The UTOPIA Video Surveillance System Based on Cloud Computing

Jong Won Park, Chang Ho Yun, Hak Geun Lee, Chul Sang Yoon, Hae Sun Jung, Yong Woo Lee
School of Electrical & Computer Engineering
The Smart City (Ubiquitous City) Consortium, the University of Seoul
Seoul, South Korea
{comics77, touch011, withteas, csyoon2014, banyasun, ywlee}@uos.ac.kr

***Abstract*— Smart City is an intelligent city which satisfies human beings' desire to enjoy IT services with any device, anytime, anywhere and a future city model based on Internet of Everything (IoE) or Internet of Things (IoT). It includes a lot of video cameras which are networked together and the networked video cameras enable a lot of smart city services. The networked video cameras generate huge amount of video information, real big data for the smart city all the time and the smart city should process the big data in real-time in most cases. It requires a lot of computational power and usually takes a lot of time thus it is a very challenging task. Cloud computing can be a good solution to address this matter and there are many cloud computing methodologies. This paper presents our own methodology to analyze the video images using MapReduce. The methodology was implemented in our smart city middleware called SOUL, which was designed for our smart city called UTOPIA. Some of the implemented system is introduced in this paper. This paper also introduces smart analyzing functions of the video surveillance system in SOUL and how they use the scalable video streaming of SOUL to acquire the video data from the networked surveillance video cameras. The system is easy to be deployed, flexible and reliable. The performance evaluation was experimented and we found that our proposed system worked well. Some analyzed results of the performance evaluation are presented in this paper.**

Keywords-Smart City; Cloud Computing; Smart Video Surveillance System; MapReduce; Networked Video System; Image Processing; Big Data.

I. INTRODUCTION

A smart city is an ICT based city, converges every possible information system, such as residential, environmental, medical, business, governmental, social information systems and the whole activities in a city into a virtual system or a global system which works for human beings. It has key aspects such as smart infrastructure, smart citizen care and smart administration which include smart traffic management, smart ecological environment management, smart energy management, etc.

In the smart city, there are a lot of networked video cameras to support smart city services and they generate huge data continuously. It is usually required that the smart city should process the huge and continuous video data, real big data in real time [1][2]. We think that this requirement can be successfully solved by cloud computing technology since cloud computing can provide the computing resources to process big data successfully.

More specifically speaking, Hadoop [3] can be a useful solution among many open source solutions for cloud computing, since Hadoop stores the data in Hadoop Distributed File System (HDFS) and provides the platform that can process them with simple MapReduce programming model [4]. It enables us to process the big semi-structured data and big unstructured data that were hard to be processed before. Indeed, the video data which continuously produced in smart city are an unstructured data and big data.

This paper presents a research result about the smart video surveillance system based on cloud computing for smart city. The smart video surveillance system collects big video data through scalable video streaming which smoothly processes big data traffic from large number of networked video cameras even with limited bandwidth and process the big data with MapReduce model.

The contributions of this paper can be described as follows:

1. This paper presents a video surveillance system for smart city which process big video image data with scalable video streaming and cloud computing in real time efficiently. Thus, it contributes to big data processing, video image data processing, cloud computing, scalable video streaming and real time processing and the video surveillance system
2. This paper proposes the video surveillance system was implemented in in our smart city middleware called SOUL, which was designed for our smart city called UTOPIA. Thus, it contributes to the field of smart city and the field of the middleware system for the smart city.
3. This paper presents the result of performance evaluation.

This paper is organized as follows. Section 2 gives overview of the SOUL smart video surveillance system in UTOPIA. Section 3 explains the architecture and operational principle of video surveillance system in SOUL. Section 4 explains the details of the cloud computing based big video data processing. Section 5 explains the performance evaluation work. Section 6 explains related works and compares them with our work. Finally, Section 7 gives the conclusion.

II. SOUL Smart Video Surveillance System in UTOPIA

In smart city, citizens can enjoy various integrated smart city services. In order to support the smart city services, we designed the ICT based smart city model which has 3-tier architecture and we call UTOPIA. UTOPIA consists of the 3 tiers: the Body tier, the Processing tier and the Presentation & Remote Control tier.

The Body tier in UTOPIA is like the body of the human being. This tier includes various kinds of information collection devices such as networked sensors, networked video cameras, microphones, GPSs and etc. It collects the various kinds of data in a smart city through the converged network.

The Processing tier works as the brain of smart city. It smartly processes the data from the Body tier. In smart city, critical decisions should be made in timely manners for various integrated smart city services. For it, we invented a smart middleware system which we call SOUL. SOUL has multi-layered architecture for many benefits such as expandability, decreasing complexity, component reusability, etc. It is the heart of the Processing tier.

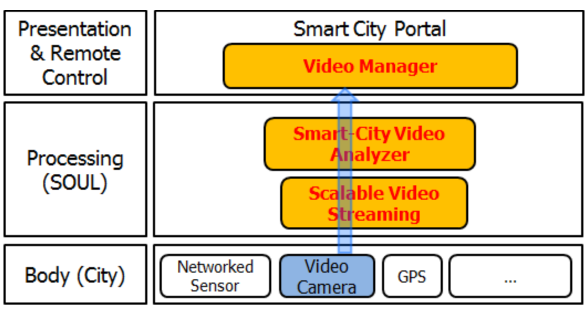

Figure 1. The architecture of UTOPIA and SOUL.

SOUL has the common device interface to various kinds of information collection devices such as networked sensors, networked video cameras, GPSs and etc. It manages the converged network, collects data from the Body tier through the scalable video streaming function, and sends the collected data to the smart city video analyzer. It supports the variable kinds of protocols and a general gateway like function for the Body tier [1][2][5].

SOUL processes the acquired data and makes smart services using the ontology-based context data management. It has intelligent inference engine and does automatic service discovery, service deployment and service execution based on inferred results so that it can provide context-aware smart services [6]. For it, SOUL manages computer resource, does cloud computing and Grid computing [7]. SOUL has the interface to the Presentation and Remote Control tier.

The Presentation & Remote Control tier provides the interface to manage the smart city and provides various smart city services to end users. The smart city portal is an example. Through it, the administrator can control and monitor the smart city. Also, the citizens enjoy the smart city services.

The portal provides the cloud computing interface to end users [8][9][10][11][12] and the tele-management interface to control remote devices such as water pumps, fire doors, emergency devices, remote networked video cameras, etc. [5][13][14][15].

The content in this paper includes our works such as the network adaptive scalable video streaming, water quality control and fire accident management, which we do not deal in detail in this paper [16]. The network adaptive scalable video streaming with the scalable video coding techniques is used in order to save network resources [17][18][19]. Figure 1 shows the concept of the SOUL Smart Video Surveillance System in UTOPIA.

III. Video Surveillance System in SOUL

The architecture of the SOUL Smart City Video Surveillance System is shown in Figure 2. The SOUL Smart City Video Surveillance System consists of Video Manager which belongs to Presentation and Remote Control tier, Smart-City Video Analyzer and Scalable Video Streaming which belongs to SOUL as shown in Figure 1.

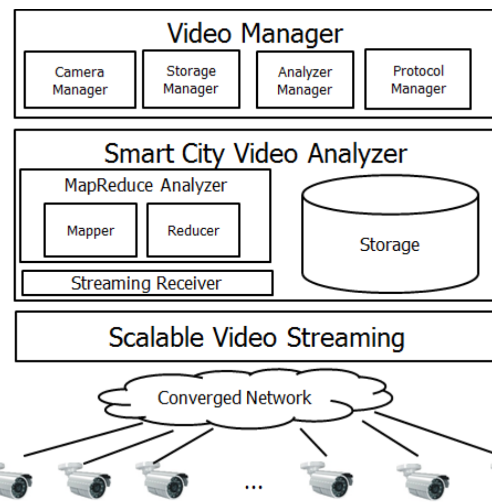

Figure 2. The architecture of SOUL Video Surveillance System.

Video Manager has several modules to manage Smart City Video Analyzer and Scalable Video Streaming. Camera Manager monitors and manages the status of cameras linked to smart city. Storage Manager monitors the status of Storage in Smart City Video Analyzer. Analyzer Manager changes the resolution, codec, and frame rates of video from user's request and monitors the status of MapReduce Analyzer. Protocol Manager manages video streaming protocols such as Real Time Streaming Protocol (RTSP) and Real Time Messaging Protocol (RTMP). Smart City Video Analyzer consists of Streaming Receiver, MapReduce Analyzer and Storage. The Streaming Receiver collects input streams from Scalable Video Streaming. It also delivers the input stream from outside SOUL to inside of SOUL. HDFS is used to store the big video data from Streaming Receiver.

MapReduce Analyzer will be described in the details in the next section.

Smart city need a lot of video cameras in order to manage smart city efficiently. Thus, SOUL was designed to efficiently manage the large amount of video data and has the scalable video streaming component to smoothly manage the video data traffic from large number of networked video cameras. It uses a noble mechanism which controls admissions of clients, extracts bit-streams, and allocates appropriate channels to do the context-aware and the network-adaptive video streaming as shown in Figure 3. The noble mechanism can save system resources and network resources through self-learning using scalable video coding techniques [19].

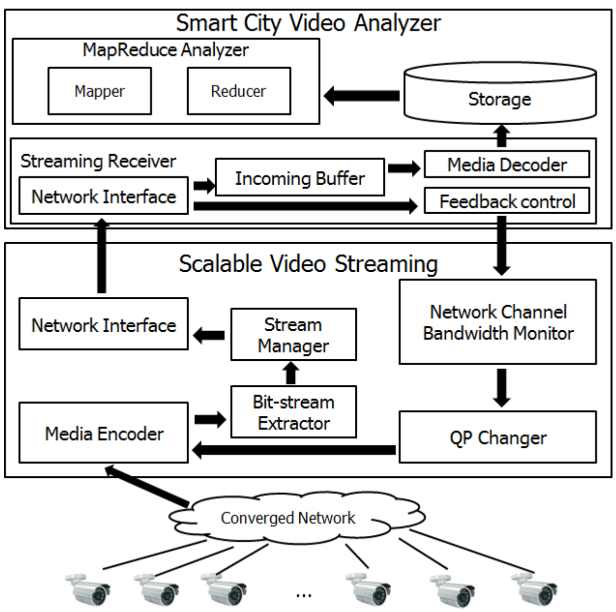

Figure 3. The operational principle of of SOUL Video Surveillance System.

The operation of the scalable video streaming is as follows. Media encoder encodes the video data acquired from each camera linked to the converged network in the smart city. When scalable video streaming encodes video data, it checks the changing instructions from the QP (quantization parameter) changer. Thus, the acquired video data are encoded with various QP values, determined by the QP changer with information from the network channel bandwidth monitor. The network channel bandwidth monitor acts as a network analyzer. It analyzes network bandwidth in real time using a feedback signal and the strength of radio wave signals and the available bandwidth of stream receiver.

In the initialized state, the Scalable Video Streaming sends a message to Streaming Receiver of Smart City Video Analyzer, which responds with their available bandwidth. This information is sent from the feedback control. If the signal is strong, Streaming Receiver considers the good condition of network channel. In contrast, if the signal is weak, the network channel is considered to be in a bad condition.

The feedback control module of Streaming Receiver helps the network channel bandwidth monitor to evaluate the condition of the network channel. After receiving video data, Streaming Receiver responds.

If the channel conditions change very frequently, the feedback control sends information in a minimum state until the channels stabilize. At the very least, this information is used again to evaluate the user network. Thus, feedback control module sends only the information of current bandwidth of Streaming Receiver to scalable video streaming. Scalable video streaming determines the QP value using these data.

The network channel bandwidth monitor observes the network channel condition continuously. This module evaluates the available bandwidth and communicates with the QP changer module to control the encoding rate of the encoder based on a predetermined QP value. Then, The QP changer module regulates the encoding level.

The live video data are encoded by the determined QP value in the media encoder, which is in the network adaptive live streaming module. Encoded video data are transmitted through a bit-stream extractor and a stream manager, which packetize data appositely. These scalable video data packets are separated to adapt to various network conditions.

IV. CLOUD COMPUTING BASED BIG VIDEO DATA PROCESSING

The operation of “MapReduce Analyzer” is shown in Figure 4. Video input data are split by static offset. The key of map phase input is the path of video file as shown in table 1. The value of map phase is each offset. Mapper trims the video file according to path of video file and offset. And mapper analyzes trimmed video through the “Analyzing Module”. After that, output of the mapper has the name of video as the key and analyzed data as the value.

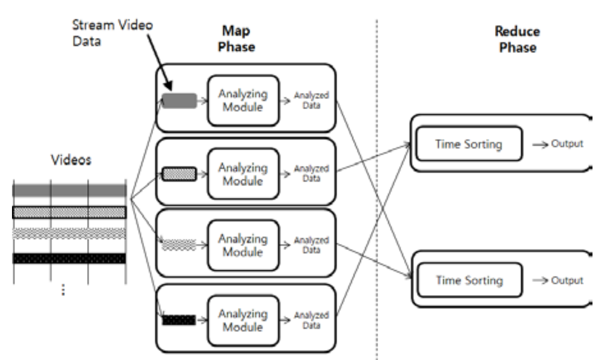

Figure 4. The operation of MapReduce Analyzer.

After generating output of the mapper, intermediate data is sorted according to the name of video. The Reducer receives the pair value of <video ID, list(analyzed data)> and sorts the analyzed data according to the time and, finally, writes the output to HDFS.

TABLE I. THE KEY VALUE PAIR OF MAPREDUCE

Map Phase	Input	<video path, offset>
	Output	<video ID, analyzed data>
Reduce Phase	Input	<video ID, list(analyzed data)>

Splitting the video source is important in our approach. Y.S. Wu et. al. [20] splits the raw video by chunk size to encode but we do not and use compressed video data. The group of picture (GOP) in H.264 contains two frame of picture types such as the Key frame and the non-Key frames. Key frame (I-frame) is the reference frame which represents a fixed image and is independent of other frame type. In our approach, we split the video static GOP length to avoid to loss the key frame and assume that the processing time of computer vision algorithm is related with the frame length.

V. PERFORMANCE (EVALUATION)

There are two main approaches to object detection: temporal difference and background subtraction [21]. We use temporal difference to evaluate performance. The application was implemented in Apache Hadoop MapReduce environment 0.23.0 and was written in Java language. The "x264 codec" which is one of open sources for H.264 codec was used. FFmpeg's libavutil and ffprobe libarary were used to handle video image. Analyzing Module uses OpenCV 2.4.3 that is a very popular computer vision library and is written in C++ language. These libraries were installed on every cluster node, due to dependencies problems between the libraries.

We used a 14 nodes cluster, where each of the 14 nodes was identical Intel core i5 760 2.8 Ghz Processors and had 8 GB of physical memory. There, all nodes were connected to a Giga-bit Ethernet switch and ran a Ubuntu 12.04 LTS 64bit server edition. The JVM version 1.6.0_22 was used. Also, all nodes used mounted HDFS through "hdfs-fuse". The number of Mappers was two per a node.

Input video data had the resolution 1920x1080 (FHD), total frame length was 36000 frames, average of bitrate was 19.4Mbit/s, the length of GOP was 25, the time of video was about 10 minute and the size of video data was 1.35GByte. When two frames were processed, it took 0.355 seconds. When 36000 frames were processed, it took about 21 minute 18 seconds.

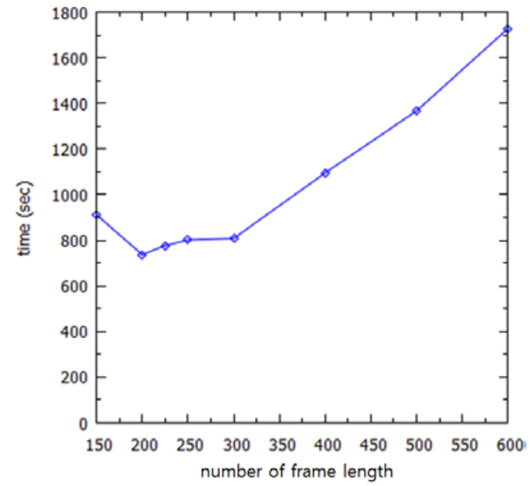

Figure 5. The processing time according to the number of frame length.

Figure 5 shows the results when the frame split size was varied with the fixed input that consisted of 2 videos which had 36000 length of GOP each. It was seen that when the splitting size was smaller than 200, the processing time linearly decreased and when the splitting size was smaller than 200, the processing time was increased.

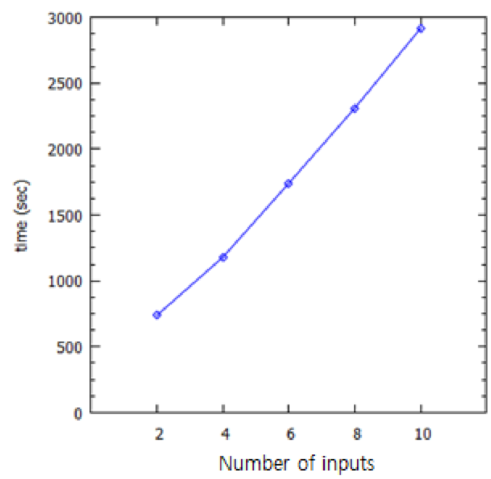

Figure 6. The Processing time according to number of input when the frame split size was 200.

Figure 6 shows the processing time when the frame split size was fixed at 200 and the number of input was varied. There, we see that the processing time increases linearly when the number of input is increased.

VI. RELATED WORK

Video surveillance system usually has the problem of big data generated from video cameras. Traditional distributed system technologies had been unable to solve the big data problem in real time [21][22][23]. The problem seems to be manageable with cloud computing and video surveillance systems seems to have new era.

C.F. Lin et al. [24] proposed a cloud based video recorder system. They propose it to replace traditional video recorder such as the network video recorder (NVR) and the digital video recorder (DVR). They use HDFS as scalable and backupable storage in their distributed replication mechanism and deal with deployment design for public cloud and private/hybrid cloud.

D.A. Rodriguez-Silva et al. [25] suggest traditional video surveillance architecture combined to cloud computing. The system consists of three parts such as the web server, the cloud processing servers and the cloud storage servers. It uses Amazon S3 storage to store video and the SSL protocol. They present a fault-tolerance mechanism which works between processing servers and storage servers.

Y.S. Wu et al. [20] present an architecture to solve the network bottleneck which the centralized video data processing system usually has. It has a well-developed peer to peer (P2P) architecture and HDFS. It proposes its own scheduling algorithm.

M.S. Hossain et al. [26] present a research result regarding virtual machine (VM) resource allocation for cloud-based video surveillance platform. The platform is a general framework of video surveillance service composition in cloud. It deals with linear programming formulation with migration control and simulation using Amazon Web Service (AWS).

R. Pereira et al. [27] propose the Split&Merge architecture to reduce video encoding times, regardless of the video data size. B. White et al. [28] present implementation research of their algorithms in several different kinds of computer systems.

VII. CONCLUSION AND FUTURE WORK

This paper presented a cloud-computing based smart video surveillance system for smart city which uses MapReduce to process a huge amount of video data in real time with the performance evaluation. Here, we used 1920x1080(FHD) resolution video data and HDFS as storage to store video. We analyzed the processing time according to the number of frame per mapper. Tracing the optimal splitting size of input data and the processing time according to the number of nodes showed the linearity in the system performance. We believe our system is very successful in managing smart city. For example, it was very useful in managing urban traffic, fire accident, etc. In our future work, we hope to experiment when the resolution and the other factors of video data was varied with various kinds of distributed storage solutions such as GlusterFS, Lustre and Ceph. We also want to optimize our system, as well as to add various kinds of new functions.

ACKNOWLEDGMENT

This work was supported by the 2014 Research Fund of the University of Seoul (Yong Woo Lee: the corresponding author). Patents in Korea and in the United State of America are registered and are in pending for the contents of this paper.

REFERENCES

[1] H. S. Jung, C. S. Jeong, Y. W. Lee, and P. D. Hong, "An Intelligent Ubiquitous Middleware for U-City: SmartUM," Journal of Information Science and Engineering, vol. 25, Issue 2, Mar. 2009, pp. 375-388.

[2] H. S. Jung, J. K. Baek, C. S. Jeong, Y. W. Lee, and P. D. Hong, "Unified Ubiquitous Middleware for U-City," Proc. International Conference on Convergence Information Technology 2007 (ICCIT 07), Nov. 2007, pp. 2347-2379.

[3] Apache Hadoop, [Online], Nov. 2014, Available from: http://hadoop.apache.org/

[4] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Proc. 6th Symposium on Operating Systems Design and Implementation (OSDI), Dec. 6-8 2004, pp.137-150.

[5] C. H. Yun, H. Han, H. S. Jung, H. Y. Yeom, and Y. W. Lee, "Intelligent Management of Remote Facilities through a Ubiquitous Cloud Middleware," Proc. IEEE International Conference on Cloud Computing (CLOUD 09), Sep. 2009, pp. 65-71.

[6] C. H. Yun, Y. W. Lee, and H. S. Jung, "An Evaluation of Semantic Service Discovery of a U-City Middleware," Proc. 12th International Conference on Advanced Communication Technology (ICACT 10), Feb. 2010, pp. 600-603.

[7] J. W. Park, C. H. Yun, S. G Gyu, H. Y. Yeom, and Y. W. Lee, "Cloud Computing Platform for GIS Image Processing in U-City," Proc. 13th International Conference on Advanced Communication Technology (ICACT 11), Feb. 2011, pp. 1151-1155.

[8] Y. W. Lee and S. W. Rho, "U-City Portal for Smart Ubiquitous Middleware," Proc. 12th International Conference Advanced Communication Technology (ICACT 10), vol. 1, Feb. 2010, pp. 609-613.

[9] S. W. Rho, C. H. Yun, and Y. W. Lee, "Provision of U-city Web Services Using Cloud Computing," Proc. 13th International Conference on Advanced Communication Technology (ICACT 11), Feb. 2011, pp. 1545-1549.

[10] P. D. Hong and Y. W. Lee, "A Grid Portal for Grid Resource Information Service," Proc. 13th International Conference on Advanced Communication Technology (ICACT 11), Feb. 13-16 2011, pp. 597-602.

[11] J. W. Park, C. H. Yun, H. S. Jung, and Y. W. Lee, "Mobile Cloud and Grid Web Service in a Smart City," Proc. The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization, May 25-29 2014, pp. 20-25.

[12] J. W. Park, J. O. Kim, J. W. Park, C. H. Yun, and Y. W. Lee, "Mobile Cloud Web-Service for U-City," Proc. International Conference on Cloud and Green Computing 2011, Dec. 2011, pp.1161-1065.

[13] J. W. Park, C. H. Yun, H. S. Jung, and Y. W. LEE, "Visualization of Urban Air Pollution with Cloud Computing," Proc. IEEE World Congress on Services (SERVICES) 2011, Jul. 4-9 2011, pp. 578-583.

[14] H. K. Park, J. W. Park, C. H. Yun, and Y. W. Lee, "Distributed and parallel processing of noise information for 3D GIS in a U-city," Proc. 13th International Conference on Advanced Communication Technology (ICACT) 2011, Feb. 13-16 2011, pp. 855-858.

[15] J. W. Park et al., "Cloud Computing for Online Visualization of GIS Applications in Ubiquitous City," Proc. Cloud Computing 2010, Nov. 21-26 2010, pp. 170-175.

[16] Y. W. Lee, H. W. Kim, H. Y. Yeom, C. H. Yun, S. W. Rho, and H. S. Jung, "Water Quality Control using E-science Technology," Proc. Fifth IEEE International Conference on e-science (e-Science 09), Dec. 2009.

[17] S. W. Ahn, H. S. Jung, Y. W. Lee, and C. Yoo, "Network Condition Adaptive Real-time Streaming of an Intelligent Ubiquitous Middleware for U-City," Proc. 4th International Conference on Ubiquitous Information Technologies & Applications (ICUT 09), Dec. 2009, pp. 1-5.

[18] E. S. Ryu, H. S. Jung, Y. W. Lee, H. Yoo, C. S. Jeong, and S. W. Ahn, "Content-aware and Network-adaptive Video Streaming of a Ubiquitous Middleware," Proc. 11th International Conference on Advanced Communication Technology (ICACT 09), Feb. 2009, pp. 1458-1461.

[19] S. W. Ahn, H. S. Jung, C. Yoo, and Y. W. Lee, “NARS: Network Bandwidth Adaptive Scalable Real-Time Streaming for Smart Ubiquitous Middleware," Journal of Internet Technology, Vol.14, No.2, Mar. 2013, pp. 217-230.

[20] Y. S. Wu, Y. S. Chang, T. Y. Jang, and J. S. Yen, “An Architecure for Video Surveillance Service based on P2P and Cloud Computing," Proc. Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing(UIC/ATC), Sep. 2012, pp. 661-666.

[21] M. Valera and S. A. Velastin, "Intelligent distributed surveillance systems: a review," Proc. IEE Vision, Image and Signal Processing, vol. 152, no. 2, Apr. 2005, pp. 192-204.

[22] H. Detmold, A. V. D. Hengel, A. R. Dick, K. E. Falkner, D. S. Munro, and R. Morrison, "Middleware for Distributed Video Surveillance," IEEE Distributed systems Online, vol. 9, no. 2, Feb. 2008, pp. 1-10.

[23] A. Hampaur et al., "S3 : The IBM Smart Surveillance System: From Transactional Systems to Observational Systems," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing 2007 (ICASSP 07), Apr. 2007, pp. 1385-1388.

[24] C. F. Lin, S. Yuan, M. Leu, and C. Tsai, "A Framework for Scalable Cloud Video Recorder System in Surveillance Environment," Proc. Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), Sep. 2012, pp. 655-660.

[25] D. A. Rodriguez-Silva, L. Adkinson-Orellana, F.J. Gonz’lez-Castaño, I. Armino-Franco, and D. Gonz’lez-Martinez, "Video Surveillance Based on Cloud Storage," Proc. IEEE Cloud Computing(CLOUD), Jun. 2012, pp. 991-992.

[26] M. S. Hossain, M. M. Hassan, M. A. Qurishi, and A. Alghamdi, "Resource Allocation for Service Composition in Cloud-based Video Surveillance Platform," Proc. IEEE International Conference on Multimedia and Expo Workshops(ICMEW), Jul. 2012, pp. 408-412.

[27] R. Pereira, M. Azambuja, K. Breitman, and M. Endler, "An Architecture for Distributed High Performance Video Processing in the Cloud," Proc. IEEE 3rd International Conference on Cloud Computing, Jul. 2010, pp. 482-489.

[28] B. White, T. Yeh, J. Lin, and L. Daivs, "Web-Scale Computer Vision using MapReduce for Multimedia Data Mining," Proc. the 10th International Workshop on Multimedia Data Mining(MDMKDD), Jul. 2010, pp. 1-10.

 ISBN: 978-1-61208-388-9