference signals and without adding interference signals, we can see that in the embedded Linux remote video monitoring system after adding interference signals, the fluctuation of signal frequency is small, which proves that embedded Linux remote video monitoring system has strong anti-interference ability and the clear video and image can be obtained.

In order to further verify the performance of the embedded Linux remote video monitoring system, the embedded Linux remote video monitoring system is tested. The test pen as the target is monitored by Vimicro 301V camera to real-time display and control. The aim of the experiment is to move the monitored target pen to the center of the monitoring field by clicking on the azimuth motion button of the embedded Linux remote video monitoring system. Using the lens adjustment button, the monitored target pen will be amplified to clearly observe the details of the monitored target. The monitoring effect of the embedded Linux remote video monitoring system is shown in Figure 7.

Figure 7 (a) is the initial screen. Through clicking on the azimuth movement button of the embedded Linux remote video monitoring system, the monitoring target can be moved to the monitoring center of the view, as shown in Figure 7 (b); the reuse of the lens adjustment button of embedded Linux remote video monitoring system, the target to be monitoring is amplified as shown in Figure 7 (c); continue to further enlarge the monitoring target, it can see further details of the target, shown in Figure 7 (d), indicating that the embedded Linux remote video monitoring system can get a clear picture of the monitor.

The embedded Linux remote video monitoring system, LabVIEW based remote video monitoring system and GPRS based remote video monitoring system are used to test. The data transmission time of three different methods is tested, and the system responsiveness is compared. The result of the test is shown in Figure 8.

Figure 7. system monitoring images
(a) Test results of remote video monitoring system based on LabVIEW

(b) Test results of remote video monitoring system based on GPRS

(c) Test results of embedded Linux remote video monitoring system

Figure 8. The transmission of data by three different methods
Figure 8 is the transmission data in the same time by three different ways based on embedded Linux remote video monitoring system, LabVIEW based remote video monitoring system and GPRS based remote video monitoring system. Figure 8 (a) is the data transmitted over a certain time by a remote video monitoring system based on LabVIEW. Figure 8 (b) is the data transmitted over a certain time by a remote video monitoring system based on GPRS. Figure 8 (c) is the data transmitted over a certain time by the embedded Linux remote video monitoring system. Compared with figure 8 (a), figure 8 (b) and figure 8 (c), we can see that the amount of data transmitted by embedded Linux remote video monitoring system is more than that of remote video monitoring system based on LabVIEW and GPRS based remote video monitoring system, indicating that the efficiency of transmitting data by the embedded Linux remote video monitoring system is high, and the response of the system is good.

4. Conclusions. The advantages of embedded system lie in its excellent performance, strong network support, flexibility and simplicity in program migration, and effective cost control. Therefore, the application of embedded system in network video monitoring has highlighted these unique advantages. In this paper, the development technology of embedded system are deeply studied and analyzed, the complete software and hardware platform of embedded system development are constructed, and the remote video monitoring system based on embedded Linux is designed and completed. The following works are mainly completed:

(1) The development course of video monitoring system, the relevant situation at home and abroad as well as the future development trend are summarized. In this environment, the related knowledge of embedded technology is introduced, which lays a foundation for the research and completion of this topic.

(2) On the basis of understanding the relevant embedded knowledge, we compare and select the hardware and software development platform of the whole system, define the overall architecture of the system, and establish the overall design of embedded Linux remote video monitoring system.

(3) After studying the working principle of the device driver, the loading of the 301V camera driver is completed.

(4) The algorithm of image preprocessing and image denoising is studied, and the network transmission of video acquisition data and the real-time monitoring of remote video are realized.

REFERENCES

[1] J. E. Banchs and D. L. Scher, Emerging role of digital technology and remote monitoring in the care of cardiac patients, Medical Clinics of North America, 99 (2015), 877–896.
[2] D. G. Dietlein, A method for remote monitoring of activity of honeybee colonies by sound analysis, Journal of Apicultural Research, 24 (1985), 176–183.
[3] O. Frederik, J. R. Danaraj, B. Fleet, H. Gunasingham, S. Jaenicke and V. E. Hodgkinson, Remote monitoring and control of electrochemical experiments via the internet using “intelligent agent” software, Electroanalysis, 11 (2015), 1027–1032.
[4] F. Gonze, R. M. Junger and A. N. Trahtman, A note on a recent attempt to improve the pin-frankl bound, Behavioural Brain Research, 17 (2015), 307–308.
[5] N. J., N. F., B. S. and et al, Biometric recognition in monitoring scenarios: A survey, Artificial Intelligence Review, 515–541.
[6] S. K., Z. Y., Z. G. and et al, Long-term remote monitoring of total suspended matter concentration in lake taihu using 250 m modis-aqua data, Remote Sensing of Environment, 62 (2015), 43–56.
[7] L. M. Kallinen, R. G. Hauser, C. Tang, D. P. Melby, A. K. Almquist, W. T. Katsiyiannis and C. C. Gornick, Lead integrity alert algorithm decreases inappropriate shocks in patients who have sprint fidelis pace-sense conductor fractures., *Heart Rhythm*, 7 (2010), 368–377.

[8] B. Katalenich, L. Shi, S. Liu, H. Shao, R. Mcduffie, G. Carpio, T. Thethi and V. Fonseca, Evaluation of a remote monitoring system for diabetes control, *Clinical Therapeutics*, 37 (2015), 1216–1225.

[9] R. Kosaka, Y. Sankai, R. Takiya, T. Jikuya, T. Yamane and T. Tsutsui, Tsukuba remote monitoring system for continuous-flow artificial heart., *Artificial Organs*, 27 (2015), 897–906.

[10] N. Kumar, J. H. Lee and J. J. P. C. Rodrigues, Intelligent mobile video surveillance system as a bayesian coalition game in vehicular sensor networks: Learning automata approach, *IEEE Transactions on Intelligent Transportation Systems*, 16 (2015), 1148–1161.

[11] X. L., X. H. K., H. X. and et al, Remote video monitoring optimization method research based on internet of things, *Environmental Earth Sciences*, 72 (2015), 226–228.

[12] T. Lewalter and T. Brodherr, Remote monitoring of implantable cardioverter-defibrillators: Financial impact for providers and benefits to patients, *European Heart Journal*, 36 (2015), 143.

[13] N. Liu, W. Chen, Q. Wang and Y. Lang, Remote video monitoring and early warning system based on android platform, *Journal of Jilin University*, 283–288.

[14] B. M. A., L. S. R., C. M. J. and et al, Eight-week remote monitoring using a freely worn device reveals unstable gait patterns in older fallers, *IEEE Transactions on Biomedical Engineering*, 27 (2015), 2588–2594.

[15] N. Parthiban, A. Esterman, R. Mahajan, D. J. Twomey, R. K. Pathak, D. H. Lau, K. C. Robertsthomson, G. D. Young, P. Sanders and A. N. Ganesan, Remote monitoring of implantable cardioverter-defibrillators: A systematic review and meta-analysis of clinical outcomes., *Pacing Clin Electrophysiol*, 27 (2015), 2591–2600.

[16] K. Q. and G. B., Remote virtual supervision system, *European Journal of Soil Science*, 79–89.

[17] E. S. Ramirez, D. Luis and E. Perez, Web monitoring module for video surveillance system xilema suria, *Bulletin of the American Meteorological Society*, 91 (2015), 1699–1701.

[18] H. U. Rong, X. Q. Luo and H. E. Shang-Ping, Simulation study on the human motion characteristics monitoring of remote video image, *Computer Simulation*, 298–301.

[19] F. H. Tsai, An investigation of gender differences in a game-based learning environment with different game modes., *Eurasia Journal of Mathematics Science & Technology Education*, 13 (2017), 3209–3226.

[20] N. Varma, J. P. Piccini, J. Snell, A. Fischer, N. Dalal and S. Mittal, Relationship between level of adherence to automatic wireless remote monitoring and survival in pacemaker and defibrillator patients, *Journal of the American College of Cardiology*, 65 (2015), 2601–2610.

[21] J. Wang, B. Tian, J. Lu, D. Macdonald, J. Wang and D. Luo, Renewable-reagent enzyme inhibition sensor for remote monitoring of cyanide, *Electroanalysis*, 10 (2015), 1034–1037.

[22] Y. G. Wang, The effect of reservoir projects on the household income of indigenous people an empirical analysis based on gangkouwan reservoir project, *Journal of Interdisciplinary Mathematics*, 20 (2017), 195–207.

[23] L. I. Xiao-Hui, Research on the open architecture of iot, *Journal of China Academy of Electronics & Information Technology*, 478–482.

[24] H. Yang, X. Liu and L. Zhang, Observer-based tracking control using unmeasurable premise variables for time-delay switched fuzzy systems, *Journal of Intelligent & Fuzzy Systems*, 32 (2017), 3973–3985.

Received August 2017; revised January 2018.

E-mail address: 18603523830@163.com
E-mail address: bidyut@cs.siu.edu