The Design of Large-Scale Surveillance Network based on GRID Structure

Fu-Chien Kao, Jyun-Yi Du and Siang-Ru Wang

Department of Computer Science and Information Engineering, University of DAYEH
168 University Rd., Dacun, Changhua 51591, Taiwan

Summary
Brightness obviously affects the pixel value of an image sensor. Image identification from a traditional sensor may generate a bias and surveillant misunderstandings because of weather. The proposed intelligent image sensor integrates an optoelectronic converting circuit and an embedded system and can convert the brightness to an analog voltage. From the voltage value, we can establish a contrast list to show the relationship of the analog voltage and image pixel values. Using the list and applying the linear interpolation method, we can eliminate the effect of the brightness and raise the identification rate. To improve the deficiency of conventional residence surveillance systems in real-time, multipoint, dispersed surveillance, the research proposed an integrated grid technology-based intelligent image surveillance system design; To further improve the congestion phenomenon created by the proposed multipoint, dispersed surveillance system on the rear-end storage system, the research incorporates the grid technology to design a distributed storage method with a embedded Ganglia Broker. The function of the proposed embedded Ganglia Broker is to balance and disperse rear-end computer storage space and system load, and then offers a load-balancing storage framework to resolve the congestion problem as a result of transferring a large amount of data; the system also adopts Globus's RFT components to process data replication and collect all anomaly image files, offering the user with management and surveillance functions...
Key words:
Ganglia Broker, Surveillance, Embedded System, Grid.

1. Introduction

Most traditional security systems record a continuous image from the sensors. It means that the modern security system records all images from cameras 24 hours a day and seven days a week, such as in the systems in banks, shops, residential communities, etc [1]. Therefore, the traditional security system requires a great mount of storage space and high costs. In addition, many unnecessary images are recorded in the system. Conventional residence surveillance systems merely store the surveillance image transmitted from the image sensor; while in recent years the constantly improving residence surveillance systems have expanded form the conventional theft, fire detection to a broader residence surveillance service, such as a surveillance system can be sued to measure the blood pressure, blood sugar, cardiogram, blood oxygen concentration, body temperature and such for a variety of health indicators [2]. As current residence surveillance systems further integrate the mobile communications functions, in the event of a burglary, the system can automatically dial the administrator by telephone or via short messaging. Essentially, as conventionally residence surveillance systems are confined to a round-the-clock recording of continuous image, it compels the system to command a large amount of storage space for storing the image. Yet as residence surveillance system functions are perfected, the demands for the rear-end server's functions will become ever more stringent. While as servers need to record image data receive fro all sensors around a residence, the more number of the sensors, the more demanding the server load would become. With that, the conventional one-on-one surveillance systems' singular storage framework can no longer handle a multiple, dispersed surveillance framework for large amount data transmission [3].

Thanks to technological progression, what the computer can offer in functions is no longer limited to simple computation but the execution of more complex scientific computation and calculations. Yet as supercomputers remain expensive in cost to prevent them from being adopted in general engineering application, this has prompted a thriving development of a computational grid-, or data grid-oriented grid dispersed framework applications; it operates through utilizing the Internet's idle computer resources to bring forth a effective enhancement to the execution by dispersing and storing tasks or data that require a large amount of computation [4-8].

To resolve the rear-end server's overload problems common among conventional surveillance systems and to explore a suitable multipoint, dispersed surveillance and storage mechanism, the research has presented an integrated grid of an intelligent image surveillance system design; the presented intelligent image sensor offers the function of voluntarily detecting whether the image capture contains anomaly, and only stores the image on the rear-end server in the wake of anomaly to effectively

Manuscript received August 5, 2009
Manuscript revised August 20, 2009

conserve the firmware's storage space, and reduce the equipment cost; and the system adopts grid-related technology to integrate and disperse the rear-end computer's storage space, and to effectively balance the server load. The thesis's subsequent sections and chapters include sections and chapters focusing on the grids, Ganglia Broker system framework, figurative analysis and a conclusion.

2. Intelligent Image Sensor

The research proposes a design of intelligent image sensor for a surveillance system based on the integration of 32-bit embedded system development kit with integrating an optoelectronic converting circuit, a GPRS/GSM module, and an image sensor to develop an intelligent image sensor with mobile communication function.

2.1 Embedded System Hardware Structure

In the research, we used the Samsung S3C4510 ARM7TDMI embedded system development kit as the development of intelligent-sensing sensor module. We used the A/D converting module of the development kit to retrieve the analog voltage values from the optoelectronic converting circuit and convert them to digital values. Then we can generate the relationship between the brightness and the analog voltage values. We also use the image-sensing module of the development kit to get the monitoring image pixel values under different brightness and analog voltages. After analyzing and comparing the pixel difference under different brightness conditions, we can build a contrast list of optoelectronic converting voltage values and pixel difference. Figure 1 shows the embedded system development kit.

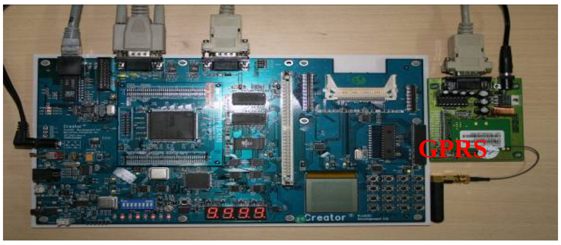

Figure 1. The Embedded System Development Kit.

2.2 The Proposed Intelligent Image Identification Method Applied to Remove the Change of Brightness

This research applies a simple optoelectronic converting circuit to determine the brightness and convert the voltage value for analysis. The integration of the relative circuit and embedded system development kit is shown as Figure 2 [2].

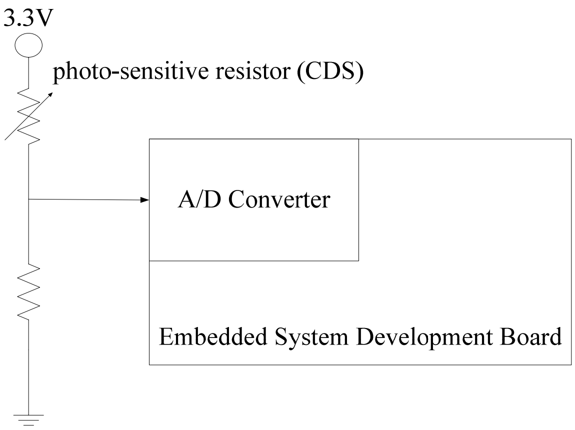

Figure 2. The Optoelectronic Converting Circuit.

As shown in Figure 2, we connected the optoelectronic converting circuit to the A/D converting connection of the development kit. By using a CDS that changes its resistance value following a change in light levels, we can get the analog voltage value under different brightness levels. The CDS we used in the research can change its value in the range of 8K and 12K Ohm. The current of the optoelectronic converting circuit is 3.3V and offered by the development kit. Then we built the relationship list of the voltage value of the optoelectronic converting circuit and the relative pixel difference in Table 1. Because the brightness range is too wide to get all analog voltage values, we use the linear interpolation method and the contrast list of the analog voltage of the brightness and the pixel difference to quickly generate the target analog voltage and the relative pixel difference. The formula of the linear Interpolation method is showed in Equation 1. The variable x in the formula is the new analog voltage value from the optoelectronic converting circuit. The function f(x) represents the relative pixel difference from the new analog voltage value. The variables a and b are the known voltage range in Table 1. Functions f(a) and f(b) are the two relative pixel differences of the analog voltage range. Using the Interpolation formula in Equation 1, we can generate the pixel difference of the new analog voltage value from the optoelectronic converting circuit. For example, assume that variable x is 0.75, which is between 0.7 and 0.8. Set the variables a as 0.7, and variable b as 0.8. We can get the result of f(0.7) as 8 and the result of f(0.8) as 20. Use these two values in Equation 1, and we can get the pixel difference f(0.75) as 14.

Table 1. The Contrast List of the Voltage and Pixel Difference.

Analog Voltage Value	Pixel Difference
0.7	8
0.8	20
0.9	29
1.0	42

$$f(x) = \frac{b-x}{b-a} f(a) + \frac{x-a}{b-a} f(b) \quad (1)$$

The flow chart of the intelligent image identification mechanism that integrates the mobile communication function is shown in Figure 3 [2]. The intelligent image identification combines the optoelectronic converting circuit with the linear insertion method. The mechanism generates pixel difference values when the level of brightness changes. We use the value to eliminate the effect of brightness on the image pixels. The process of the image identification is to determine the contrast list of the optoelectronic converting voltage and pixel difference from the optoelectronic converting circuit. From the list, we can use the linear interpolation method to calculate the relative pixel value of the voltage retrieved from the optoelectronic converting circuit. After we delete the difference from the original image pixels, we can get a new value. We compare the new value with the old value, and we can judge whether the result is normal or not. If the new value is abnormal, an alarm is raised by the system and sends a short message to the specific user/users, and the intelligent image sensor saves the images in the server. Otherwise, the system continues monitoring.

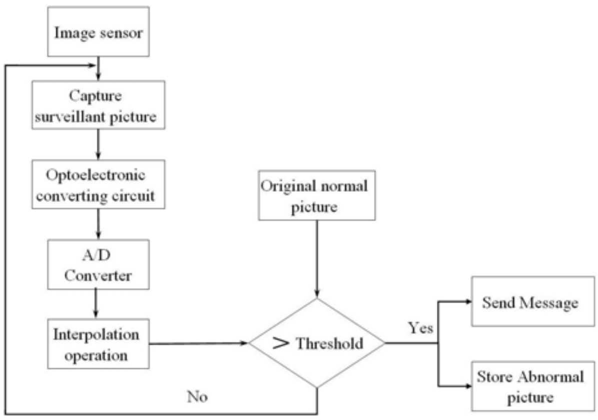

Figure 3. The Infrastructure of the Intelligent Image Identification Surveillance System.

3. Grid

3.1Grid Concept

Grid computation refers to integrating shares resources on the World Wide Web Internet by systematically regrouping such resources to achieve resource sharing and a collaborated computation. In 2002, Ian Foster presented that grids need to satisfy three concepts [4],

(1) Grids are able to integrate a variety of resources and coordinate a host of users enabling users of different environment to access resources in different environment. With that, under the grid environment, there is a need to take into account pertinent security, certification, and authorization issues.
(2) Grids need to utilize standard, open, universal protocols, and these protocols must solve the certification, resource sharing, and resource storage and retrieval problems.
(3) Grids allow the resources to be accessed and utilized to satisfy the user's varied utilization needs.

Under the foresaid concept, rules have been developed to define a virtual organization [5]. Under such virtual organization, any resource can satisfy the foresaid concept and development. In recent years, many organizations around the world have presented many types of grids, and there are two groupings when classified by the nature of problems the grids are designed to solve: (1) Computational grids and (2) Data grids, while in fact a majority of grid computing environments are a combination of the two.

(1) Computational Grids: As computational grids emphasizes on computing resource integration, thus computing the nodes in a grid has been done through clustering computers or with high-performance computers.
(2) Data Grids: Data grids integrate a virtual organization's file system, enabling the user to conduct cross-virtual organizational file storage and retrieval.

There are organizations around the world that utilize the grid technology to set up large-scale computational systems, such as earth's satellite imagery observation, biomedicine DNA computational simulations, weather simulations, military simulations and such that all turn to using the grid technology to process dispersed computation and manage enormous figurative data. While currently a most commonly used intermediary software for launching these large-scale grids has been Globus Toolkit [6]. Globus is a research and development brainchild of U.S. Argonne National Laboratories, and 12 universities and research institutes have taken part in the project. Globus has achieved developing grid-related theory and concept by conducting relevant studies on resource management, data management, information management and grid security, and is currently being widely adopted in many large-scale grid platforms.

3.2 Grid Framework

Developing a comprehensive grid system calls for an infinite number of gird protocols, services and package software developed. Its development objectives and focus center on the protocols, allowing a virtual organization's users to store/retrieve and share the resources. Figure 4 depicts a five-tier grid framework [7] [8], and each layer's framework and services are described as follows,

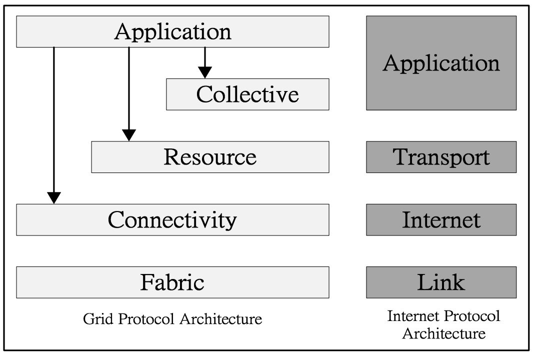

Figure 4. Five-Tier Grid Framework

(1) Fabric Layer: Grid resources refer to all physical data that can be utilized in a grid, including computational resource, storage resource, networking resource and the like.
(2) Connectivity Layer: Under the grid concept, users in different environments are able to utilize the resources in different environments. Thus a grid needs to define a mechanism that allows intercommunication between the resources and the resources, and between the resources and the user; it also needs to ascertain the issues that the user has undergo the virtual organization's certification or the storage and retrieval of resources are conducted securely.
In a grid, the user's and resources' communication is built upon a secured communications mechanism; the connectivity layer defines the communications protocols and certification mechanism that enable resources on the fabric layer to exchange data securely, while communications protocols are built upon the current networking protocols, i.e. TCP, UDP, DNS and the like, rather than an original networking communication protocol.
(3) Resource Layer: The resource layer defines the resources' independent operations and specifications, rather than specify the cooperation between resources and resources. The resource layer is built under the connectivity layer's communications protocols and certification mechanism, and the defined protocols including the resources' initialization, execution and the resources' shared operations, etc.
(4) Collective Layer: It defines, on the resource layer, each resource's attribute and method. The collective layer defines the resources' mutually collaborated storage and retrieval methods and strategies. The collective layer's defined protocols and services enable the user to store and retrieve multiple resources and work together to complete a process.
(5) Applications Layer: The applications layer offers the user with applications software, grid portal and tools. The user is able to access all resources in a grid using these software tools. From the viewpoint of the applications program, resources in a virtual organization are accessed and utilized to achieve the objective of a mutually collaborated computation of the entire grid's resources.

3.3 Grid Middleware

Globus Toolkit is currently a most influential grid computing software worldwide; Globus Toolkit offers relevant component service functions in resource management, signal service and figure management, ad is a grid-launching software tool for various platforms, helping and developing the establishment of large-scale gird experiment platforms. Globus Toolkit provides some critical components; the relevant description is as follows,
(a) Grid Security Infrastructure (GSI): GSI [10] [11] grids' security certification and encrypted communication offer single sign-on, identity authentication, and transmission encryption functions.
(b) Resource Management: Grid resource management serves to collectively manage a variety of resources dispersed in a grid, enabling all resource utilization requests are able to access the resources they are looking for. GRAM [12] [13] offers standardized interface protocols and API that allow the users to access remote resources.
(c) Information Services: MDS [14] [15] offers a set of standard mechanisms to gather the resources' status and adjusted information. The grid information service center offers a grid environment's information registration inquiry and update processes, allowing a real-time response on the state of a grid implementation.
(d) Data Management: GridFTP [16] [17] serves to provide high-performance data transmission, supporting the three-way transmission, parallel transmission functions, and only enables transmission under a safe GSI authentication. As system file backup is often executed in

data grids, prompting a multiple number of identical files in the grid system, the grid presents a mechanism and solves the identical file problem.

3.4 Java Cog Kit

The user accesses all resources within an integrated grid through the developed program directly. Java Cog Kit's Jblobus.jar module provides the basic API, enabling the program be used on GridFTP, GRAM, GSI functions directly. The GRAM, GSI functions are often used to establish a grid portal, as depicted in Figure 5.

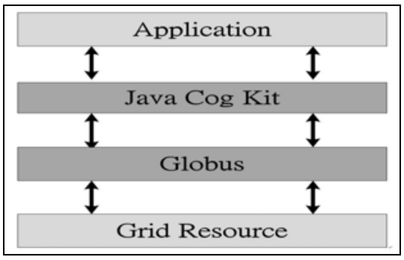

Figure 5. Java Cog Kit and Globus'S Correlation Chart.

4. Ganglia Broker

To enable anomaly images captured by a multiple image sensors be stored on the rear-end server, the research has presented a grid technology for integrating resources scattered in the rear-end computer. A Broker design with a load-balancing function is used to solve the image storage "colliding" problem. The dispersed storage framework presented utilizing grid-technology to integrate the rear-end computer not only effectively balanced the rear-end server load, but can timely handle the storage of abnormal image in multipoint surveillance. The system needs to eliminate causing the other rear-end servers to idle at the time when storing image to the same rear-end server, which causes the system network load to reach a saturated limitation.
To maintain the system's stability, the research utilizes embedded Ganglia Broker to execute the system's balancing work. The designed Broker filters the surveillance data derived from Ganglia's surveillance software [22-24], such as analysis data of the CPU utilization ratio or the memory cache's utilization ratio and such, to locate the lightest load rear-end server for storing the front-end anomaly image. Ganglia, a software initially developed by a team at the State of California University in Berkley, was initially used in computing clustering systems at UCLA in Berkley. Now there are many large-scale grid systems adopting Ganglia to monitor the entire system; for example, the San Diego Supermarket Center also adopts a Ganglia surveillance system. Ganglia records each node's surveillance data using an XML file format, operates on TCP/IP communications protocol, and supports Unicast/Multicast transmission mode, offering the user a Web portal's web page use and inquiry.

4.1 Embedded Ganglia Broker

A distributed surveillance tool, Ganglia is often used to monitor high-performance clustering systems. Ganglia is able to survey the system status; these constant programs exchange and communicate the data using UDP broadcast, and upload them to Gmetad using a tree structure to sort all of the surveillance data. And the communication between clustering computers and clustering computers has the data exchange executed through TCP using the XML file format.
Ganglia, in the clustering computer structure, is an equivalent of the tree structure, and the nodes at the non-leaf nodes are in charge of gathering the leaf node's surveillance data. Ganglia codes have five cores, such as Ganglia Monitoring Daemon (gmond), Ganglia Meta Daemon (gmetad), Gmetric, Gstat and Ganglia PHP Web Frontend, as depicted in Figure 6. This research proposed a design of embedded Ganglia Broker by embedding those related service(such as web service) into the embedded system development board.

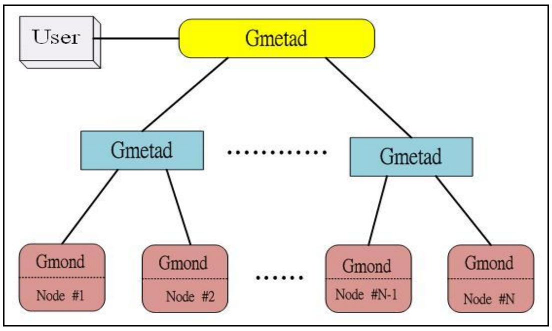

Figure 6. Ganglia structure.

By modifying the source surveillance service Ganglia prototype program supplies, the research proposed a embedded Ganglia Broker(EGB) that offers the service functions of parsing XML files, and locating the lightest load rear-end server to support the front-end mage sensor's resource monitoring, the monitoring interface is shown in Figure 7.

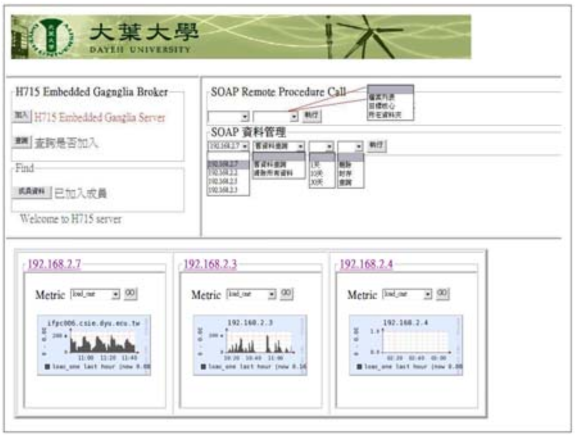

Figure7. System Resource Monitoring

5. System Structure

To reduce the numerous, continuous surveillance images generated by an image sensor, and to improve the deficiency of the conventional approach in a sequential capturing of the surveillance point to keep from providing a dispersed, multipoint real-time surveillance service, the research presents an intelligent image sensor design, combing the grid technology to offer a distributed storage method with a load-balancing function. The proposed system framework, as depicted in Figure 8, consists of a front-end intelligent image sensor, an image storage proxy server – the embedded Ganglia Broker, grid structure's rear-end image storage servers.

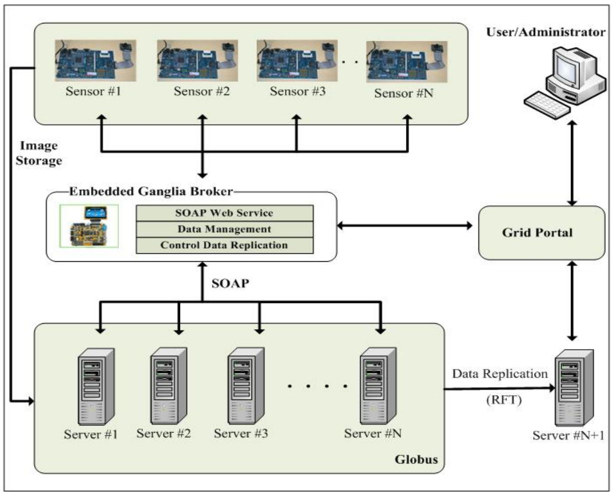

Figure 8. System Structure

When a front-end intelligent image sensor identifies anomaly image, the sensor links up on-line to the embedded Ganglia Broker, and EGB feeds the location of the rear-end's smallest sever, prompting the image server to link up on-line to anomaly image stored at the rear-end server, and the Broker stores the relevant information on the anomaly image. The system also integrates anomaly image files scattered in the rear-end server using the grid technology, and enables the user to monitor anomaly image through the designed grid portal.

5.1 Embedded Ganglia Broker Implementation

When the front-end image sensor detects the generation of voluminous image data, to ensure a real-time storage effect, the system serves to ensure that all image data can be dispersed and stored at the rear-end computer; therefore, the research proposed a embedded Ganglia Broker to interpret the surveillance data gathered at the rear-end as the basis for selecting image to be stored on the server. The EGB component – Gmetad program records system information on the rear-end's nodes, such as the CPU utilization rate, memory cache size, networking flow volume and so forth, and produce these information into XML files. The proposed embedded Ganglia Broker will download and interpret the XML files, and locate a rear-end image storage server with the lightest load, as depicted in Figure 9.

Figure 9. The implementation screen of embedded Ganglia Broker

In the meantime, the EGB generates the only anomaly image's file name to ensure that the anomaly image the image sensor generates will not create a repetitive file name on the rear-end file system. The embedded Ganglia Broker presented not only supplies the front-end sensor with a lightest load server's IP and file name, allowing the front-end image sensor to link up to the anomaly image stored on the rear-end computer on its own, but also stores information, such as the image file name, anomaly occurring time, smallest IP load, location where anomaly image occurred, to the database.

5.2 The Implementation of System Structure

The system framework's actual implementation is as depicted in Figure 10, and relevant launching software includes, Linux CentOS5, Globus Toolkit, Ganglia, Java and Java Cog Kit.

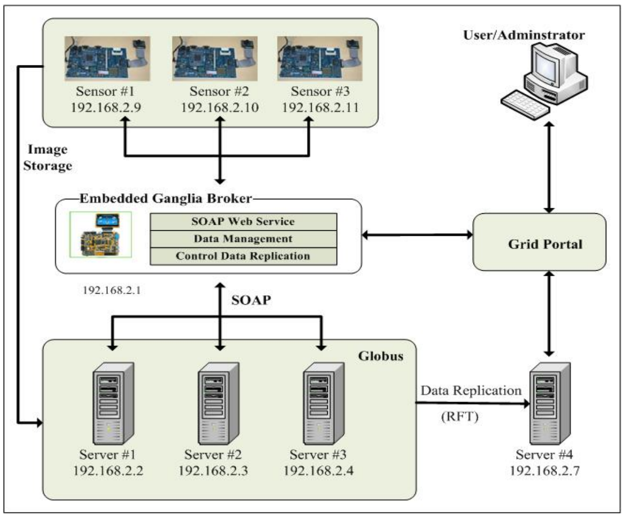

Figure 10. The system implementation diagram

The system integrates all anomaly image files scattered on the rear-end server using the Globus's RFT component, and defines the automated execution scheduling using the Crobtab tool loaded on the rear-end image storage server. The research develops the image resource replicating program using Java Cot Kit by defining the Crobtab tool to run the program on the system every one hour, and the captured image data is replicated onto the rear-end image storage server (192.168.2.7) for inquiry by the user.

6. Conclusions

The research presents an integrated grid technology-based intelligent image surveillance system design. The research proposes an intelligent image sensor for surveillance systems with mobile communication function. By combining the simple optoelectronic converting circuit design with the linear Interpolation method, the research eliminates the effects of brightness that we supposed to be the main reason for incorrect image identification. The research has conducted an image identification and comparison analysis in a full image of 19200 pixels and the 81-pixel critical region. We have found that the image identification becomes faster and more effective after eliminating the effect of brightness, and this is even when we just compare the critical regions of the image. System uploads the anomaly image to the grid-supporting rear-end sever (ISSG) for storage enabling the system not only to bypass storing voluminous image data to conserve the cost of storage equipment, but also to effectively support the multipoint, real-time surveillance by dispersing the rear-end server's load.

The proposed distributed storage mechanism with load-balancing function utilizes the embedded Ganglia Broker to balance and disperse the load and storage space of rear-end image storage server group(ISSG), and operates also Globus RFT components to complete data replication, and to sort the anomaly image files for the user's management and inquiry. As opposed to the rear-end server of conventional surveillance system that can only support a small number and fixed surveillance spots as hindered by its storage space and server performance, the proposed system framework is more than suitable for distributed, multipoint surveillance, and can be incorporated with a front-end image sensor and a rear-end server to offer an effective, large-scale surveillance system application. Besides, the proposed EGB decreases the cost of system effectively.

Acknowledgments

This paper has been supported in part by the grant NSC95-2520-S-212-001-MY3 from the National Science Council of Taiwan.

References

[1] Apache HTTP SERVER PROJECT, http://httpd.apache.org
[2] Ann Chervenak , Ewa Deelman , Ian Foster , Leanne Guy, Brian Tierney,(2002) "Giggle: A Framework for Constructing Scalable Replica Location Services", Proceedings of the IEEE/ACM SC2002 Conference.
[3] BORJA SOTOMAYOR , "Globus Toolkit 4 PROGRAMMING JAVA Services" , Pages ; 7-10.
[4] BORJA SOTOMAYOR , "Globus Toolkit 4 PROGRAMMING JAVA Services" , Pages : 39-41.
[5] Fu-Chien Kao, Yun-Sheng Tsai,(2008) "The Design of Intelligent Sensor Network based on Grid Structure", Proceeding of IEEE Asia-Pacific Service Computing Conference, pp.623-630.
[6] Grid Security Infrastructure （GSI）, Globus Project-GlobusToolkit,2005, http://www-unix.globus.org/toolkit/docs/4.0/security/
[7] Globus GRAM Architecture , http://www.globus.org/toolkit/docs/4.0/execution/prewsgra/
[8] Globus MDS Architecture , http://www.globus.org/toolkit/docs/4.0/info/
[9] Globus RLS Architecture , http://www.globus.org/toolkit/docs/4.0/data/rls/index.pdf
[10] Http://ganglia.sourceforge.net/
[11] Ian Foster , Carl Kesselman and Steven Tuecke ,(2001) "The Anatomy of the Grid Enabling Scalable Virtual

Organizations", Supercomputer Application, Page: 2 – 6. [11] IBM Red Book , "Introduction to Grid Computing with Globus" , Pages : 51-78.

[12] Ian Foster , Carl Kesselman and Steven Tuecke ,(2001) "The Anatomy of the Grid Enabling Scalable Virtual Organizations" Supercomputer Application, Page: 6-14.

[13] Ian Foster, (2002)"What is the Grid? A Three Point Checklist", Argonne National Laboratory & University of Chicago July 20.

[14] IBM Red Book , "Introduction to Grid Computing with Globus" , Pages : 51-78.

[15] IBM Red Book , "Introduction to Grid Computing with Globus" , Pages : 135-138.

[16] IBM Red Book , "Introduction to Grid Computing with Globus", Pages : 138-140.

[17] IBM Red Book , "Introduction to Grid Computing with Globus", Pages : 140-142.

[18] IBM Ganglia , http://www.ibm.com/developerworks/cn/views/grid/tutorials.jsp?cv_doc_id=97804

[19] Java Cog Kit, http://wiki.cogkit.org/index.php/Main_Page

[20] Maozhen Li , Mark Baker , "The Grid: Core Technologies", Page : 153-243.

[21] OpenSSL , http://www.openssl.org/

[22] RFT , http://www.globus.org/toolkit/docs/4.0/data/rls/index.pdf

[23] Simple CA, http://www.globus.org/toolkit/docs/4.0/security/simpleca/

[24] Taiwan Secom Co., Ltd http://www.secom.com.tw/product.asp?act=1

[25] The Globus Project, http://www.globus.org/

[26] The Grid Architecture , http://www.globus.org/toolkit/about.html

[27] Zhi-Hua Ji, (2007) "The Design of Vehicle Surveillance System with Intelligent Image Sensor", Thesis of Da-Yeh University Department of Computer Science and Information Engineering.