

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Development and Implementation of Intelligent Video Surveillance Alarm System

^{1,2}Ruogu Zhou, ¹Feng Ding and ¹Li Lu

¹School of Computer, National University of Defense Technology, Changsha, 410073, China

²Department of Information Technology, Hunan Vocational College of Commerce, Changsha, 410205, China

Abstract: There exist many deficiencies in current video surveillance system such as high expenditure, information omitting, data analysis difficulty, long response. This study proposes a design scheme of intelligent alarm system with the main technology of motion detection, image compression and WEB server pre-fetch which improves the algorithm based on differential image motion detection, controls the costs and reduces false positives. The system features as low cost but fast response with improvement of Loeffler algorithm and the quality of image compression while the decreasing of storage space and transmission bandwidth. It also improves the intelligent Wcol prediction algorithm, predicts the page that user will read soon and eventually improves the efficiency of WEB server. Experiments show that: this system reduces the false alarm rate, avoids filtering, has certain adaptability to light changes, reduces or even eliminates the interference of the CCD jitter, can accurately extract the moving object location. The total time of compressing 100 BMP images and completing the USB storage is lower than 48 sec. The performance that the server with pre-fetch module visits the next page at the second time is better than the performance that the server with no pre-fetch module visits the next page at the second time.

Key words: Home security, monitor, motion detection, image compression, prediction algorithm

INTRODUCTION

Today video surveillance systems show two main drawbacks, namely: they are not flexible in adapting to different operative scenarios (they only work in a well known and structured world); they generally need the assistance of a human operator in order to recognize and to tag specific visual events (Wong *et al.*, 2009). Because humans being can not serve as fully trusted observer, this model itself may lead to omission phenomenon (Hapsari and Prabuwno, 2010). The need of people to judge against threat will cause a delay of security alerts and too long response time of security threats. Mass storage of data also increases the cost of video surveillance.

For these shortcomings of video surveillance systems, many new technologies are proposed. Mecocci *et al.* (2003) devised an automatic real-time video surveillance system, capable of autonomously detecting anomalous behavioral events. The proposed system is capable of automatically adapting to different scenarios without any human intervention and uses robust self-learning techniques to automatically learn the typical behaviour of the targets in each specific operative

environment. A wireless video surveillance system supporting wide area access based on 3G network is proposed (Ruichao *et al.*, 2009). Guo *et al.* (2010) implements and analyzes several commonly used methods for moving object detection in video surveillance system, such as temporal difference, median filtering and Single Gaussian model. Wong *et al.* (2009) proposed a surveillance system, the Omni directional scenes within a digital home are first captured using a web camera attached to a hyperbolic optical mirror. The captured scenes are then fed into a laptop computer for image processing and alarm purposes. Log-polar mapping is proposed to map the captured Omni directional image into panoramic image, hence providing the observer or image processing tools a complete wide angle of view. It can cover a wide angle of view (360deg Omni directional), small in size, low cost, lightweight, output images are with higher data compression and accurate trespasser detection. Moon and Pan (2009) analyses conventional methods of privacy protection in surveillance camera systems and applied scrambling and RFID system to existing surveillance systems to prevent privacy exposure in monitoring simultaneously for both privacy protection and surveillance. A new approach is proposed for a smart

video surveillance system by Xu and Wu (2010). The proposed surveillance system uses canonical stereo configuration to set up a pair of static cameras to support a salient map to control a Pan-Tilt-Zoom (PTZ) camera/cameras, which captures high definition image/video of the interesting moving object in the surveillance area for further forensic investigations. The other contribution of the proposed approach is that, 3D information generated by the set of static cameras is used to support reliable spatial-temporal based image segmentation and object detection.

The system which has been developed by us is characterized by its smart, convenience and power, which can process images in real time with its motion detection, thus eliminate the need to monitor staff. Its auto-recording of scene and efficient image data backup provides users with clear and reliable picture.

Software architecture of intelligent video surveillance alarm system: The hardware platform of the intelligent video surveillance alarm systems involves ARM9 2410 processor and OV7620 + OV511 image acquisition card. Linux operating system is its software platform.

System software modules are divided into digital image acquisition module, image control detection module, image compression module, memory module and WEB modules. The overall framework of the system software is shown in Fig. 1. Image control detection, image compression, WEB pretreatment is the key to implementation of the system. Essential parts of the overall framework of the system which is shown in Fig. 1, include image control detection, image compression and WEB pretreatment.

The improved algorithm of differential image motion detection: The purpose of motion detection is to detect the video image sequence and to extract the moving target (Collins *et al.*, 2000). Currently, there are some methods for motion detection such as optical flow which is poor at

real-time control and with high time expense, noise-Gaussian-distribution-based methods with a large amount of calculation and the method of difference image, which is relatively simple and with strong real-time control (Seferidis and Ghanbari, 1993; Kim *et al.*, 1999; Hongjian, 2009).

Generally there are two kinds of difference image method one is the difference between the current continuous two images (continuous image difference method) another is the difference between the current image and the fixed background image (the fixed background motion detection). The disadvantage of the continuous image difference method is that the detected location of objects is inaccurate and the detected target is much larger than the real one. The fixed background motion detection is performed as follows: firstly the background image is stored, then the difference image by the current image and the background image is read, finally the value of each pixel and a pre-sets threshold is compared. If a pixel value of background is greater than the pre-sets threshold, it will be considered as the pixel of the moving objects and if not, it will be background point. Since this technology aims to get current image and there only exists the moving object in the difference image, which is the subtract of current frame and fixed background frame.

The new motion detection algorithm based on the difference image method introduces the block thought, determines the detection threshold with the method based on noise Gaussian distribution method. Compared with traditional rule of thumb threshold method, it is more accurate and more reliable since this method joins the brightness adjustment mechanism and reduces the light changes. Specific methods are as follows: firstly, the image is reasonably divided to blocks which are the basic units of motion detection; secondly, the mean gray value of the current frame image block and the mean gray value of fixed background image block are compared; thirdly, the method based on noise Gaussian distribution method

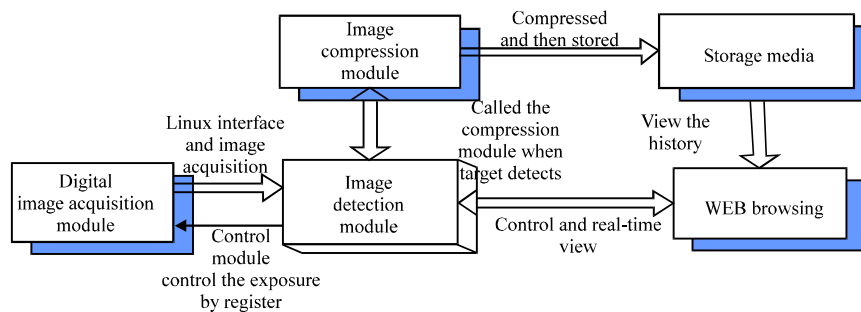


Fig. 1: Intelligent video surveillance alarm system software architecture

to get the detection threshold and the threshold to determine whether there is moving objects in the image are applied which aid to extract the blocks where the moving objects are in. All the operations are prepared for the follow-up image compression. A brightness adjustment mechanism for the current frame is added to alleviate the impact of ambient light in motion detection which can effectively reduce or even eliminate the impact of illumination change. With the image data of a new frame, it can adaptively change the average brightness of the reference frame pixel.

The establishment of the fixed background model: At the beginning of motion detection, a pre-stored background image with no moving target constitutes the initial model. The following method to extract background model was conducted the first two consecutive frames with the difference method to determine whether there is moving objects. If there is no, the average gray values of N consecutive frames would be calculated out and one frame as the reference frame would be directed.

Since real time of motion detection algorithm is of great significance, the low cost computation algorithm is taken as the primary consideration. All moving objects have their size and the pixel brightness of each object has a certain similarity too. So there is no need to compare the image brightness of each pixel value. The image is divided into blocks whose size is similar to the objects to be detected. This method will improve the real time of motion detection.

In this system, scene images are divided into blocks with sizes about 1/4 of the body (i.e., 320×240, image is divided into 8×8, a total of 64), in order to avoid misjudgment of small objects. Selecting about 1/4 size of the human body can distinguish the movement of large objects and the motion of small size objects.

Specific identification process is as follow: when the average brightness value of a block changes was detected and the threshold was exceeded, the average brightness value of the adjacent new blocks was detected to determine whether their average brightness values exceed the threshold. Provided that the average brightness value of the new block exceeds the threshold, adjacent block based on this new block would be tested. Only when a total of four or more blocks that their average brightness exceeds the threshold were found, moving object would be identified as large-size objects.

The mechanism and experimental analysis of brightness adjustment: Changes of ambient light exert impact on fixed background motion detection. In the determination of conditions, a light-sensitive item was added. The improved conditions are as follow:

$$\text{Max} \left| \frac{1}{n} \sum_{x \in a_i} S_c(x, t + \Delta t) + \lambda \frac{1}{N} \sum_{x \in A} [S_c(x, t) - S_c(x, t + \Delta t)] - \frac{1}{n} \sum_{x \in a_i} S_c(x, t) \right| > T$$

and a_i is a piece of the overall image A, the total pixel number is n:

$$\frac{1}{n} \sum_{x \in a_i} S_c(x, t + \Delta t)$$

is the average brightness of all pixels in $t + \Delta t$ time for the a_i block:

$$\frac{1}{n} \sum_{x \in a_i} S_c(x, t)$$

is the average brightness of all pixels in t time for the a_i block:

$$\lambda \frac{1}{N} \sum_{x \in A} [S_c(x, t) - S_c(x, t + \Delta t)]$$

is the differences brightness of image A in time $t + \Delta t$ and time t and the value can be positive or negative. N is the number of all pixels, the value of inhibition coefficient λ can be 1.

Two pictures are taken in the same scene but under different light intensity in experiment. The 2nd and 3rd column of Table 1, respectively showed the average brightness value of each divided blocks of two scenes. Data in column 2 is obviously lower than it in column 3 because of light difference. The 4th column indicated the difference value of each block between the 2nd column and the 3rd column.

In addition, the average brightness value of the whole dark light scene is 119.365257 and that of the whole light brightness scene is 146.048981. In 4th column of Table 1, the value of the 48th-block is 7, the value of the 56th-block is -5, so moving objects are in these blocks, we can not consider these blocks. Taking dark light scene as the reference frame, the light brightness scene as the detected frame, according to algorithm, the difference value between 119.365257 and 146.048981, takes -27 must be added to the average brightness of light brightness scene. The adjusted difference value of average brightness was showed on the 5th column of Table 1.

Thus, the adjusted difference values of average brightness are between -9 and 10. In other words, provided that the threshold value T is greater than 10, the changes of light intensity will not affect the results of motion detection. It shows that this improved algorithm has certain adaptability to brightness variations.

Table 1: The average brightness value test

Block No.	The average brightness value of dark light scene	The average brightness value of light brightness scene	The difference value between dark light scene and light brightness scene (rounding)	The adjusted mean brightness	The average brightness value of moving object picture	The adjusted average value of moving object picture	The difference value of average brightness value between column 7 and column 2 (Rounding)
0	161.99	195.09	33	6	169.29	167.39	5
1	143.46	174.66	31	4	148.96	147.06	4
2	174.90	203.91	29	2	180.42	178.52	4
3	150.01	180.74	31	4	155.10	153.20	3
4	137.61	168.05	30	4	142.65	140.75	3
5	137.22	168.28	31	4	143.96	142.06	5
6	127.17	159.41	32	5	130.71	128.81	2
7	152.69	185.71	33	6	157.04	155.14	2
8	192.13	229.57	37	10	204.35	202.45	10
9	197.93	230.51	33	5	209.53	207.63	10
10	229.23	239.07	10	3	234.93	233.03	4
11	182.48	212.75	30	3	191.34	189.44	7
12	154.87	192.15	37	10	160.50	158.60	4
13	172.29	207.01	35	8	161.92	160.02	-12
14	114.47	151.51	37	10	123.37	121.47	7
15	136.57	174.32	38	11	144.37	142.47	6
16	163.75	194.17	30	3	168.30	166.40	3
17	216.10	233.70	18	-9	219.49	217.59	1
18	217.66	235.32	18	-9	221.47	219.57	2
19	213.26	235.40	22	-5	217.63	215.73	2
20	206.06	233.69	28	1	169.86	167.96	-38
21	184.72	206.64	22	-5	102.98	101.08	-84
22	118.46	151.25	33	6	122.41	120.51	2
23	145.55	178.86	33	6	148.23	146.33	1
24	94.60	127.93	33	6	100.21	98.31	4
25	94.48	127.47	33	6	100.03	98.13	4
26	98.83	128.92	30	3	102.81	100.91	2
27	114.93	142.19	27	0	120.10	118.20	3
28	142.82	170.68	28	1	112.42	110.52	-32
29	138.55	167.25	29	2	103.82	101.92	-37
30	105.91	139.73	34	7	111.26	109.36	3
31	148.42	183.10	35	8	152.31	150.41	2
32	92.26	127.28	35	8	126.85	124.95	33
33	89.79	124.61	35	8	97.59	95.69	6
34	58.26	83.17	25	-2	60.81	58.91	1
35	38.62	57.24	19	-8	43.49	41.59	3
36	56.47	81.78	25	-2	62.61	60.71	4
37	75.55	102.77	27	0	80.91	79.01	3
38	72.43	102.48	30	3	80.19	78.29	6
39	88.97	122.03	33	6	97.22	95.32	6
40	141.02	172.51	31	4	126.40	124.50	-17
41	126.03	155.98	30	3	103.75	101.85	-24
42	76.99	99.51	23	-4	76.93	75.03	-2
43	50.15	69.05	19	-8	53.89	51.99	2
44	45.41	63.47	18	9	48.78	46.88	1
45	66.76	88.84	22	-5	69.12	67.22	0
46	69.49	93.34	24	-3	74.12	72.22	3
47	106.12	135.22	29	-2	112.42	110.52	4
48	134.07	140.85	7	1	124.01	122.11	-12
49	121.04	142.07	21	-6	117.87	115.97	-5
50	120.09	141.88	22	-6	124.32	122.42	2
51	95.83	118.65	23	-4	102.34	100.44	5
52	117.62	145.71	28	1	126.23	124.33	7
53	109.57	137.73	28	1	117.49	115.59	6
54	127.57	162.09	35	8	136.86	134.96	7
55	121.83	155.61	34	7	130.20	128.30	6
56	87.33	82.48	-5	1	111.37	109.47	22
57	29.29	47.62	18	-9	44.22	42.32	13
58	51.23	68.64	17	-10	56.07	54.17	3
59	65.85	87.64	22	-5	70.62	68.72	3
60	83.99	109.77	26	-1	89.50	87.60	4
61	62.86	82.97	20	-7	66.74	64.84	2
62	41.21	58.14	17	-10	46.55	44.65	3
63	43.38	60.58	17	-10	49.19	47.29	4

Threshold determination and experimental analysis: The motion detection algorithm based on CCD noise characteristics can be viewed as a Gaussian distribution of deviations, its background noise can be regarded as Gaussian and the average value of various random electronic noises can be regarded as 0. Therefore, the problem of determining the moving object is transformed to find whether the difference image are wholly in the rule of Gaussian distribution.

According to the " 3σ " feature of Gaussian distribution, for Gaussian data, most of them distribute within the 3σ (where, σ is the standard deviation). We can set the detection threshold according to this feature and use it to determine whether the image blocks have moving objects.

In the initial motion detection, we estimate the average value of the difference image block and the initial value of variance, according to the difference image between two consecutive images and in the two consecutive images there are not any moving object. According to multiple difference images, we calculate the error between the brightness value of difference image and the estimated value. Finally, we use the iterative weighting method. By the way, the value of the standard deviation σ is determined by the physical properties of CCD, so when motion detection begins, we can determine the initial value of the standard deviation and set 3σ to be the threshold of motion detection.

Next, according to the determined threshold, we look at the algorithm's effect of moving object detection. The standard deviation is stably at about 4; take 12 as its threshold value. Consider dark light image to be the reference frame. We take a moving object picture in experiment scene. And the 6th column of Table 1 shows the average brightness value of the picture. The average brightness value of the whole image is 121.260445.

According to the reference frame, adjust the average brightness value of each block in moving object picture, that is every block in moving object picture adds the difference value of the whole average brightness value between dark light scene and light brightness scene, i.e., $119.365257 - 121.260445 = -1.895188$, take -1.90 , as shown in the 7th column of Table 1.

It can be seen from the 8th column of Table 1: the 21st, 22nd, 28th, 29th, 32nd, 40th, 41st, 56th blocks change very apparently. Comparing moving object picture and dark light scene, obviously, there is a person, he caused a change in the first block of 21st, 22nd, 28th, 29th and the amplitude is relatively large. Look at the more obvious changes in the 32nd, 40th, 41st, 56th blocks. We find the regional block 32nd, 40th, 41st have a case, the first 56 regional block has a cloth.

According to the threshold, we can detect a moving object in the 21st, 22nd, 28th, 29th, 32nd, 40th, 41st, 48th, 56th, 57th blocks and it is consistent with the actual situation.

According to the experimental results, the 3σ threshold for motion detection can detect the movement of large objects but also can detect the movement of small objects. And 3σ threshold should be the minimum threshold; the movement of small objects is also very sensitive. The system is mainly for large objects, so the appropriate threshold value can be increased to 3.5 or 4σ .

THE OPTIMIZATION OF JPEG COMPRESSION ALGORITHM

Storage capacity and transfer rate is an important fact of cost and performance of the video surveillance system. In order to reduce the demand for network bandwidth and save storage space, the data video system captured need to compress (Logashammugam and Ramachandran, 2008). Compression is the most expensive part of system performance, we need design carefully. Compression module system is DCT-based. The compression process is irreversible. System uses less number of bits to get better quality restored image. The image is inputted into the encoder, through the DCT transforms, through quantization, through entropy encoded, at last compressed image is outputted.

The introduction of adaptive quantization table based on motion images: The final image compression quality and file size depend on the quantization table. Quantification is a one-to-many mapping. It is in accordance with the JPEG standard and uses a linear uniform quantizer (Han and Jie, 2007). The introduction of adaptive quantization table is to implement that: If no movement is detected, system uses the standard quantization table to loss more details. But once movement is detected, system uses optimized quantization table to retain more specific information.

Consider the complexity of the implementation and the restriction of the environment conditions, we use two set of brightness and color quantization tables to separately quantify the images with no motion and the image with motion. It not only effectively improves the quality of the detected movement image but also effectively reduces the size of the image with no movement. It also increases the number of stored images, and an implement simply, has low complexity.

The introduction of adaptive quantization table based on moving images: To the moving target and its background, we can also use the method of adaptive quantization tables. The basic idea is: we mark the results of block motion detection, according to the magnitude of each motion, adaptively change the number of quantization levels, quantify the independent part of the movement crudely and quantify the dependent part of the movement carefully. Ensuring the clarity of moving target, we also allow a small amount of movement independent blocks to have mosaic effect, thus we can speed up the compression rate and reduce the image size.

The implementation of this method based on motion detection, after the block motion detection of a picture, we refine the motion blocks and quantify the independent part of the movement crudely, desert the useless information, allow these blocks have mosaic appearance. The adaptive quantization table adopts two different quantization tables. It is contrary to the optimization of quantization table. It uses multiplying quantization table (the standard JPEG quantization table multiplied by 2) to quantify the independent part of the motion picture and makes more details to be lost, so as to reduce the size of the entire picture to improve compression ratio.

The comparison of two methods: The core of the two optimization methods is that they all use adaptive quantization table. The difference is the using level of the adaptive quantization table. Motion detection to the image with movement and the image with no movement use different quantization tables. It can effectively reduce the size of the picture which people is not interested and assure the quality of interested image. The adaptive quantization for different regions of each picture improves the efficient of image compression, highlights the goals and blurs unrelated areas. But the whole process of quantification is complex and for the performance of current development platform, the efficient will be greatly affected.

Image filtering: JPEG can compress "continuous" image well, if the image has many sudden changes in color brightness, JPEG can not work well (Haseeb and Khalifa, 2006). In view of this principle, the noise filtering of picture to a certain degree is necessary.

Median filters use nonlinear processing methods to remove noise which can be done to remove noise and also protect the edge of the image (Gonzalez and Woods, 2002). Median filtering algorithm in the system still uses a sliding window but modifies the image pixel value of the

window's center to the value in the middle of all the window's pixel values. As a result, the noise can be removed; system can better keep the edge information.

Currently, the median value of median filter is the middle value of all the elements after sorting. But sorting needs to move a large number of elements and its efficient is low. We just try to find the middle value, so we do not really have to sort. Use the quick sort to sort, when sorting fulcrum is in the middle of the original array, we can end the cycle. The average time complexity of quick sort is $O(n \log_2 n)$. Its space complexity is $O(\log_2 n)$.

After filtering, the image compression ratio has increased significantly, compared to the average, the JPEG compression of 24-bit BMP images increased by about 1%.

The DCT fast algorithm: The same JPEG image compression test to a 320×240 image, in the case of $Y_h: Y_v: U_h: U_v: V_h: V_v = 2: 2: 1: 1: 1: 1$, has six 8×8 information groups, corresponding to the sub-block of MCU is 16×16 , the operation number of DCT and IDCT are $320 \times 240 \times 6 / (16 \times 16) = 1800$ (Abboud, 2006). Therefore, it is necessary to use fast DCT algorithm to do some other improvements: (1) Similar to the YUV and RGB conversion, convert floating-point operations to fixed-point operations and pre-calculate the $\cos()$ values to prepare for calls. (2) 8×8 2-D DCT are converted to two 8-point 1-dimensional DCT. The formula is follow:

$$F(u, v) = \frac{1}{2} c(v) \sum_{j=0}^{N-1} \left[\frac{1}{2} c(u) \sum_{i=0}^{N-1} f(i, j) \cos \frac{(2i+1)u\pi}{2N} \right] \cos \frac{(2j+1)v\pi}{2N}, N=8$$

One-dimensional fast DCT algorithms are many (Feig, 1990; Vetterli, 1985), we use a smaller computational Loeffler algorithm (Loeffler *et al.*, 1989). The DCT algorithm for 16×16 blocks needs to do 31 times multiplications and 81 times additions. But the DCT algorithm for 8×8 blocks only needs to do 11 times multiplications and 29 times additions. The direct calculation requires 56 times multiplications, 56 times additions. Visibly fast algorithms save a lot of computation.

THE SERVER PRETREATMENT

The WEB server performance of monitoring system directly affects the speed of customer reading. To improve server performance, the system adds server page prediction modules.

The current algorithms are: referrer field method, PPM method, the prefetching method based on semantic

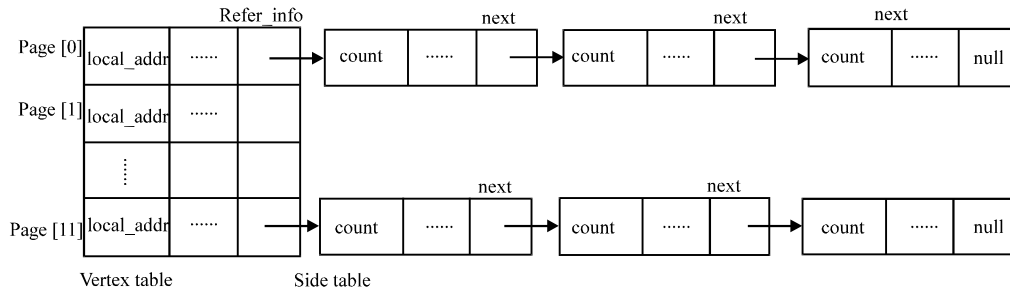


Fig. 2: The adjacent table of web site

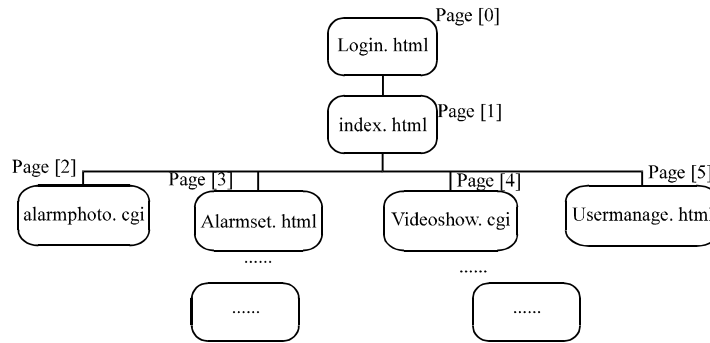


Fig. 3: The corresponding Fig. of the pages and the vertex table

understanding, Wcol method (Umapathi and Raja, 2006). Wcol method is very simple: if the URL user requested is a HTML document, the server will analyze the link above the embedded images and other HTML documents and then pre-fetch them and put them out to the cache.

In this system pages and links are simple. Therefore, we use Wcol intelligent prediction algorithm and do some improvements. The method considers the inter-linked pages as a graph and use adjacency list to store the graph.

Store the relevant pages of the site with a vertex table and then store the related links in the side table. When a user requests a URL, the server receives the request, parses the request URL, then pre-fetch module tries to find out from the adjacent list whether the page is in memory. If the page is not in memory, module goes to the disk to read the page and put it to memory. If the page is in memory, module directly read the page to user. At the same time, according to the information of this page which is found out from the adjacency table, the pre-fetch module finds the relevant links from side table and predicts the pages, images and other information that users may visit next, at last puts the information into memory to improve access speed.

Data structure: Pre-fetch works start around the adjacency list. All pages of the website constitute a vertex

table. The reference links of a page include page information and picture information and all of them constitute a side table. Adjacency list is shown in Fig. 2. The data structure is defined as follow:

```

typedef struct{
    Char local_addr[30]; //The local address
    int total_file; // The related Links of this page
    int in_off; // Whether the page is in memory or not
    hash_inf *refer_info; //The pointer of the side table
}page_inf;
page_inf page[15];
typedef struct node{
    int count; //Visit times
    char local_addr[30]; // Local access address
    int in_off; // Whether the page is in memory or not
    struct node *next; // Pointer to the next node
} hash_inf;
    
```

The development of web site is divided into static pages module and CGI program module. The web site totally has 12 pages. The page structure is shown in Fig. 3. This structure can be viewed as a tree. Therefore, the vertex table array page [15] stores the information of the total 12 pages. The storage order of these pages is under the level priority shown in Fig. 3.

Workflow: When server starts, system builds the data structure of adjacency list. Page [2] and page [3] store image-related pages. Associated with the operation of the

Table 2: The test results of the server with no pre-fetch module

The server with no pre-fetch module						
Time	The first test			The second test		
	Group 1	Group 2	Group 3	Group 1	Group 2	Group 3
The first loading time of picture						
5 sec	1	0	1	1	1	1
6 sec	3	5	4	3	3	4
7 sec	5	4	4	5	5	4
8 sec	2	3	2	2	2	2
9 sec	1	0	4	1	1	1
The average loading time	6.91	6.83	6.83	6.91	6.91	6.83

Table 3: The test results of the server with pre-fetch module

The server with no pre-fetch module						
Time	The third test			The fourth test		
	Group 1	Group 2	Group 3	Group 1	Group 2	Group 3
The first loading time of picture						
5 sec	1	0	1	2	0	1
6 sec	2	5	4	3	4	5
7 sec	6	3	4	5	3	3
8 sec	2	4	2	1	5	2
9 sec	1	0	1	1	0	1
The average loading time	7	6.91	6.83	4.67	5.08	4.75

system, the pictures have been increasing. The structure of side table needs to constantly update. System inserts the latest generated image information into the side table's first node. When users access page [2] and page [3], pre-fetch procedure reads pictures from this side table into cache, waiting for the access of users, so as to achieve predication. For other common pages, their visit numbers always change, system uses count value to record visit numbers. When visit numbers plus one, system travels the whole adjacency list, the count values of all nodes of this page plus 1. At the same time, system re-links the nodes at this side table by the order of count values. Next time when user visits some node of the vertex list, the predicted page which will be read next into the cache is the first node of the vertex and also is the first node of the side table whose count value is the largest.

Performance test: In order to test the performance of this server, we compare the server with pre-fetch module to the server with no pre-fetch module, totally make four times test and each is divided into three groups.

First of all, let 12 computers on the LAN access alarmphoto.cgi the alarm image page in the server with no pre-fetch module at the first time and test its first loading time. The loading time is at about 5, 6, 7, 8 and 9 sec. In the first group we find that there is one computer whose loading time is 5 sec, three computers whose loading time is 6 sec, five computers whose loading time is 7 sec, two computers whose loading time is 8 sec, one computer whose loading time is 9 sec and the average loading time

of the first group is 6.91 sec. The average loading time of the second group is 6.83 sec. The average loading time of the third group is 6.83 sec. Secondly, let 12 computers on the LAN access the next photo of alarm image page in the server with no pre-fetch module and test its second loading time. The average loading time of the first group is 6.91 sec. The average loading time of the second group is 6.91 sec. The average loading time of the third group is 6.83 sec. It is shown in Table 2.

Thirdly, let 12 computers on the LAN access alarmphoto.cgi the alarm image page in the server with pre-fetch module and test its first loading time. The average loading time of the first group is 7 sec. The average loading time of the second group is 6.91 sec. The average loading time of the third group is 6.83 sec. Fourthly, let 12 computers on the LAN access the next photo of alarm image page in the server with pre-fetch module and test its second loading time. The average loading time of the first group is 4.67 sec. The average loading time of the second group is 5.08 sec. The average loading time of the third group is 4.75 sec. It is shown in Table 3.

The time of the first page download shows that: the first picture download time of the server with no pre-fetch module is between 5 and 9 sec, the average time is 6.85 sec and the first download time of the server with pre-fetch module is between 5 and 9 sec too, the average time is 6.91 sec. Experiment results show that the performances that the two servers visit the same page at the first time are the same.

The time of the second picture download shows that: the second picture download time of the server with no pre-fetch module is still between 5 and 9 sec, the average time is 6.88 sec and the second download time of the server with pre-fetch module is between 3 and 7 sec, the average time is 4.83 sec. Experiment results show that the performance that the server with pre-fetch module visits the next page at the second time is better than the performance that the server with no pre-fetch module visits the next page at the second time.

CONCLUSION

The study introduces a number of key technologies of the software architecture for video surveillance system, proposes a new motion detection algorithm, an improved image compression algorithm and an improved web server pre-fetch algorithm. The block-based fixed background difference image motion detection algorithm has a good effect. It greatly reduces the false alarm rate, avoids filtering, reduces the amount of computation, has certain adaptability to light changes, reduces or even eliminates the interference of the CCD jitter, can accurately extract the moving object location and in the course of each test only need take one frame. It is an ideal method for motion detection. The proposed improved image compression algorithm implements the adaptive quantization table compression which is based on motion detection. It converts floating-point calculations into a fixed-point calculation and uses sub-sampling and algorithm optimization to improve the performance. Test proves: the total time of compressing 100 BMP images and completing the USB storage is lower than 48 sec. This compression module in the whole system can complete the emergency compression storage function and does not affect the performance of the real-time image acquisition of the whole system. Compared with the no pre-fetch module web system, the system with improved pre-fetch algorithm has better real-time and efficiency. Experiments show that: compared with other monitoring systems, this system is lower cost, faster response and so on.

ACKNOWLEDGMENTS

The work was supported by major project fund of Ministry of Science (ECK2150A), Hunan Science and Technology Project Fund (2008GK3134) and Science Research Project Fund of Hunan Provincial Education Department (10C0202).

REFERENCES

- Abboud, I., 2006. Deblocking in BDCT image and video coding using a simple and effective method. *Inform. Technol. J.*, 5: 422-426.
- Collins, R.T., A.J. Lipton and T. Kanade, 2000. Introduction to the special section on video surveillance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22: 745-746.
- Feig, E., 1990. A fast scaled DCT algorithm. *Proc. SPIE*, 1244: 2-2.
- Gonzales, R.C. and R.E. Woods, 2002. *Digital Image Processing*. 2nd Edn., Prentice-Hall, New Jersey, ISBN-10: 0201180758.
- Guo, H., Y. Liang Z. Yu and Z. Liu, 2010. Implementation and analysis of moving objects detection in Video Surveillance. *Proceedings of the Information and Automation (ICIA), IEEE International Conference*, June 20-23, Harbin, pp: 154-158.
- Han, L. and Y. Jie, 2007. Large pixel aerial camera image compression algorithm and its application. *Missiles and Guidance*.
- Hapsari, G.C. and A.S. Prabuwo, 2010. Human motion recognition in real-time surveillance system: A review. *J. Applied Sci.*, 10: 2793-2798.
- Haseeb, S. and O.O. Khalifa, 2006. Comparative performance analysis of image compression by JPEG 2000: A case study on medical images. *Inform. Technol. J.*, 5: 35-39.
- Hongjian, W., 2009. Vehicle flow measuring based on temporal difference image. *Proceedings of the 2nd International Conference Intelligent Computation Technology and Automation*, Oct. 10-11, Changsha, Hunan, pp: 717-720.
- Kim, M., J.G. Choi, D. Kim, H. Lee, M.H. Lee, C. Ahn and Y.S. Ho, 1999. A vop generation tool: Automatic segmentation of moving objects in image sequences based on spatio temporal information. *IEEE Trans. Circuits Syst. Video Technol.*, 9: 1216-1226.
- Loeffler, C., A. Ligtenberg, G.S. Moschytz and E.T.H. Zurich, 1989. Practical fast 1-D DCT algorithm with 11 multiplications. *Proc. Int. Conf. Acoustics, Speech Signal Process.*, 2: 988-991.
- Logashanmugam, E. and R. Ramachandran, 2008. An improved algorithm for image compression using wavelets and contrast based quantization technique. *Inform. Technol. J.*, 7: 180-184.
- Mecocci, A., M. Pannozzo and A. Fumarola, 2003. Automatic detection of anomalous behavioural events for advanced real-time video surveillance. *Proceedings of the IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications*, July 29-31, Siena University, Italy, pp: 187-192.

- Moon, H.M. and S.B. Pan, 2009. Implementation of the privacy protection in video surveillance system. Proceedings of the 3rd IEEE International Conference on Secure Software Integration and Reliability Improvement, July 8-10, Shanghai, China, pp: 291-292.
- Ruichao, L., H. Jing and S. Lianfeng, 2009. Design and implementation of a video surveillance system based on 3G network. Proceedings of the International Conference on Wireless Communications and Signal Processing, Nov. 13-15, Nanjing, pp: 1-4.
- Seferidis, V.E. and M. Ghanbari, 1993. General approach to block-matching motion estimation. *Optical Eng.*, 32: 1464-1464.
- Umapathi, C. and J. Raja, 2006. A prefetching algorithm for improving web cache performance. *J. Applied Sci.*, 6: 3122-3127.
- Vetterli, M., 1985. Fast 2-D discrete cosine transform. Proceedings of the Acoustics, Speech and Signal Processing, (ICASSP'85), Tampa, Florida, pp: 1538-1541.
- Wong, W.K., J.T.Y. Liew, C.K. Loo and W.K. Wong, 2009. Omnidirectional surveillance system for digital home security. Proceedings of the International Conference on Signal Acquisition and Processing, April 3-5, Kuala Lumpur, pp: 8-12.
- Xu, Z. and H.R. Wu, 2010. Smart video surveillance system. Proceedings of the IEEE International Conference on Industrial Technology, March 14-17, Australia, pp: 285-290.