

Article

A Cloud-Based UTOPIA Smart Video Surveillance System for Smart Cities

Chel-Sang Yoon, Hae-Sun Jung, Jong-Won Park, Hak-Geun Lee, Chang-Ho Yun
and Yong Woo Lee *

Department of Electrical, Electronic, and Computer Engineering, University of Seoul, #90, John-nong-dong, Dong-dae-mun-gu, Seoul 02504, Korea; csyoon2014@uos.ac.kr (C.-S.Y.); banyasun@uos.ac.kr (H.-S.J.); comics77@uos.ac.kr (J.-W.P.); withteas@uos.ac.kr (H.-G.L.); touch011@uos.ac.kr (C.-H.Y.)

* Correspondence: ywlee@uos.ac.kr

Received: 18 July 2020; Accepted: 16 September 2020; Published: 20 September 2020



Featured Application: The UTOPIA Smart Video Surveillance System can be used for object recognition, tracking, and behavior analysis as well as object detection, and can also use Storm and Spark as well as MapReduce.

Abstract: A smart city is a future city that enables citizens to enjoy Information and Communication Technology (ICT) based smart services with any device, anytime, anywhere. It heavily utilizes Internet of Things. It includes many video cameras to provide various kinds of services for smart cities. Video cameras continuously feed big video data to the smart city system, and smart cities need to process the big video data as fast as it can. This is a very challenging task because big computational power is required to shorten processing time. This paper introduces UTOPIA Smart Video Surveillance, which analyzes the big video images using MapReduce, for smart cities. We implemented the smart video surveillance in our middleware platform. This paper explains its mechanism, implementation, and operation and presents performance evaluation results to confirm that the system worked well and is scalable, efficient, reliable, and flexible.

Keywords: networked video system; smart video surveillance; smart processing; big video data; MapReduce; cloud computing; image processing

1. Introduction

This paper presents a research result for smart cities that converge Information and Communication Technology (ICT) with city infrastructures, the activities in a city, environmental, medical, social, residential, and governmental systems. It provides smart services for citizen care, infrastructure management, traffic management, ecological environment management, energy management, etc. There are definitions for smart cities in global use. The Europe Union defines that “A smart city is a place where traditional networks and services are made more efficient with the use of digital and telecommunication technologies for the benefit of its inhabitants and business.” [1]. The International Telecommunication Union (ITU) defines that “A smart sustainable city is an innovative city that uses information and communication technologies and other means to improve quality of life, efficiency of urban operation and services, and competitiveness, while ensuring that it meets the needs of present and future generations with respect to economic, social, environmental as well as cultural aspects.” [2]. The International Standard Organization (ISO) also has its definition for smart cities.

In smart cities, there are many video cameras that continuously produce big video data to provide smart city services. In general, smart cities need to stream and process a huge amount of video data efficiently and fast, and by many times in real-time [3,4]. Cloud technology can successfully address

this requirement. In particular, Hadoop open source solution [5] with Cloud can be a good solution because it provides a platform to manage data with a simple MapReduce programming model and store data in a Hadoop Distributed File System (HDFS) [6]. It has been observed that the smart cities continuously generate unstructured big data. MapReduce allows us to handle large unstructured data as well as semistructured data. They were previously difficult to process.

This paper presents Cloud-based UTOPIA Smart Video Surveillance (USVS) for smart cities. The Cloud-based USVS system gathers video data with scalable video streaming to effectively and smoothly handle video data flow from numerous video cameras even when network bandwidth is limited and fast process them for object detection with MapReduce models. The USVS is for UTOPIA, which is a smart city paradigm that consists of three tiers: UTOPIA infrastructure tier, UTOPIA middleware platform (SOUL) Tier, and UTOPIA portal tier. Video cameras are deployed in the UTOPIA infrastructure tier. Video cameras in the UTOPIA infrastructure tier can be remotely controlled according to the stored scenarios in the UTOPIA middleware platform tier. The administrator and authorized end users can also remotely control the video cameras in the UTOPIA infrastructure tier through the UTOPIA portal tier. The UTOPIA middleware platform tier plays a central role in the operation of smart cities and plays a pivotal role for USVS. The UTOPIA middleware platform tier has various functions such as Cloud computing, grid computing, ontology based context-aware intelligent processing, stream reasoning, real-time processing, image processing, scalable big data (including video data) streaming, ubiquitous computing, location-based processing, etc., so that UTOPIA can provide a variety of converged intelligent services and enable dynamic serviced execution [3,4]. It has layer architecture so that it can have advantages of layer architecture such as good reusability and maintenance of functionality, easiness to add more functions, easy modification, better environment for scalability and fault tolerance, and better environment to deal with security threats. It provides a user-transparent infrastructure that generates smart city services. It enables users to control remote devices in real-time. It provides a convenient and easy-to-use user interface and supports various kinds of ubiquitous user portals. With a special device interface layer, it does smoothly support various kinds of Internet of Things (IoT) devices including many kinds of video cameras, sensors, sensor networks, and communication protocols for the UTOPIA infrastructure tier. SOUL has an alias name of SmartUM and was first announced in 2007. UTOPIA and SOUL are outputs of a project by the Smart City Consortium, which was made by the Seoul Metropolitan Government, major Korean universities in Seoul such as University of Seoul, Seoul National University, Korea University, Yonsei University, etc., and other organizations in 2005, used over an eight million US dollar research fund during the first research period (2005–2010), and has had more than two hundred participating researchers. SOUL has been continuously evolving since 2010. Further details of SOUL will be explained in Section 3.

With powerful functions and architecture, SOUL has been supporting the following applications as well as the smart surveillance service for UTOPIA.

- Smart traffic management;
- Smart traffic accident management;
- Smart fire management;
- Smart disaster management;
- Smart real-time air-pollution map service;
- Smart real-time noise-map service;
- Smart remote control;
- Smart water management;
- Smart and green environment management;
- Smart building management;
- Smart apartment management;
- Smart security management;
- Smart public service management;

- Smart health care service management;
- Smart medical service management;
- Smart rescue service management;
- Smart city infra-structure management;
- Real-time lost child search service;
- Many other smart city service managements.

The Cloud-based smart video surveillance with scalable video streaming was implemented in SOUL and became a part of SOUL, therefore it inherits all the capabilities, features, and benefits of SOUL. This paper proposes an all-in-one system of USVS, which includes video surveillance, scalable video streaming, Cloud computing, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA), and deals with image processing for object detection with MapReduce. Thus, our work in this paper is novel. We could not find any similar work. Thus, we think this work is unique yet.

This paper gives the following contributions:

1. This paper proposes an all-in-one system of USVS, which includes video surveillance, scalable video streaming, Cloud computing, a middleware platform (SOUL) for smart city, and a smart city paradigm (UTOPIA), and deals with image processing for object detection with MapReduce. The all-in-one system of USVS is unique and novel. This contributes to smart city research.
2. The Cloud-based smart video surveillance with scalable video streaming was implemented in SOUL, thus the middleware of a smart city was expanded to include the Cloud-based smart video surveillance with scalable video streaming. This contributes to the smart city middleware research and smart city research.
3. This paper presents the performance evaluation with our prototype system for USVS, SOUL, and UTOPIA written using Linux Standard Base and Linux Kernel. This paper contributes to the research of performance evaluation by a real measurement in a smart city.

This paper is organized as follows. Section 2 explains related works. Section 3 gives an overview of the USVS. Section 4 presents the mechanism of the USVS. Section 5 explains the big video processing using the USVS in detail. Section 6 presents the performance evaluation. Finally, Section 7 concludes our study.

2. Related Work

Our work in this paper was based on general purpose smart city middleware. A few academic or research papers for general purpose smart city middleware and platform are available. So far, we have not been able to find any work that is completely similar to the work presented in our paper, even though there are good works for video surveillance itself and smart city itself. Thus, we believe that our paper presents a novel and unique system. In this related work section, we selected the works that were useful to understand the work in our paper or supplement the portion that our paper did not deal with or that our paper did not give a detailed explanation for.

A video surveillance system for smart cities must be able to process big data that is continuously generated by video cameras. Traditional distributed system technology could not handle big data properly [7–9]. Cloud computing successfully helps us to manage big data, and novel technology based on the Cloud seems to be a new trend in video surveillance systems.

White et al. [10] present a study on the implementation of image processing algorithms in various types of computer systems. In comparison, our paper does not deal with various computer vision algorithms, but only explains our used algorithm. So, we asked readers who are interested in the image processing algorithms to refer to [10]. White et al. [10] present their study for data mining and do not deal with a smart city and any middleware for smart cities, but our paper is for a smart city and based on a middleware for smart cities.

Pereira et al. [11] suggest split and merge architecture that reduces video encoding time. Our paper presents a middleware platform based approach for smart cities, but Pereira et al. [11] do not.

Mehboob et al. [12] propose an algorithm for 3D conversion from traffic video content to Google Map. Time-stamped glyph-based visualization is used in outdoor surveillance videos for the algorithm, which can be used for event-aware detection. In comparison, our paper dealt with object detection with MapReduce, but Mehboob et al. do not, and our paper dealt with real-time scalable video streaming using a smart city middleware, but Mehboob et al. do not.

Memos et al. [13] propose an algorithm for a media-based surveillance system to improve media security and privacy of wireless sensor networks in the Internet of Things (IoT) network for a smart city framework. Compared to their paper, our paper proposed the USVS system as an all-in-one system for object detection with MapReduce rather than an algorithm for a media-based surveillance system and our paper did not deal with media security or the privacy of wireless sensor networks.

Duan et al. [14] discuss the standard for the future deep feature coding to manage the AI-oriented large-scale video, existing techniques, standards, and the possible large-scale video analysis solutions for deep neural network models. In comparison, their paper focuses on deep learning with neural network models, but our paper proposed an all-in-one system of USVS, which includes video surveillance, scalable video streaming, Cloud computing, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA), and dealt with image processing for object detection with MapReduce using USVS.

Tian et al. [15] propose a block-level background modeling algorithm to support a long-term reference structure for efficient surveillance video coding and a rate-distortion optimization for the surveillance source algorithm to process big video data in a smart city. In comparison, their paper focuses on a block-level background modeling algorithm, but our paper proposed an all-in-one system of USVS, which includes video surveillance, scalable video streaming, Cloud computing, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA), and dealt with image processing for object detection with MapReduce using USVS.

Jin et al. [16] present optimization algorithms and scheduling strategies based on an Unmanned Aerial Vehicle (UAV) cluster. In comparison, our paper proposed an all-in-one system of USVS, which includes video surveillance, scalable video streaming, Cloud computing, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA), and dealt with image processing for object detection with MapReduce using USVS.

Calavia et al. [17] propose an intelligent video surveillance system that can detect and identify abnormal and alarming situations by analyzing object movement in smart cities. It does ontology based semantic translation. Compared to our paper, it does not mention any smart city middleware, nor it does mention that it uses Cloud computing.

Hossain et al. [18] present a study on virtual machine resource allocation to support video surveillance platforms that uses the Amazon Web Service to process linear programming formulas with migration control and simulation. They focus on Cloud computing. Compared to our paper, their work does not deal with MapReduce, scalable video streaming, and any smart city middleware. They do not mention smart city. On the other hand, our paper proposed an all-in-one system of USVS, which includes video surveillance, scalable video streaming, Cloud computing, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA), and dealt with image processing for object detection with MapReduce using USVS.

Lin et al. [19] propose a video recorder system based on Cloud. A digital video recorder and network video recorder with their methodology are suggested. Their distributed replication mechanism uses the Hadoop distributed file system and handles the deployment design of public and private/hybrid Clouds. In comparison, they focus on the video recorder system and do not deal with image processing for object detection, but we did, and they do not mention MapReduce but we did use MapReduce. We proposed an all-in-one system of USVS, which includes video surveillance,

scalable video streaming, Cloud computing, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA), but they do not.

Wu et al. [20] propose a system that has a peer-to-peer architecture with the Hadoop distributed file system to solve the network bottlenecks that centralized video data processing systems typically have. They divide the input video source by chunk size, however we divided the input video source by the video static Group-Of-Picture (GOP) length. The advantage of our approach is explained in detail in Section 5. They focus on the P2P and Cloud in their architecture and its operation, but they do not mention MapReduce, any smart city middleware, or a smart city. We proposed an all-in-one system of USVS, which includes video surveillance, scalable video streaming, Cloud computing, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA).

Rodriguez-Silva et al. [21] propose a Cloud-based traditional video surveillance architecture that consists of a Cloud processing server, a Cloud storage server and a web server. They stored a video in Amazon S3 storage. They do not mention any smart city middleware or smart city. On the other hand, our paper proposed an all-in-one system of USVS, which includes video surveillance, scalable video streaming, Cloud computing, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA), and dealt with image processing for object detection with MapReduce using USVS.

Shao et al. [22] present an intelligent processing and utilization solution to big surveillance video data based on the event detection and alarming messages from front-end smart cameras. In comparison, they do not deal with MapReduce and do not mention any smart city middleware. On the other hand, our paper proposed an all-in-one system of USVS, which includes video surveillance, scalable video streaming, Cloud computing, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA), and dealt with image processing for object detection with MapReduce using USVS.

Nasir et al. [23] present fog computing based resource-efficient distributed video summarization over a multi-region fog computing paradigm for IoT-assisted smart cities. In comparison, our paper dealt with a cloud-based system, but Nasir et al. deal with a fog-based system, they use Spark but we used MapReduce, they do not mention smart city middleware but we proposed an all-in-one system, USVS, of video surveillance, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA).

Janakiramaiah et al. [24] present an intelligent video surveillance framework using the deep learning algorithm with convolution neural networks. In comparison, their framework is not a general purpose framework for a smart city and their paper focuses on the deep learning algorithm with convolution neural networks, but our middleware platform was a general purpose middleware platform, and we dealt with MapReduce for object detection using USVS.

Xu et al. [25] present their approach for abnormal visual event detection in smart city surveillance with edge computing, multi-instance learning, and the autoregressive integrated moving average model. In comparison, they do not mention any smart city middleware but we proposed an all-in-one system of USVS, which includes video surveillance, scalable video streaming, Cloud computing, a middleware platform (SOUL) for a smart city, and a smart city paradigm (UTOPIA), and dealt with image processing for object detection with MapReduce using USVS. They use edge computing, but we used Cloud computing. They deal with abnormal visual event detection, but we dealt with object detection. SOUL is going to support edge computing soon.

3. Smart Video Surveillance in UTOPIA

Since 2000, we have developed a smart city model called UTOPIA that has a three-tier architecture to support smart city services. Citizens can enjoy a variety of converged smart city services in UTOPIA. As introduced in the introduction section, UTOPIA consists of three tiers: the UTOPIA infrastructure tier, the UTOPIA middleware platform tier, and the UTOPIA portal tier. In UTOPIA, various IoT technologies are implemented to change each city component into a smart element and all smart elements are connected using various communication technologies to make a super-connected

city, and further, various ICT technologies are applied to operate the super-connected city and give converged smart services to citizens. We call the body of the UTOPIA “the UTOPIA Infrastructure Tier”. The UTOPIA infrastructure tier supports various kinds of IoT devices, video cameras, sensor networks, GPS, etc. UTOPIA collects various types of data through integrated networks from the UTOPIA infrastructure tier.

The UTOPIA middleware platform tier acts as the brain in UTOPIA smart cities. It manages the data received from the UTOPIA infrastructure tier. Various integrated smart city services should be provided online in UTOPIA smart cities and many decisions for the smart city should be made in timely manners. Thus, we developed SOUL middleware platform system. SOUL has a multilayered architecture so that it can have many advantages: component reusability, decreasing complexity, expandability, etc. SOUL has the Common Device Interface (CDI) that supports video cameras, sensor networks, converged networks, GPS, etc. The CDI supports variable kinds of protocols and provides a gateway function for SOUL. The CDI collects data through scalable video streaming, and sends the collected data to the smart city video analyzer [3,4,26]. SOUL processes the acquired data to smart services with ontology-based context data management by the Intelligent Processing Layer (IPL) that has an inference engine [3,27]. The inference engine enables automatic service discovery, service deployment, and service execution, based on the inferred results [3,27]. The Smart Processing Layer (SPL) supports Cloud computing and grid computing and does various convergence processing to create smart services for various applications [3,28]. The Application Interface Layer (API) supports the UTOPIA portal with an easy-to-use and convenient user interface [3]. It supports various types of smart city applications and user devices [3]. Using the functions of this layer, the user can control remote IoT devices such as IoT fire door and other emergency remote IoT devices in the real-time mode.

At the UTOPIA portal tier, smart city end-users use smart city services interactively and smart city administrators and smart city operators can operate the smart city. Users at the portal tier can use all kinds of user devices including mobile devices such as smartphones, tablet computers, etc. UTOPIA supports ubiquitous computing so that users at the portal tier can use the smart city services anytime, anywhere, and with any device. Authorized users at the smart city portal can control remote devices: remote networked video cameras, emergency devices, fire doors, etc. [3,26,29–37]. The content in this paper utilized the result of our previous accumulated works. For a detailed explanation of content not essential to the main content of this paper, we refer you to the reference given in this paper [3,4,26–40].

USVS in Figure 1 works as follows. The video cameras in the UTOPIA infrastructure tier make videos for smart cities. The video is streamed according to the available network bandwidth of the computer network connected to the video camera. USVS saves network resources by using the network adaptive scalable video streaming [38–40]. The streaming service is provided by the Scalable Video Streaming (SVS) in the UTOPIA middleware platform tier. The video manager provides functions to control all of these processes, and enables smart city system administrators to control SVS remotely. The Smart City Video Analyzer (SCVA) analyzes the video received through the SVS with the help of the Cloud Computing Platform (CCP). The Smart City Video Analyzer is used for object detection, object recognition, tracking, and behavior analysis functions. In this paper, we present the results of research on object detection using MapReduce. For distributed parallel processing using MapReduce, UTOPIA’s Cloud Computing facility in Figure 1 was used.

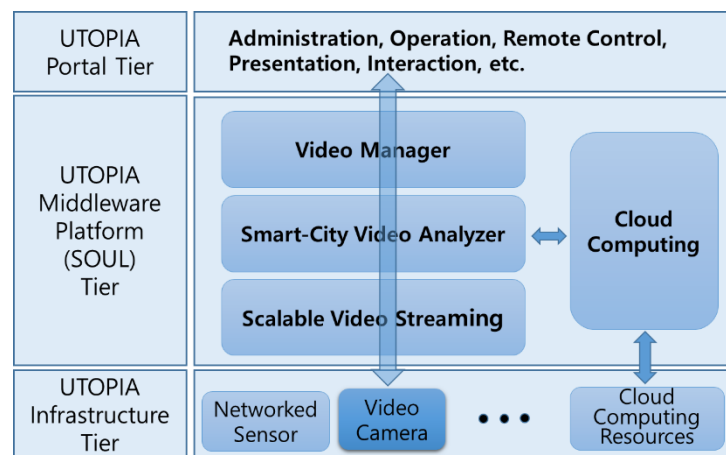


Figure 1. UTOPIA Smart Video Surveillance (USVS).

4. Video Surveillance System Architecture

The USVS system consists of SVS, a Smart-City Video Analyzer (SCVA), and a Video Manager (VM) in the SOUL middleware platform tier, as shown in Figure 2. The VM manages the SVS and the SCVA. The Camera Manager (CM) manages and monitors the status of cameras connected to the smart city. The Storage Manager (SM) monitors the status of the storage in the SCVA. The Analyzer Manager (AM) manages the frame rates, codec, and resolution of video at the user's request and monitors the status of the MapReduce Analyzer (MA). The Protocol Manager (PM) manages video streaming protocols. The real time messaging protocol and real time streaming protocol were supported.

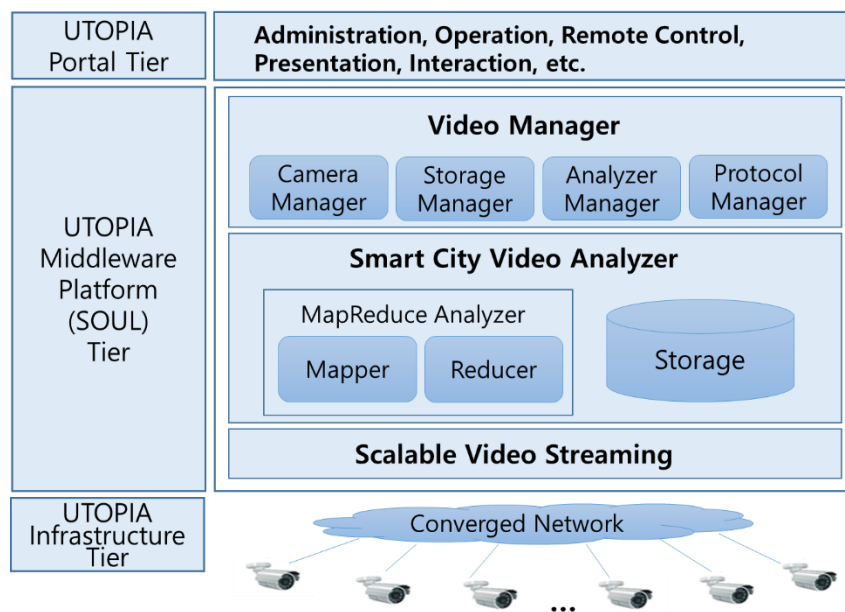


Figure 2. The architecture of USVS.

The SCVA had components of the Streaming Receiver (SR), the storage, and the MA. The SR receives video streams from the SVS to collect video data over converged networks from scattered video cameras. HDFS was used by the SR to store large video data. In the next section, the MA is described in detail.

Many video cameras are required to efficiently operate a smart city in smart cities. So, the USVS system was designed to efficiently manage big video data, and its scalable video streaming component enables smooth management of big video data traffic from numerous network video cameras.

Figure 3 shows that it uses a novel mechanism to control the admissions of clients, extracts the bit stream, and allocates appropriate channels for context-aware and network-adaptive video streaming. Self-learning using scalable video coding techniques saves the system and network resources [40].

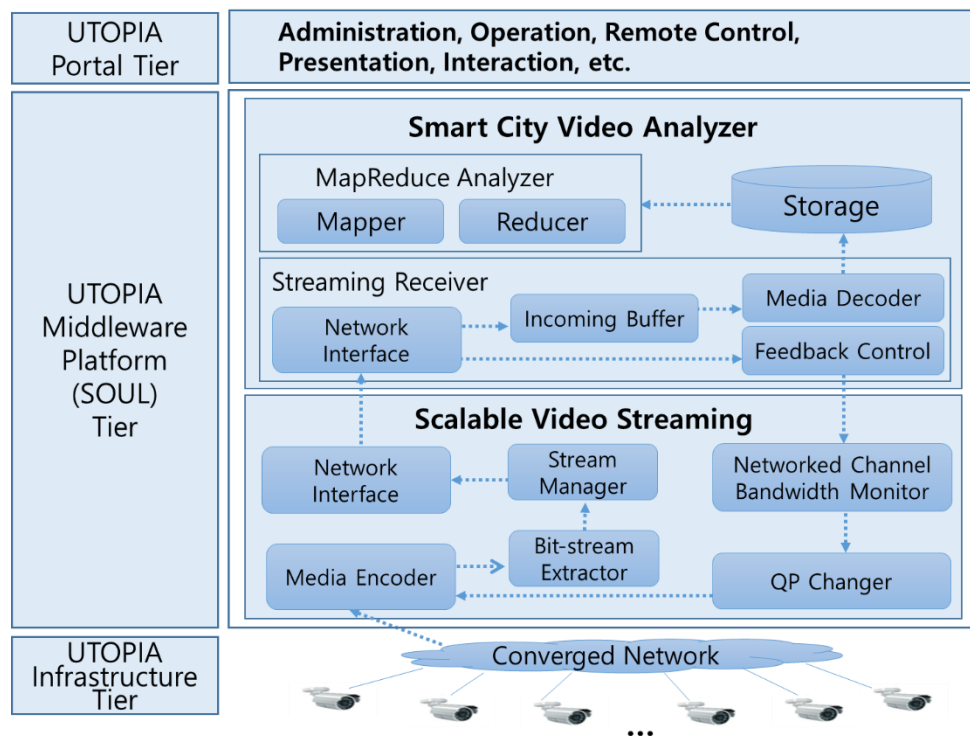


Figure 3. The operation of the USVS.

The working mechanism of the SVS is explained now. The data from each networked video camera in the smart city are encoded by a Media Encoder (ME). The SVS checks the change instructions from the Quantization Parameter (QP) changer when encoding the data. The QP changer encodes the obtained data with various QP values determined using the information of the Network Channel Bandwidth Monitor (NCBM). The NCBM plays the role of a network analyzer. It uses the available bandwidth of the Streaming Receiver (SR), the strength of radio wave signals, and a feedback signal to analyze the network bandwidth in real-time. When initialized, the SVS sends a message to the SR of the SCVA and the Feedback Control (FC) of the SR respond with available bandwidth. The SR thinks that the network channel is in good condition, if the signal is strong. Conversely, the SR thinks that the network channel is in poor condition, if the signal is weak. The FC of the SR helps the NCBM evaluate the condition of the network channel. The SR responds after receiving data. If the channel condition changes very often, the FC sends the information at a minimum until the channels stabilize. In order to evaluate the network channel, this information is used again. The FC sends the information of the current bandwidth of SR to the SVS. The SVS determines the QP value using this information. The NCBM continuously monitors the condition of the network channel. The NCBM evaluates the available bandwidth and communicates with the QP changer to control the encoding rate according to a predetermined QP value. After it, the QP changer adjusts the encoding level. For network adaptive live streaming, live video data are encoded with the determined QP value by the ME. After encoding, the encoded video data are transmitted to the Bit-stream Extractor (BE) and the Stream Manager (STM), which packetize data appropriately. Finally, the data packets are separated for adaptation to various network conditions.

5. Cloud-Base Big Video Data Processing

Figure 4 shows the video data processing by the MapReduce Analyzer (MA). Firstly, video input data are divided by static offset. Our approach to divide input video data is different from others' approach. For example, Wu et al. [20] divide the input video source by chunk size, however we divided the input video source by the video static Group-Of-Picture (GOP) length. When video data are stored, it is usually compressed and stored using a codec. Special care must be taken when dividing a video data compressed in GOP units using a codec. GOP consists of a key frame and a non-key frame. The key frame is a frame that is independently restored without referring to other frames and is also called an Infra Frame (I-frame). The non-key frame has only information created by the difference between the current frame and the previous I-frame. Therefore, when dividing video data without considering the GOP, the information of the I-frame, which is a key frame, is lost, and the data of other frames that refer to the key-frame cannot be read. As a result, the video data is damaged. To prevent this, the compressed file can be restored and then divided, but it is inefficient because it takes a very long time to process the encoding operation. Therefore, in this study, in order to prevent the key frame from being lost the video data was divided by the GOP unit. When dividing video data by the GOP unit, there are several other advantages, and there are other considerations, but they were not explained in this paper.

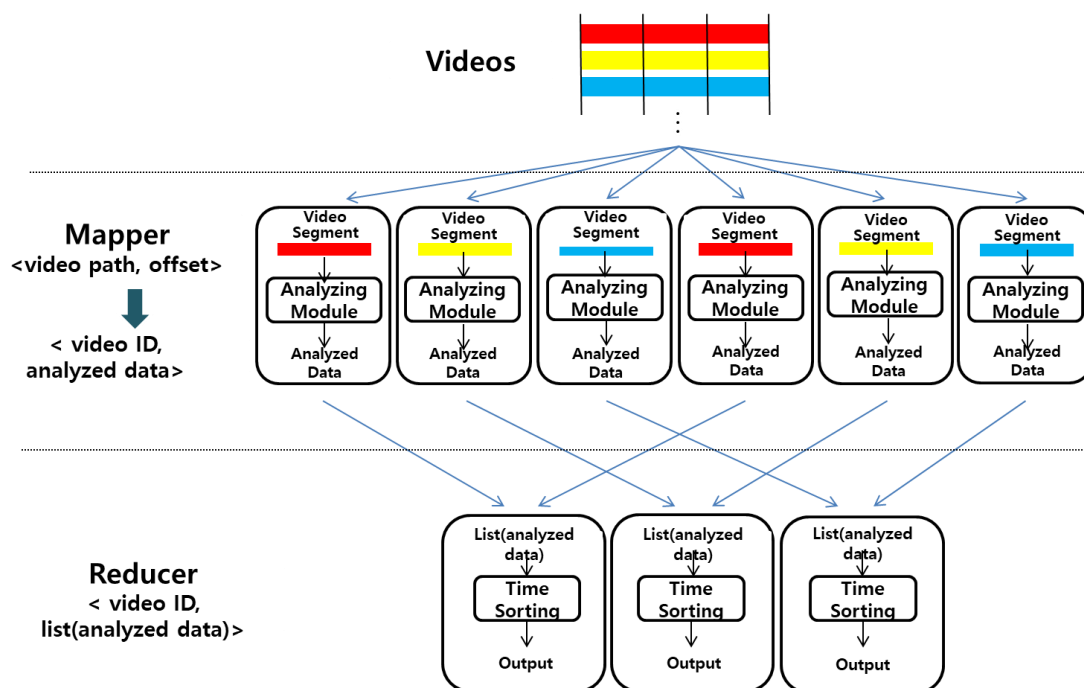


Figure 4. The video data processing by the MapReduce Analyzer.

The MA divides each video source into segments by GOP units and provides them to the Mapper with the key-value pairs of the path information of each divided segment and the offset. Table 1 shows the key-value pairs of our MA. Secondly, the Mapper does its given work. Thirdly, the Mapper produces an output in key-value pairs of “video ID” and “analyzed data”. Fourthly, the intermediate data are sorted by “video ID”. Fifthly and finally, the Reducer does receive the key value pair of “video ID, list (analyzed data)” does its given work, sorts the analyzed data by time, and writes the output to HDFS.

Table 1. The key-value pairs used for the MapReduce Analyzer.

Step	I/O	<Key, Value>	Operation
Map Step	Input	<video path, offset>	Dividing the input video and analyzing the divided portions.
Reduce Step	Output	<video ID, analyzed data>	
	Input	<video ID, list (analyzed data)>	Sorting the analyzed input by time.

The MA can do video processing such as object detection, object recognition, object tracking, and behavior analysis. The functional modules in the MA can be replaced so that the administrator can select the appropriate Mapper module and a Reducer module from UTOPIA's library of the video surveillance system according to smart city requirements and predefined scenarios.

The video surveillance system processes big video data with UTOPIA's Cloud computing system. Figure 5 shows that when a cloud computing service is required, the USVS can request the cloud computing service through the Cloud Computing Platform (CCP) to receive the required service. The Cloud Computing Engine (CCE) of the CCP provides Cloud computing services to the USVS by managing Cloud computing resources.

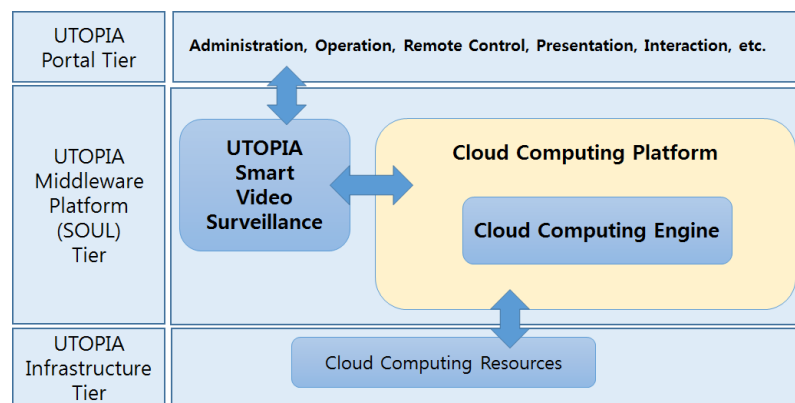
**Figure 5.** UTOPIA's Cloud computing.

Figure 6 shows a brief architecture of the UTOPIA CCE. UTOPIA's Cloud supports both the domestic Cloud and the foreign Cloud and both the private Cloud and public Cloud. The domestic Cloud refers to having Cloud computing resources inside the UTOPIA infrastructure. The foreign Cloud refers to having Cloud computing resources outside the UTOPIA Infrastructure. Various types of hybrid Cloud computing are possible through the aggregation of them in UTOPIA. The UTOPIA CCE leases computing resources based on a VM (Virtual Machine) infrastructure and usually supports the master–slave deployment setting as shown in Figure 7. Users such as the system programmer specify the required computing power using the user menu provided by the UTOPIA CCP. Users can fix the requested computing power or ask an intelligent service to adjust the computing power to the required computing power automatically and dynamically after the first run.

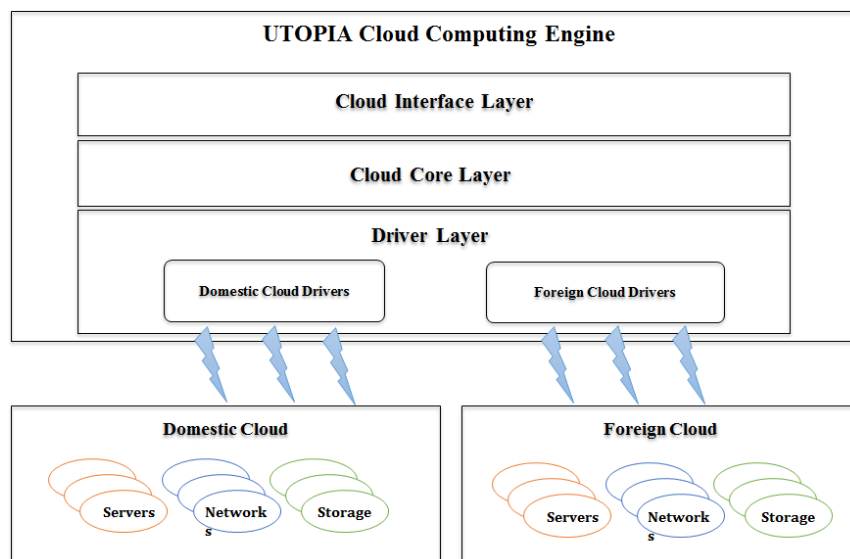


Figure 6. The architecture of the UTOPIA Cloud Computing Engine.

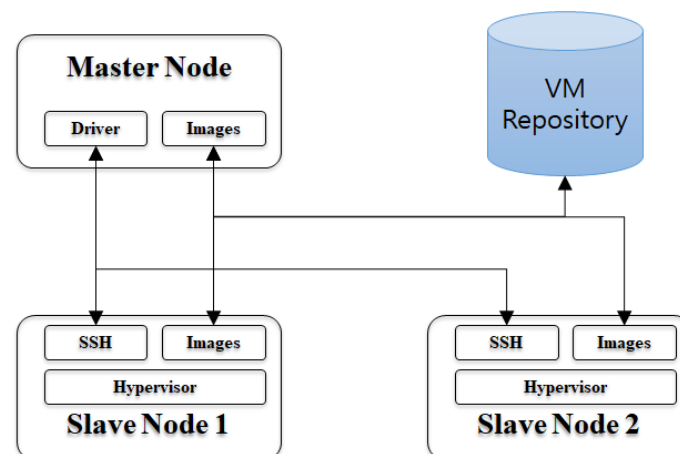


Figure 7. The Cloud computing deployment for domestic Cloud computing.

6. Performance

Detecting a person in a given image was performed in our experiments. Two approaches are the most widely used for object detection: one is the temporal difference method and the other is the background subtraction method [7,41–44]. Our experiments use the temporal difference method, even though our USVS can use any algorithm. The temporal difference method is known to have good performance in dynamic environments because it is very adaptive, but the background subtraction method is known to have good performance in case of extracting object information. In UTOPIA, we selected and used an appropriate algorithm according to our requirement in each case. Our main concern in the performance evaluation experiments of this paper was neither to evaluate the performance of any specific algorithm, nor to compare the performance of different kinds of algorithm, but to evaluate the overall performance of our suggested all-in-one system of USVS with SOUL in UTOPIA. Detailed information regarding experiments, experiment results, and the performances of various algorithms and comparison among various kinds of algorithms is also well explained in [45–51]. Figure 8 shows the flowchart of the temporal difference method used for our experiments. The algorithm works as follows. Firstly, video input data are divided by static offset to get working frames. Then, a difference image is obtained by calculating the difference of successive frames. Next, the algorithm converts a difference image to a gray-scale image. Finally, the object is extracted by converting the gray-scale image into a binary image using a threshold value.

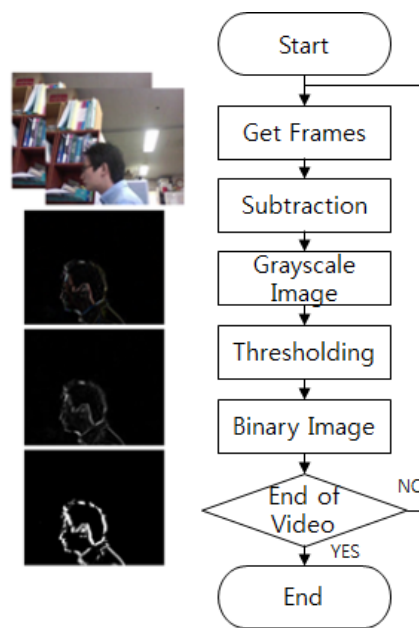


Figure 8. Flow chart of the object detection algorithm implemented for performance evaluation experiments.

We conducted performance evaluation experiments on a Linux-based UTOPIA prototype system with the Linux Standard Base (LSB) in the experimental environment that resembled actual field environments. Currently the ISO Linux Standard Group (WG24) under JTC1/SC22 is making new LSB standards for compatibility among different Linux distributions. Our experiments used an open-source for MPEG H.264 codec, “x264 codec”, and libavutil and ffprobe in FFmpeg library written in Java. We made the MapReduce Analyzer in C++ using OpenCV 2.4.3 since we think OpenCV 2.4.3 is well made and good in performance. The MapReduce Analyzer uses the Apache Hadoop MapReduce environment 0.23.0 that is written in Java.

The characteristics of the workload used for our experiments are as follows. The video resolution was 1920×1080 (FHD: Full High Definition). For a workload, video files from networked video cameras were collected in the “Storage” of USVS for 10 min with 19.4 Mbps transfer rate using the UTOPIA prototype system. The networked cameras send video data to USVS and USUV collected the workload online and processed it online in real-time as explained in previous sections in the UTOPIA prototype system. A workload contains a total of 36,000 frames. The size of a workload is 1.35 Giga Bytes. For our experiments, we divided the video data by the 25 frames GOP length, to prevent losing the key frame. When two frames were executed on one node, the execution time of a frame was 0.035 s, and when a workload that contained 36,000 frames was executed on one node, the total execution time was 1278 s.

From our Cloud IaaS (Infra-structure as a Service) facility, we used a cluster system, which consisted of fourteen homogeneous distributed nodes where each node had Intel i5 760 2.8 GHz Quad-Core processors, DDR3 8 GB RAM, and a 7200 rpm 500 GB hard disk. A Giga-bit Ethernet switch connects all nodes as shown in Figure 9. The cluster system used a homogenous Ubuntu 12.04 LTS 64bit server edition and the homogenous JVM version 1.6.0_31 for the experiments. The HDFS was mounted using “hdfs-fuse” in all nodes.

We performed experiments using the UTOPIA prototype system in the experimental environment to confirm that the developed USVS system works properly and efficiently. Figure 10 shows how the total execution time changed when the number of frames allocated to each Map task was increased. The total execution (or processing) time means the time period from the time when USVS starts to receive the workload from the given networked video cameras to the time when USVS finishes the processing for the given workload. Two workloads were fed at the same time, so, it means that 72,000 frames were given and the size of the total workload was 2.7 Giga Bytes. Fourteen nodes of the

cluster system were used in the experiments. When the number of frames allocated to each map task increased, the total processing time in USYS decreased until it reached 200 frames. After that, the total processing time began to increase. Thus we know that there was the optimal number of allocated frames per map task that gives the best total processing time and the optimal number of allocated frames per map task was 200 in the experiments. We think that the optimal number of allocated frames per map task will be different if we change the experiment environments such as the system configuration and the workload.

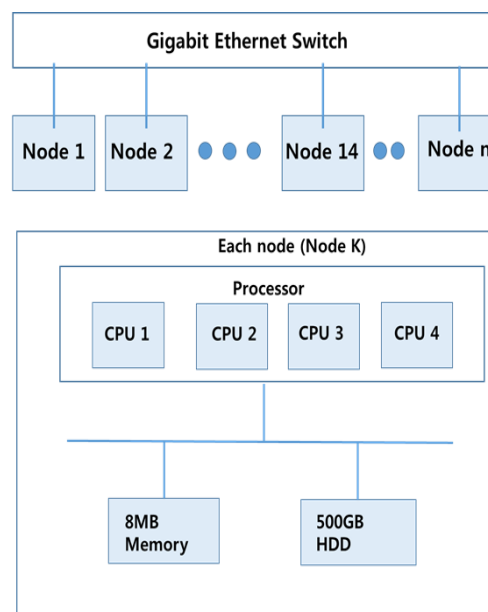


Figure 9. Configuration of the cluster system for Cloud computing in performance evaluation experiments.

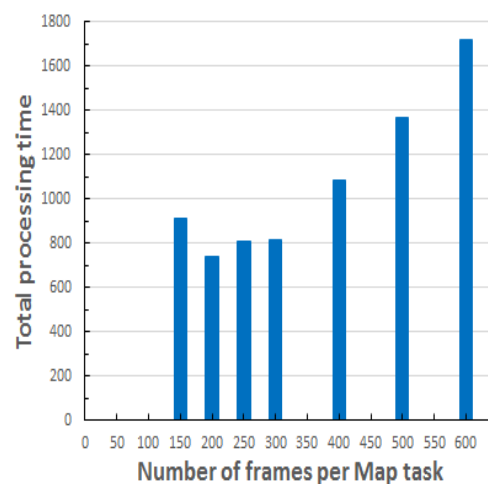


Figure 10. The total processing time when the number of frames allocated to each map task was increased.

We also performed experiments using the UTOPIA prototype system in the experimental environment to check the scalability of the developed USVS. Figure 11 shows the total processing time in USYS when the number of workloads was increased. In the experiments, the number of frames allocated to each map task was fixed to 200, which was the optimal number of frames per map task, 14 nodes of the cluster system were used, and we increased the number of workloads by 2. The Figure 11 shows that the “total processing time” increased linearly. Thus we confirm that the scalability of our USVS was good.

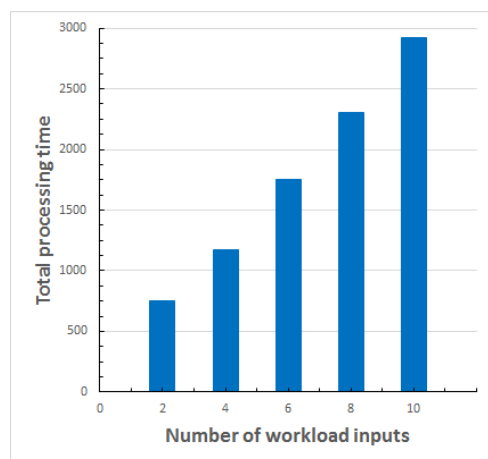


Figure 11. The total processing time when the number of workloads was increased: the number of frames allocated to each map task was fixed at 200.

7. Conclusions

In this paper, we introduced the Cloud-based UTOPIA Smart Video Surveillance (USVS) that uses MapReduce to manage big video data. The USVS system can gather huge video data from various kinds of video cameras through Scalable Video Streaming (SVS) so that the data can be effectively and smoothly managed even when network bandwidth is limited and fast processed with a novel MapReduce method.

We implemented our Cloud-based Smart Video Surveillance in UTOPIA, which consists of three tiers: the UTOPIA infrastructure tier, the UTOPIA SOUL middleware platform tier, and the UTOPIA portal tier. Video cameras are deployed in the UTOPIA infrastructure tier and can be remotely controlled according to the stored scenarios and remote users. The SOUL middleware platform tier is the brain of UTOPIA, includes USVS and provides various smart convergence solutions and functions including Cloud computing. The USVS has the VM, the SCVA, and the SVS in the SOUL middleware platform tier. The VM has the CM, the SM, the AM, and the PM. The VM manages the SCVA and the SVS. The SCVA has the MA, the storage, and the SR.

Detecting a person in a given image was performed for the performance evaluation experiments. Libavutil and fprobe in the FFmpeg library written in Java were used with an open-source for MPEG H.264 codec, “x264 codec”, and MA was implemented in C++ using OpenCV 2.4.3 with the Apache Hadoop MapReduce environment 0.23.0. The experiments were performed in a cluster system, that used a Giga-bit Ethernet switch for connection and had fourteen homogeneous distributed nodes where each node had an Intel i5 760 2.8 GHz Quad-Core processors, DDR3 8 GB RAM, and a 7200 rpm 500 GB hard disk with a homogenous Ubuntu 12.04 LTS 64bit server edition and a homogenous JVM version 1.6.0_31, where the HDFS was mounted using “hdfs-fuse” in all nodes. In our experiment, we divided the video data by the 25 frames GOP length so that we did not lose the key frame. The video resolution was 1920×1080 (FHD) and the used video workload was collected online for 10 min and the workload was processed online in real-time in an experimental UTOPIA environment with SOUL and USVS for performance evaluation. We confirmed that our developed USVS worked well with good performance in the first experiments and the scalability was good in the second experiments. The USVS can be used for object recognition, tracking, and behavior analysis functions, as well. Our USVS can also uses Storm and Spark for real-time processing as well.

Author Contributions: Conceptualization, Y.W.L., C.-S.Y., J.-W.P., H.-G.L., H.-S.J., and C.-H.Y.; methodology, C.-S.Y., Y.W.L., H.-G.L., J.-W.P., C.-H.Y., and H.-S.J.; software, C.-S.Y., H.-G.L., J.-W.P., C.-H.Y., Y.W.L., and H.-S.J.; validation, C.-S.Y., Y.W.L., J.-W.P., H.-G.L. and H.-S.J.; formal analysis, C.-S.Y., Y.W.L., H.-S.J., J.-W.P. and H.-G.L.; investigation, C.-S.Y., H.-S.J., J.-W.P., H.-G.L., and Y.W.L.; resources, C.-S.Y., H.-S.J., J.-W.P., H.-G.L., Y.W.L., and C.-H.Y.; data curation, C.-S.Y., Y.W.L., H.-S.J., J.-W.P. and H.-G.L.; writing—original draft preparation, Y.W.L., C.-S.Y., H.-S.J., J.-W.P., and H.-G.L.; writing—review and editing, Y.W.L., C.-S.Y., H.-S.J.; visualization, C.-S.Y.,

Y.W.L., H.-S.J. and J.-W.P.; supervision, Y.W.L.; project administration, Y.W.L. and H.-S.J.; funding acquisition, Y.W.L. and H.-S.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Technology Innovation Program (20006924) funded By the Ministry of Trade, Industry and Energy (MOTIE, Korea). This work was also supported by the National Research Foundation of Korea (NRF) and the Center for Women in Science, Engineering and Technology (WISSET) Grant funded by the Ministry of Science and ICT (MSIT) under the Program for Returners into R&D.

Acknowledgments: We give thanks to the members of the Smart City Consortium and Seoul Grid Center for their contribution to previous works.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. *What Are Smart Cities?* European Commission: Brussels, Belgium, 2020. Available online: https://ec.europa.eu/info/eu-regional-and-urban-development/topics/cities-and-urban-development/city-initiatives/smart-cities_en (accessed on 31 May 2020).
2. Definition of a Smart Sustainable City, ITU, ITU-T, Smart Sustainable Cities at a Glance. Available online: <https://www.itu.int/en/ITU-T/ssc/Pages/info-ssc.aspx> (accessed on 31 May 2020).
3. Jung, H.S.; Jeong, C.S.; Lee, Y.W.; Hong, P.D. An Intelligent Ubiquitous Middleware for U-City: SmartUM. *J. Inf. Sci. Eng.* **2009**, *25*, 375–388.
4. Jung, H.S.; Baek, J.K.; Jeong, C.S.; Lee, Y.W.; Hong, P.D. Unified Ubiquitous Middleware for U-City. In Proceedings of the 2nd International Conference on Convergence Information Technology 2007 (ICCIT 07), Gyeongju, Korea, 21–23 November 2007; pp. 2347–2379.
5. Apache Hadoop. Available online: <http://hadoop.apache.org/> (accessed on 15 November 2014).
6. Dean, J.; Ghemawat, S. MapReduce: Simplified Data Processing on Large Clusters. In Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI), San Francisco, CA, USA, 6–8 December 2004; pp. 137–150.
7. Valera, M.; Velastin, S.A. Intelligent distributed surveillance systems: A review. *IEEE Proc. Image Signal Process.* **2005**, *152*, 192–204. [[CrossRef](#)]
8. Detmold, H.A.; Hengel, V.D.; Dick, A.R.; Falkner, K.E.; Munro, D.S.; Morrison, R. Middleware for Distributed Video Surveillance. *IEEE Distrib. Syst. Online* **2008**, *9*, 1–10. [[CrossRef](#)]
9. Hampapur, A.; Borger, S.; Brown, L.; Carlson, C.; Connell, J.; Lu, M.; Senior, A.; Reddy, V.; Shu, C.; Tian, Y. S3: The IBM Smart Surveillance System: From Transactional Systems to Observational Systems. In Proceedings of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing 2007 (ICASSP 07), Honolulu, HI, USA, 16–20 April 2007; pp. 1385–1388.
10. White, B.; Yeh, T.; Lin, J.; Davis, L. Web-Scale Computer Vision using MapReduce for Multimedia Data Mining. In Proceedings of the 10th International Workshop on Multimedia Data Mining (MDMKDD), Washington, DC, USA, 25 July 2010; pp. 1–10.
11. Pereira, R.; Azambuja, M.; Breitman, K.; Endler, M. An Architecture for Distributed High Performance Video Processing in the Cloud. In Proceedings of the 3rd IEEE International Conference on Cloud Computing, Miami, FL, USA, 5–10 July 2010; pp. 482–489.
12. Mehboob, F.; Abbas, M.; Rehman, S.; Khan, S.A.; Jiang, R.; Bouridane, A. Glyph-based video visualization on Google Map for surveillance in smart cities. *EURASIP J. Image Video Process.* **2017**, *2017*, 1–16. [[CrossRef](#)]
13. Memos, V.A.; Psannis, K.E.; Ishibashi, Y.; Kim, B.G.; Gupta, B.B. An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework. *Future Gener. Comput. Syst.* **2018**, *83*, 619–628. [[CrossRef](#)]
14. Duan, L.; Lou, Y.; Wang, S.; Gao, W.; Rui, Y. AI-Oriented Large-Scale Video Management for Smart City: Technologies, Standards, and Beyond. *IEEE Multimedia* **2018**, *26*, 8–20. [[CrossRef](#)]
15. Tian, L.; Wang, H.; Zhou, Y.; Peng, C. Video big data in smart city: Background construction and optimization for surveillance video processing. *Future Gener. Comput. Syst.* **2018**, *86*, 1371–1382. [[CrossRef](#)]
16. Jin, Y.; Qian, Z.; Yang, W. UAV Cluster-Based Video Surveillance System Optimization in Heterogeneous Communication of Smart Cities. *IEEE Access* **2020**, *8*, 55654–55664. [[CrossRef](#)]

17. Calavia, L.; Baladrón, C.; Aguiar, J.M.; Carro, B.; Sánchez-Esguevillas, A. A semantic autonomous video surveillance system for dense camera networks in smart cities. *Sensors* **2012**, *12*, 10407–10429. [[CrossRef](#)]
18. Hossain, M.S.; Hassan, M.M.; Qurishi, M.A.; Alghamdi, A. Resource Allocation for Service Composition in Cloud-Based Video Surveillance Platform. In Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Melbourne, VIC, Australia, 9–13 July 2012; pp. 408–412.
19. Lin, C.F.; Yuan, S.M.; Leu, M.C.; Tsai, C.T. A Framework for Scalable Cloud Video Recorder System in Surveillance Environment. In Proceedings of the 9th International Conference on Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), Fukuoka, Japan, 4–7 September 2012; pp. 655–660.
20. Wu, Y.S.; Chang, Y.S.; Jang, T.Y.; Yen, J.S. An Architecture for Video Surveillance Service based on P2P and Cloud Computing. In Proceedings of the 9th International Conference on Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), Fukuoka, Japan, 4–7 September 2012; pp. 661–666.
21. Rodriguez-Silva, D.A.; Adkinson-Orellana, L.; González-Castaño, F.J.; Armino-Franco, I.; González-Martínez, D. Video Surveillance Based on Cloud Storage. In Proceedings of the 5th IEEE Cloud Computing (CLOUD 2012), Honolulu, HI, USA, 24–29 June 2012; pp. 991–992.
22. Shao, Z.; Cai, J.; Wang, Z. Smart monitoring cameras driven intelligent processing to big surveillance video data. *IEEE Trans. Big Data* **2017**, *4*, 105–116. [[CrossRef](#)]
23. Nasir, M.; Muhammad, K.; Lloret, J.; Sangaiah, A.K.; Sajjad, M. Fog computing enabled cost-effective distributed summarization of surveillance videos for smart cities. *J. Parallel Distrib. Comput.* **2019**, *126*, 161–170. [[CrossRef](#)]
24. Janakiramaiah, B.; Kalyani, G.; Jayalakshmi, A. Automatic alert generation in a surveillance systems for smart city environment using deep learning algorithm. *Evol. Intell.* **2020**, *1*–8. [[CrossRef](#)]
25. Xu, X.; Liu, L.; Zhang, L.; Li, P.; Chen, J. Abnormal visual event detection based on multi-instance learning and autoregressive integrated moving average model in edge-based Smart City surveillance. *Softw. Exp.* **2020**, *50*, 476–488. [[CrossRef](#)]
26. Yun, C.H.; Han, H.; Jung, H.S.; Yeom, H.Y.; Lee, Y.W. Intelligent Management of Remote Facilities through a Ubiquitous Cloud Middleware. In Proceedings of the 2nd IEEE International Conference on Cloud Computing (CLOUD 09), Bangalore, India, 21–25 September 2009; pp. 65–71.
27. Yun, C.H.; Lee, Y.W.; Jung, H.S. An Evaluation of Semantic Service Discovery of a U-City Middleware. In Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT 10), Gangwon-Do, Korea, 7–10 February 2010; pp. 600–603.
28. Park, J.W.; Yun, C.H.; Gyu, S.G.; Yeom, H.Y.; Lee, Y.W. Cloud Computing Platform for GIS Image Processing in U-City. In Proceedings of the 13th International Conference on Advanced Communication Technology (ICACT 11), Gangwon-Do, Korea, 13–16 February 2011; pp. 1151–1155.
29. Lee, Y.W.; Rho, S.W. U-City Portal for Smart Ubiquitous Middleware. In Proceedings of the 12th International Conference Advanced Communication Technology (ICACT 10), Gangwon-Do, Korea, 7–10 February 2010; pp. 609–613.
30. Rho, S.W.; Yun, C.H.; Lee, Y.W. Provision of U-city Web Services Using Cloud Computing. In Proceedings of the 13th International Conference on Advanced Communication Technology (ICACT 11), Gangwon-Do, Korea, 13–16 February 2011; pp. 1545–1549.
31. Hong, P.D.; Lee, Y.W. A Grid Portal for Grid Resource Information Service. In Proceedings of the 13th International Conference on Advanced Communication Technology (ICACT 11), Gangwon-Do, Korea, 13–16 February 2011; pp. 597–602.
32. Park, J.W.; Yun, C.H.; Jung, H.S.; Lee, Y.W. Mobile Cloud and Grid Web Service in a Smart City. In Proceedings of the 5th International Conference on Cloud Computing, GRIDs, and Virtualization, Venice, Italy, 25–29 May 2014; pp. 20–25.
33. Park, J.W.; Kim, J.O.; Park, J.W.; Yun, C.H.; Lee, Y.W. Mobile Cloud Web-Service for U-City. In Proceedings of the 9th International Conference on Cloud and Green Computing 2011, Sydney, NSW, Australia, 12–14 December 2011; pp. 1601–1605.
34. Park, J.W.; Yun, C.H.; Jung, H.S.; Lee, Y.W. Visualization of Urban Air Pollution with Cloud Computing. In Proceedings of the 7th IEEE World Congress on Services (SERVICES) 2011, Washington, DC, USA, 4–9 July 2011; pp. 578–583.

35. Park, H.K.; Park, J.W.; Yun, C.H.; Lee, Y.W. Distributed and parallel processing of noise information for 3D GIS in a U-city. In Proceedings of the 13th International Conference on Advanced Communication Technology (ICACT) 2011, Gangwon-Do, Korea, 13–16 February 2011; pp. 855–858.
36. Park, J.W.; Lee, Y.W.; Yun, C.H.; Park, H.K.; Chang, S.I.; Lee, I.P.; Jung, H.S. Cloud Computing for Online Visualization of GIS Applications in Ubiquitous City. In Proceedings of the 1st Cloud Computing 2010, Lisbon, Portugal, 21–26 November 2010; pp. 170–175.
37. Lee, Y.W.; Kim, H.W.; Yeom, H.Y.; Yun, C.H.; Rho, S.W.; Jung, H.S. Water Quality Control using E-science Technology. In Proceedings of the 5th IEEE International Conference on e-science (e-Science 09), Oxford, UK, 9–11 December 2009.
38. Ahn, S.W.; Jung, H.S.; Lee, Y.W.; Yoo, C. Network Condition Adaptive Real-time Streaming of an Intelligent Ubiquitous Middleware for U-City. In Proceedings of the 4th International Conference on Ubiquitous Information Technologies & Applications (ICUT 09), Fukuoka, Japan, 20–22 December 2009; pp. 1–5.
39. Ryu, E.S.; Jung, H.S.; Lee, Y.W.; Yoo, H.; Jeong, C.S.; Ahn, S.W. Content-Aware and Network-Adaptive Video Streaming of a Ubiquitous Middleware. In Proceedings of the 11th International Conference on Advanced Communication Technology (ICACT 09), Gangwon-Do, Korea, 15–18 February 2009; pp. 1458–1461.
40. Ahn, S.W.; Jung, H.S.; Yoo, C.; Lee, Y.W. NARS: Network Bandwidth Adaptive Scalable Real-Time Streaming for Smart Ubiquitous Middleware. *J. Internet Technol.* **2013**, *14*, 217–230. [\[CrossRef\]](#)
41. Zheng, Y.; Fan, L. Moving object detection based on running average background and temporal difference. In Proceedings of the IEEE International Conference on Intelligent Systems and Knowledge Engineering, Hangzhou, China, 15–16 November 2010; pp. 270–272.
42. Parekh, H.S.; Thakore, D.G.; Jaliya, U.K. A Survey on Object Detection and Tracking Methods. *Int. J. Innov. Res. Comput. Commun. Eng.* **2014**, *2*, 2970–2978.
43. Kumar, S.; Yadav, J.S. Background Subtraction Method for Object Detection and Tracking. In Proceedings of the International Conference on Intelligent Communication, Control and Devices (ICICCD 2016), Dehradun, India, 2–3 April 2016; pp. 1057–1063.
44. Maddalena, L.; Petrosino, A. Background subtraction for moving object detection in RGBD data: A survey. *J. Imaging* **2018**, *4*, 71. [\[CrossRef\]](#)
45. Ojala, T.; Pietikainen, M.; Harwood, D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel, 9–13 October 1994; pp. 582–585.
46. Cruz, J.; Shiguemori, E.; Guimarães, L. A comparison of Haar-like, LBP and HOG approaches to concrete and asphalt runway detection in high resolution imagery. *Int. Sci. J. Comp. Int. Sci.* **2015**, *6*, 121–136. [\[CrossRef\]](#)
47. Sharma, V.; Nain, N.; Badal, T. A survey on moving object detection methods in video surveillance. *Int. Bull. Math. Res.* **2015**, *2*, 209–218.
48. Zafeiriou, S.; Zhang, C.; Zhang, Z. A survey on face detection in the wild: Past, present and future. *Comput. Vis. Image Underst.* **2015**, *138*, 1–24. [\[CrossRef\]](#)
49. Yazdi, M.; Bouwmans, T. New trends on moving object detection in video images captured by a moving camera: A survey. *Comput. Sci. Rev.* **2018**, *28*, 157–177. [\[CrossRef\]](#)
50. Canedo, D.; Neves, A.J. Facial Expression Recognition Using Computer Vision: A Systematic Review. *Appl. Sci.* **2019**, *9*, 4678. [\[CrossRef\]](#)
51. Chapel, M.N.; Bouwmans, T. Moving Objects Detection with a Moving Camera: A Comprehensive Review. *arXiv* **2020**, arXiv:2001.05238v1. preprint.

