Smart City Surveillance in Fog Computing

Ning Chen, Yu Chen, Xinyue Ye, Haibin Ling, Sejun Song and Chin-Tser Huang

Abstract The Internet and Internet of Things (IoT) make the Smart City concept an achievable and attractive proposition. Efficient information abstraction and quick decision making, the most essential parts of situational awareness (SAW), are still complex due to the overwhelming amount of dynamic data and the tight constraints on processing time. In many urban surveillance tasks, powerful Cloud technology cannot satisfy the tight latency tolerance as the servers are allocated far from the sensing platform; in other words there is no guaranteed connection in the emergency situations. Therefore, data processing, information fusion and decision making are required to be executed on-site (i.e., near the data collection locations). Fog Computing, a recently proposed extension of Cloud Computing, enables on-site computing without migrating jobs to a remote Cloud. In this chapter, we firstly introduce the motivations and definition of smart cities as well as the existing challenges. Then the

N. Chen (✉) · Y. Chen
Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY, USA
e-mail: nchen14@binghamton.edu

Y. Chen
e-mail: ychen@binghamton.edu

X. Ye
Department of Geography, Kent State University, Kent, OH, USA
e-mail: xye5@kent.edu

H. Ling
Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA
e-mail: hbling@temple.edu

S. Song
School of Computing and Engineering, University of Missouri-Kansas City, Kansas City, MO, USA
e-mail: songsej@umkc.edu

C.-T. Huang
Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA
e-mail: HUANGCT@cse.sc.edu

C.X. Mavromoustakis et al. (eds.), *Advances in Mobile Cloud Computing and Big Data in the 5G Era*, Studies in Big Data 22,
DOI 10.1007/978-3-319-45145-9_9

concepts and advantages of Fog Computing are discussed. Additionally, we investigate the feasibility of Fog Computing for real-time urban surveillance using speeding traffic detection as a case study. Adopting a drone to monitor the moving vehicles, a Fog Computing prototype is developed. The results validate the effectiveness of our Fog Computing based approach for on-site, online, uninterrupted urban surveillance tasks.

1 Smart City

The world's population has been increasingly concentrated in urban areas at an unprecedented scale and speed [1–3]. This rapid process of urbanization has brought profound influence on the daily life of urban citizens, but at the same time, as the byproducts of urbanization, a series of issues have emerged, such as environmental pollution, energy consumption, urban crimes, and traffic congestion. In addition, urbanization has led to social, environmental, economic, political transformations. All these issues resulted from urbanization process are challenging the city governments, urban planners and stakeholders. In the following sections, the motivations and concepts of smart cities are presented, and the challenges are discussed.

1.1 Motivations and Concepts

The United Nations World Urbanization Prospects reported that the urban population in the world has grown up rapidly from 746 million, 30 % of the world's population in 1950, to 3.9 billion which is 54 % of the total population of the world in 2014 [4]. In addition, the United Nations estimated that the percentage could reach as high as 66 % in 2050 and the urbanization progress will be much faster in some developing counties such as China. This rapid pace has improved the living quality of urban residents through developing physical infrastructure, transportation system, as well as education and health facilities. However, the negative effects have also emerged. A variety of serious ecological and social issues are listed below.

- **Air pollution**: Along with the growing population in urban areas, vehicles are necessary transportation tools providing convenience for citizens. However, as more vehicles running in the cities, huge volumes of vehicle exhaust can be produced every day which is hazardous and causes health issues. Due to rapid economic growth and industrialization in many developing countries, it is noteworthy that industry gases emissions has also played an important role in contributing to air pollution.
- **Traffic congestion**: Traffic congestion in urban areas caused by the huge numbers of vehicles. It leads to long traffic delays, especially during the rush hours. Besides wasting our time in traffic, there are some underlying but more significant impacts.

For example, more fuel is consumed due to the higher burning rate during the traffic jams, which is one of the main sources of air pollution leaving alone the loss of money.

- **Car accidents**: The growing number of vehicles in urban areas could give rise to fatal car accidents as well. A report from Texas Department of Transportation [5] shows that in 2014, 344 people in Texas were killed in crashes involving speed-over limit of which 210 people were the drivers and the other 134 persons were passengers or pedestrians. Therefore, a smart, on-line urban speeding vehicle surveillance system would be helpful to reduce the number of car accidents.

The above urban issues are not exclusive. There could be more concerns relating to economical, religions and political issues. Cities are not the places people simply live together anymore. The urbanization are reflecting the people's pursuing for better life and the big transforms of culture, economic, society and politics. Considering the issues and problems of urbanization, people are seeking for technical solutions to make sustainable developing and environment friendly cities.

The evolution of information and communication technology has provided people the opportunities to solve the urbanization issues. The Internet has been an important part of daily life and various kinds of city dynamics are combined tightly with digital sensors and networking systems. With the ubiquitously deployed sensors and pervasively available computing devices, the cities are largely digitalized. Human mobility, energy consumption, air quality, traffic conditions and other index of city dynamics can be recorded.

In recent years, new networking system and computing architectures have been proposed to deal with the rapid process of urbanization and to provide Internet-based services such as Big Data, Internet of Things (IoT), Cloud Computing, Fog Computing etc. They are different technologies, but also relating to each other. The Internet of Things connects not only digital sensors and devices, but also physical infrastructure together by which people can get real-time data from a remote location and information can be exchanged between devices [6]. The large scale of data collected from ubiquitous digital sensors are valuable to analyze the city and the answers for urban issues can be explored in massive urban data. In big data [7] technologies, the hidden relationship and reasons for urban issues will be exploited.

Grid computing, utility computing, Cloud Computing and Fog Computing are different computing architectures for different applications. The appearance of Cloud Computing has solved the problems such as how to store the massive volume of urban dynamic data and how to analyze the urban big data with powerful computing tools. The data centers in Cloud Computing provide the users more flexibility without the need for capital outlay which reduce the maintenance cost and potential risks. Fog Computing, as the extension of Cloud Computing, enables the computation tasks accomplished at the edge of network which would be ideal for latency sensitive services and in some extreme conditions. The combination of all the developing information and communication technologies can help cities collect data, deliver data to data centers and analyze data for living patterns and city dynamics which takes lots of advantages for urban planning and governance.

Considering the urbanization issues to be solved and the existing technologies, the concept of 'Smart Cities' emerges. Smartness means understanding and learning: understanding the new patterns and learning how to deal with the new patterns just as in smart cities. People leverage the information and communication technologies to collect the complex urban dynamic data from which underlying dynamic patterns would be extracted by data mining techniques to understand the city. From understanding and analyzing of complex urban data, urban planners can make the cities more intelligent and the city governance more efficient. There is not a consensus about smart city definitions. From different perspectives of distinct relative stakeholders, the definitions are differing from each other.

While there are many different definitions for smart city, below is one that provides a clear vision [8]:

> A smart city is a system integration of technological infrastructure that relies on advanced data processing with the goals of making city governance more efficient, citizens happier, businesses more prosperous and the environment more sustainable.

According to the above definition of smart city, the main goal of smart cities is to make citizens happier by utilizing information technologies. Smart cities use data sensing and acquisition technologies to collect data regarding every aspect of cities, data transmission technologies to send the urban data to analysis centers and data mining technologies to fuse and analyze the urban data to extract valuable information.

Figure 1 illustrates a four-layer hierarchical smart city architecture, which consists of: data sensing and acquisition layer, data storage and management layer, data analysis layer and smart applications layer. The first layer at the bottom is the data sensing and acquisition layer. In smart city, the data sensing and acquisition is the fundamental part of the entire system. In this layer, heterogeneous networked sensors are deployed ubiquitously in smart cities to collect dynamic urban data and the advanced communication equipments are utilized to transmit the unstructured urban data to the second layer. The second layer is the data storage and management layer. The dynamic urban data is characterized as massive volume at both spatial and temporal scale. In addition, the data are collected by different sensors in different formats. Therefore, we need a layer to store the big urban data effectively and also are capable of filtering useful data efficiently from heterogeneous data sources.

The third layer is the data analysis layer which is composed of computing centers. The valuable data from data storage layer will be sent to the upper layer for analysis. In the data analysis layer there can be different computing tools. Cloud data centers can be used for long-term analysis and batch processing jobs with powerful computation capability. Fog Computing can be utilized for on-site processing and instant decision making. The top layer is the application layer. Once the computation results are obtained in the data analysis layer, they will be transferred to the appli-

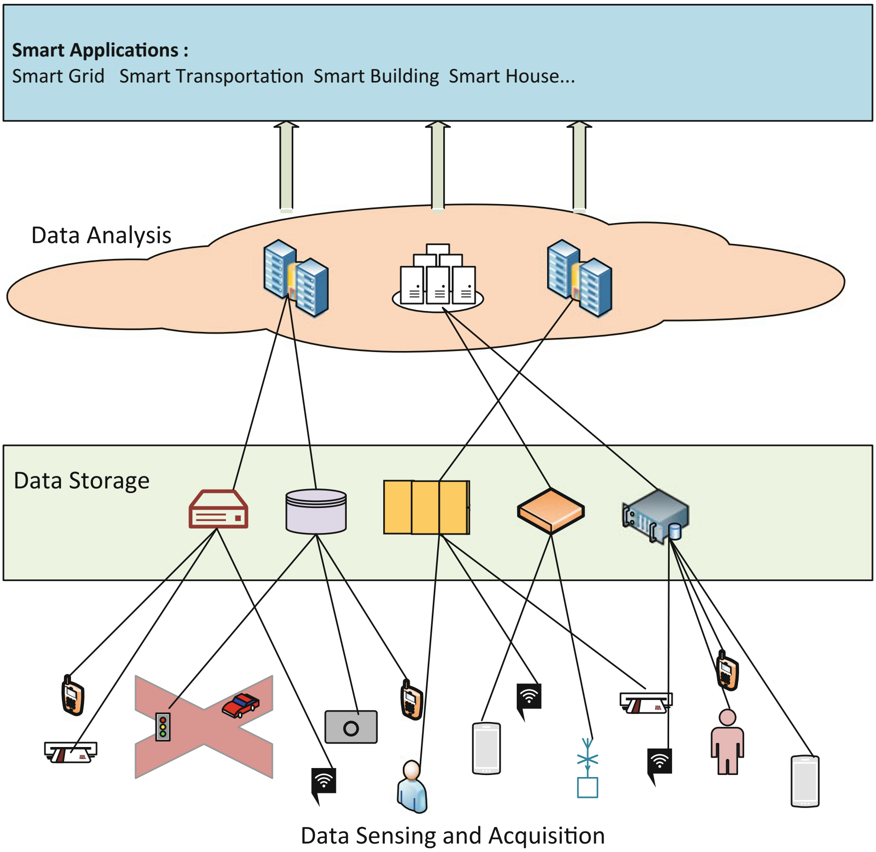


Fig. 1 A four-layer smart city architecture

cation layer. This layer consists of a wide range of smart applications such as smart grid, smart house and smart building, which make the smart city more intelligent.

1.2 Challenges

Smart city is a powerful strategy to deal with the severe problems along with the process of urbanization. Information and communication technologies are adopted to obtain the essential understanding to urban issues. Although the innovative technologies are helpful to improve the situations, the smart city is still facing a lot of challenges. How to obtain effective and correct data sets for certain applications and how to analyze heterogeneous data need to be further discussed. In this section, several major challenges in smart city are presented.

- **Data sensing and acquisition** As discussed above, the data sensing and acquisition are the fundamentals to smart cities since data are depicting the dynamic of cities, but currently problems in city wide data sensing and acquisition do exist. Lack of enough sensors for some specific mission critical missions is still one major constraint, for example, urban surveillance for over-speed traffic monitoring. Installing new digital sensors at a city wide range is a kind of solution but would take the additional power consumption as well. Different to traditionally digital sensors deployed for specific purposes, recently citizens are seen as sensors as well. People always record what they see and hear which would be published in social networks. Associating with spatial and temporal tags, these data sets are ideal for smart city applications such as city regional semantics recognition, smart transportation, trace analysis and so on. However, the negative part of this strategy seeing citizens as sensors are quite obvious. First of all, data collected from personal digitally social networks can be noisy, how to extract useful data is a big challenge. Secondly, this kind of data sometimes can be misleading since the initial purpose of these data are not for the urban analysis. The data literacy is a problem. The last issue is that accessing personal data always cause privacy debates which has to been taken into serious consideration.
- **Data management** One goal of smart cities is to reveal the underlying relationship between complex urban dynamic data and urban phenomenons. The massive urban dynamic data are collected by pervasively deployed digital sensors or citizens as sensors. The wide deployment of sensors is the foundation for the mission critical data-driven tasks and urban analysis. However, sensors widely deployed in smart cities are heterogeneous, especially between digital sensors and citizen as sensors, the urban data collected could be in different forms and unstructured as numerical forms and word forms. It is a big challenge to obtain the valuable data from such a big data base. What's more noteworthy, the data collected by digital sensors are always quantitative, but the data from citizens could be qualitative. For same set of quantitative data, combining with different qualitative data, they could have significantly different meanings. How to combine heterogeneous data from a layered sensor environments is non-trivial.
- **Data processing** Data processing is the kernel part for urban applications. Mining the data, building the model, extracting new patterns are critical in smart city. Although a variety of data mining algorithms have been developed, more novel algorithms are in need. Some of the currently available machine learning algorithms can only solve the problems under certain context or certain distribution. In many research areas, the models established are under certain conditions, which are not suitable for many real-time, or mission critical urban applications.

2 Urban Surveillance

Human caused disasters are one of most concerning issues in smart city, for example, car accidents caused by intentionally overspeed driving. Such kind of disasters not only lead to the loss of money, but also could be fatal to innocent people. Urban surveillance, utilizing the ubiquitous sensors and data processing technologies, is an essential part for quick detection and prompt response to urgent situations in smart city. In this section, the motivations and the existing problems of urban surveillance is discussed.

2.1 Urban Surveillance: Motivations

As the rapid urbanization and pervasively increasing of urban dynamic data, smarter management strategies are expected for administrators and planners. Efficient urban surveillance is important for situational awareness (SAW), which is essential for many critical and dynamic data-driven missions [9–11]. Considering the wide dimensions of cities, urban surveillance is becoming an indispensable part in urban planning. Along with the prosperity, the urbanization has also witnessed the increase of crimes and violations. Considering the limited law enforcement resources, the digitalized urban surveillance can provide urban residents a safer residence environment.

Cameras, smart phones, transportation cards or any other digital devices can be utilized for data collection, which enable timely tracking, analysis and decision making. Beside situational monitoring for crimes and violations, urban surveillance is also a powerful tool in some special environments that may be dangerous to human operators. For example, the chemical products storage and the surrounding environment where risk of explosion exists. In addition, trace data of individuals or communities are also very useful for epidemics dissemination control, abnormal illegal events detection and even early alarm for terrorism activity. In urban surveillance, un-interrupted dynamic data sensing, real-time massive data analysis and instant accurate decision making for sudden disasters are quite critical and significant. Understanding and analyzing the large-scale complex urban dynamic data is of great importance to enhance both human lives and urban environments.

Target detection and potential danger recognition from surveillance data is achieved through an exploitation of a layered sensor environment, and real time detection is ideal [9]. However, this mission is challenging due to the lack of powerful computing infrastructures at the surveillance site that is able to process the big dynamic data. Outsourcing all the data to remote data center may not be able to meet the tight latency constraints [12]. Therefore, a smart surveillance strategy is expected to leverage the computing power close to the job site, such that it is feasible to achieve the goal of instant decision making.

2.2 Urban Surveillance: Open Problems

Urban surveillance is a significant part for situational awareness and mission critical tasks. However, there are still some open problems in building a real-time urban surveillance system in practice. Here we illustrate two of the major issues, data sensing and computing architecture.

- **Surveillance data sensing**: Widely deployed sensors and pervasively used smart devices enable urban surveillance. Meanwhile, surveillance sensors are not enough in certain specific tasks. Let's consider the urban traffic surveillance as an example. In the United States, the traditional way to catch over speeding drivers is that the policemen patrolling on the road. Now with the help of cameras installed along the roads or at the intersections, the over speeding surveillance can be more efficient. However, due to the limited resource of police department and the number of installed cameras at fixed locations, overspeed drivers can still drive at whatever speed they like as long as they are aware of the police officer and remember to slow down near the cameras. Obviously, new sensors and detecting strategies are needed. Ideally the urban surveillance data-driven system should be capable to recognize the potential danger efficiently, to infer the reasons accurately and to make decisions quickly.
- **Computing architecture**: It is ideal that the urban surveillance system can fuse and process the dynamic urban data from a heterogeneous layered sensors in a real-time manner, especially for latency-sensitive applications. However, the existing computing architecture cannot serve this kind of applications well. At the current stage of urban surveillance, Cloud Computing is recognized as a promising solution for large scale data processing. The real-time, dynamic data are collected from sensors deployed for critical surveillance tasks. It implies that huge volume of data need to be transferred to Cloud center for pattern recognition and decision making. However, Cloud cannot satisfy the requirements of all the mission critical applications, especially those requesting tight response time. As various surveillance missions are emerging, current computing architecture failed to satisfy the requirements.

3 Fog Computing

Dynamic data fusion is highly desired for urban surveillance, especially in the context of natural or human caused disasters. Cloud Computing provides cost efficient solutions for large scale, batch data processing jobs. As mentioned above, however, the fast development of ubiquitously deployed sensors for data collection and mobile computing techniques is pushing systems to the boundary where Cloud Computing is not able to satisfy users' requirements [13]. Not only because powerful Cloud cannot meet the tight latency tolerance as being allocated far away from the sensing area, but also there is often no guaranteed connection during the emergency. Therefore, to meet the requirements of mission critical situations, it is crucial to provide the functionalities of data processing, information fusion, and decision-making in an on-site manner. Fog Computing [13–15], a recently proposed extension and complement for Cloud Computing, enables computing at the network edge without outsourcing jobs to remote Cloud centers [15]. In this section, the concepts of Fog Computing is introduced and then an architecture including both Cloud Computing and Fog Computing paradigm is given.

3.1 Fog Computing: Concepts

The Internet of Things are connecting physical infrastructure together providing us the opportunity to access the remote data, sense the situation and control the physical systems for efficient resource management and accurately customized services. Cloud Computing is a promising computing paradigm providing users hardware or software infrastructure with flexibility [16]. Furtherly, Cloud Computing free the users with no need for capital outlay and maintenance cost.

Besides the economy and flexibility, Cloud Computing is charactered as powerful computation capability and batch processing for which it is ideal for large scale urban dynamic data analysis [17]. However, the developing of Internet of Things are leading to a wide variety of innovative applications and surveillance missions. The ubiquitous smart devices and advanced networking technologies are giving rise the new location awareness services which require on-site processing and low latency quality of service. Based on the tight requirements proposed by some applications, the Cloud Computing cannot be feasible for all the tasks anymore.

Fog Computing, as the extension of and complement architecture to Cloud Computing, is a promising computing paradigm to meet the requirements proposed by rapidly developing Internet of Things. The definition of Fog Computing can be given as below:

> Fog Computing is a distributed computation paradigm that leverages the huge number of heterogeneous devices deployed at the edge of the network, which are connected with each other and collaborate with each other by sharing computation, storage and communication functionalities.

As depicted in definition, the ubiquitously deployed digital devices serve as Fog Computing nodes. Cloudlet, personal laptops, smart phones, tablets and even routers could be Fog nodes. Some of the advantages of Fog Computing are listed as below:

- **High availability**: With the prosperity of the Internet of Things, more and more digital devices can serve as Fog Computing nodes as long as they are capable of processing and storing data.
- **Location awareness**: In Fog Computing paradigm, the fog nodes are deployed close to the data source and the computing results are often used locally. The location awareness is highly desired for many smart city surveillance applications.
- **Low latency** Comparing to the remote Cloud data center, Fog Computing paradigm enables the computation directly on site, at the edge of network, it removed

the round trip time from job site to Cloud centers. This characteristic is essential to many latency sensitive applications.
- **Networking efficiency:** In Cloud Computing scenario, the data needs to be transmitted from end users to remote data center. It relies on the network conditions and actually causes unnecessary network traffic since in many cases the data are only locally significant.
- **Security and privacy**: Processing the collected data on-site can also reduce the risk of being intercepted or compromised by attackers.

3.2 Fog Computing Architecture

Figure 2 shows a Fog Computing architecture. The layer at the bottom consists of end users with arbitrary locations. There would be a wide variety of processing tasks considering the large amount of end users in the Internet of Things. The amount of real-time, dynamic user data could be tremendous and the users request based processing applications are heterogeneous. It is certain that the large scale data are difficult to be stored at the end user side because of the limited storage capability. Furthermore, some mission critical tasks are latency sensitive, the large round trip

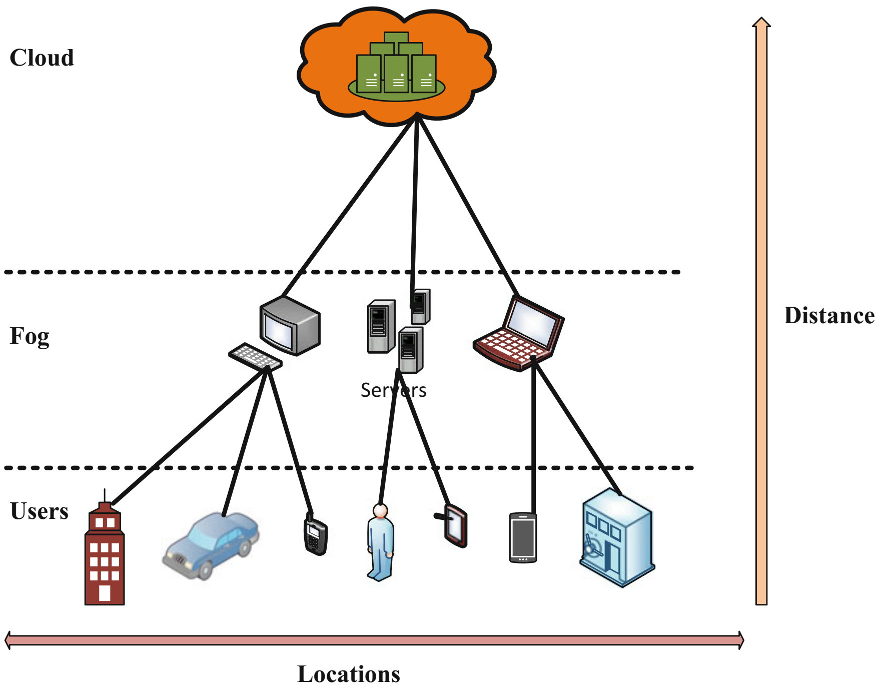


Fig. 2 The Fog Computing architecture

Table 1 Comparison between Fog and Cloud

Computing paradigm	Fog Computing	Cloud Computing
Resource allocation	Distributed	Cluster
Real-time interaction	Yes	No
Latency	Low	High
Devices	Heterogeneous	Virtual machines
Computation capability	Normal	Powerful

time between end user and Cloud center will not be tolerated. Therefore, the Fog Computing layer is allocated between traditional Cloud layer and end users layer.

The collected data will be transferred to Fog nodes for processing. Once the job is done, the Fog nodes send the results back to the user applications and upload meta data to the Cloud center for further analysis.

The top layer is still the Cloud layer which will provide powerful computing infrastructure to end users. Tremendous amount of data after being processed by Fog layer are stored in the Cloud centers and advanced data mining technologies are utilized to obtain the thorough and long-term analysis which is an indispensable part in smart cities.

A comparison between Fog Computing and Cloud Computing is given in Table 1 below.

The main difference between Fog paradigm and Cloud paradigm shown in the table actually verifies that they are complementary to each other. Fog and Cloud tackle different requirements for distinct tasks in smart city. Cloud Computing with powerful hardware and software infrastructure is more suitable for large scale data analysis for deeper insight. In contrast, Fog Computing is not as powerful as Cloud Computing. But with the property of low latency and real-time processing, Fog Computing is more desirable in situational awareness tasks of urban surveillance.

4 A Case Study

Traffic surveillance enables traffic monitoring and traffic light adjustment by utilizing the data collected from the widely deployed cameras and sensors [18–20, 20–22]. Real-time traffic data helps the police department optimize the resources to deal with accidents at particular locations in a more efficient and active way. In traffic surveillance, how to effectively monitor speeding vehicles is always a big concern. Overspeed driving not only hurts drivers, but also may cause fatal consequences to innocent people like pedestrians or people in non-overspeed vehicles. A report from Texas Department of Transportation [5] shows that in 2014, 344 people in Texas were killed in crashes involving speed-over limit of which 210 people were the drivers and the other 134 persons were passengers or pedestrians. Therefore, a smart, on-line urban speeding vehicle surveillance system would be very helpful to reduce the number of car accidents.

We took real-time traffic monitoring as a case study to evaluate the feasibility of a Fog Computing based urban surveillance solution [10, 23]. In our scheme, a drone in the sky monitors vehicles on the ground and a raw video stream is collected and sent back to the controller on ground, which implements the real-time vehicle surveillance. And the raw video stream is sent to Fog Computing nodes nearby, such as a laptop, like the one in police cars. The Fog node tracks the moving target in the video and calculates the speed. Our scheme not only can monitor the traffic in much wider area compared to cameras on ground, but also process the video data and get the speed outcome in real time, which is critical for immediate response to the speeding violation.

This section is organized as follows. First we introduce the Fog Computing based urban surveillance system for speeding detection. Then the target tracking and speed calculation algorithms are presented. Finally, the experimental results are reported with some discussions.

4.1 System Architecture

Figure 3 illustrates the three-layer system architecture of the Fog Computing based urban surveillance system. It consists of the remote Cloud center, on-site/near-site computing Fog, and data collection units, such as sensors and cameras. The kernel is the Fog layer, which is formed by multiple computing units, including drones, computers carried by the vehicles, and computing devices of first responders. They monitor the area concurrently from different positions. When they are collecting real-time data streams, each of them also needs processed information for instant decision making. Although it is ideal that all the collected data are sent back to central Cloud facility for thorough global analysis, there is no guarantee that a reliable communication network to remote Cloud center is always available. In addition, not all data is globally significant. It does not need to create unnecessary traffic in the networks. Instant on-site decision making also reduces the risk of exposing the data to eavesdroppers in transmission channels. Therefore, the Fog, which consists of the computing devices carried by the units in or near the monitoring area, can fulfill the requirements very well.

In the existing traffic monitoring systems, police officers and cameras along the road are not sufficient to effectively keep the drivers from driving above the speed limit, especially because of the limited monitoring range of police cars and the number of cameras. In the proposed Fog based surveillance system, drones flying in the sky generate real-time raw video stream, and send it back to the ground station and display it on the screen. Thus the human operator can monitor the traffic of a much wider range, detect fast moving vehicles on the screen, and get the speed information when they find someone is suspicious.

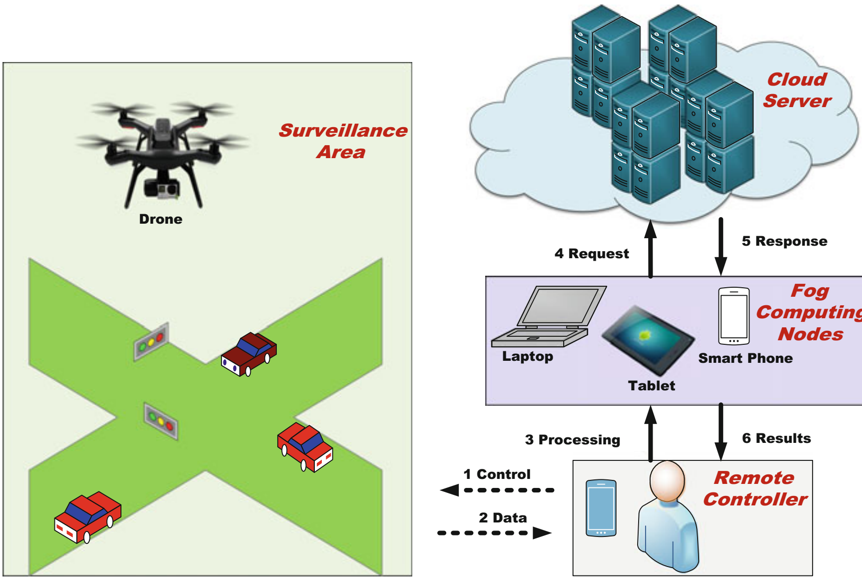


Fig. 3 System architecture

It is worth to note that the nodes in Fog layer provide both computation power and storage space. The raw video data can be pre-processed and stored in the Fog nodes at first and be delivered to the remote Cloud center for a long-term analysis to build a historical record of the traffic condition. For example, with the collection of traffic events in a long period, a big picture may be constructed, which is very valuable for city planners and policy makers.

4.2 Algorithms

In order to catch speeding vehicles, two algorithms are needed: one is the target tracking algorithm and the other is the vehicle speed calculation algorithm.

A. Target Tracking Algorithm

When a fast moving vehicle appears in the real-time video, it needs to be locked immediately and be tracked effectively frame by frame. In practical scenarios, not only the suspicious vehicle itself appears in the video frame, but also occlusion, background clutter, the change of illumination and the noise. It becomes more challenging if the user needs to conduct the job in the night, when the road is dark. Other than the requirements of robustness and effectiveness of the tracking algorithm, the processing time in the Fog nodes could be another big concern. When a user locks one moving vehicle in the video, the speed information of that vehicle is needed instantly.

Considering the specific requirements, a real time robust L1 tracker using accelerated proximal gradient approach [24] is selected. This accelerated real time L1 tracker is casted by using the sparse representation in the particle filter framework.

The particle filter implements the recursive Bayes estimation using the method of non-parametric Monte Carlo simulation. It uses a large number of particles that are transferred in the state space to estimate the probability density function of state variables. Particle filter is an efficient tool to solve the problem in non-linear system. In addition, the distribution of random variables are unnecessary to be Gaussian distribution. Two steps are essentially involved in the particle filter: prediction and update.

We denote x_t to represent the state variable describing the motion of the target in frame t. y_t denotes the observation of the moving target in frame t. In target tracking applications, we assume state variable x_t is only related to x_{t-1} and the observation at frame t is only related to x_t, which means observations among $y_{1:t} = \{y_1, y_2, \cdots, y_t\}$ are independent of each other, given $x_{1:t}$. It is assumed that at frame $t-1$, the probability density distribution is $p(x_{t-1}|y_{t-1})$. In prediction step, $p(x_t|y_{t-1})$ is derived from $p(x_{t-1}|y_{t-1})$:

$$p(x_t, x_{t-1}|y_{t-1}) = p(x_t|x_{t-1}, y_{t-1})p(x_{t-1}|y_{t-1}) \tag{1}$$

In Eq. (1), given x_{t-1}, x_t and y_{t-1} are independent, so Eq. (1) becomes:

$$p(x_t, x_{t-1}|y_{t-1}) = p(x_t|x_{t-1})p(x_{t-1}|y_{t-1}) \tag{2}$$

Then compute the integration of Eq. (2) over x_{t-1}:

$$p(x_t|y_{t_1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{t-1})dx_{t-1} \tag{3}$$

With Eq. (3), we can move forward to the update step by using Bayes rules:

$$p(x_t|z_t) = \frac{p(x_t|x_{t-1})p(x_t|y_{t-1})}{p(y_t|y_{t-1})} \tag{4}$$

$p(y_t|x_t)$ is the observation likelihood. In the particle filter, the posterior probability above is estimated by N samples, denoted by $S_t = \{x_t^1, x_t^2, x_t^3, \cdots, x_t^N\}$ with different weights. Due to the lack of knowledge about variable distribution, sequential important distribution $q(x_t^{(i)}|y_t)$ is used to generate the samples. The weight is:

$$W_t^{(i)} \propto \frac{p(x_t^{(i)}|y_t)}{q(x_t^{(i)}|y_t)} \tag{5}$$

and the weight can be updated as follows:

$$W_t^i = w_{t-1}^i \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t|x_{1:t-1}, y_{1:t})} \tag{6}$$

The observation likelihood depicts the similarity between the target candidate and the target templates [25].

In sparse approximation, the signal y can be linearly represented by the atoms of the over-complete dictionary D.

$$y = D \cdot x \tag{7}$$

where x is the coefficient of each atom in the dictionary D. In moving target tracking algorithm, over-complete dictionary consists of target templates denoted by $T = t_1, t_2, t_3, \cdots, t_n$. With the target templates, a target candidate can be represented as follows:

$$y \approx T \cdot x = x_1 t_1 + x_2 t_2 + \cdots + x_n t_n \tag{8}$$

Because of the sparsity in sparse approximation, for a good target candidate, most coefficients of the target templates should be zero, which means a good target candidate can be nearly represented by several target templates. In other words, the coefficients of a bad target candidate can be relatively equally with smaller number.

In the real scenarios of our monitoring videos, other than the target object, occlusion, noise, shadows, sometimes even darkness would appear. So we have to consider the error. Therefore, trivial templates denoted by $I = i_1, i_2, i_3, \cdots, i_n$ are introduced in this algorithm and the Eq. (8) is rewritten as follow:

$$y = \begin{bmatrix} T \; I \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} \tag{9}$$

where e represents the coefficients of trivial templates. In a further consideration, it is reasonable to assume that the coefficients of a good candidate should be positive, which can also be considered as the non-negative constraints. Hence, in this scenario, positive and negative trivial templates should be involved. Then Eq. (9) is rewritten as:

$$y = \begin{bmatrix} T \; I \; -I \end{bmatrix} \begin{bmatrix} x \\ e_+ \\ e_- \end{bmatrix} = D \cdot m \tag{10}$$

Here, $D = (T \; I \; -I)$ and $m^T = (x \; e_+ \; e_-)$. What we want to know is the coefficients m of the target templates and trivial templates, but in the over-complete dictionary $D^{m \times n}$, m is much smaller than n, which means the solution of Eq. 10 is not unique. Some constraints are indispensable in order to get a unique solution in the sparse representation. Fortunately, we can solve this problem as an $L1$ norm least squares problem.

$$min\|Dm - y\|_2^2 + \lambda\|m\|_1 \tag{11}$$

$\|\cdot\|_1$ denotes l_1 norm and $\|\cdot\|_2$ denotes l_2 norm here respectively. As mentioned above, trivial templates are brought into the dictionary to deal with the noise and occlusion. But what if there is no occlusion? The target object should be well approximated by the target templates from previous frames. Additionally, in case of no occlusion in the frame, the trivial templates would impact the detection accuracy otherwise and bring some computation complexity. So in this accelerated $l_!$ norm tracking algorithm, another coefficient μ_t is introduced to improve the constraint (11). The revised constraint is as below:

$$min \frac{1}{2}\|y_t - Dm\|_2^2 + \lambda\|m\|_1 + \frac{\mu_t}{2}\|m_I\|_2^2 \tag{12}$$

where m_I is the coefficients of trivial templates in this target tracking sparse approximation problem: $m = [m_T\ m_I]$. If occlusion is detected in a video frame, μ_t is zero. Otherwise, μ_t is supposed to be some specific value.

In practical experiments, solving such kind of modified l_1 norm minimization could be pretty time consuming. A fast numerical method called accelerated proximal gradient [26] is applied to solve this problem. This approach is designed for solving the optimization problem as below:

$$min\ F(a) + G(a) \tag{13}$$

and the accented proximal gradient is fast for some specific types of function G.

After solving the l_1 least squares minimization problem and obtaining the sparse coefficients m, the observation likelihood of state variable x_t^i can be expressed as:

$$p(y_t|x_t^i) = \frac{1}{\Gamma} exp\{-\alpha\|y_t^i - T_t m_T^i\|_2^2\} \tag{14}$$

where α is used to control the shape of a Gaussian Distribution and Γ is a normalized factor. m_T denoted the coefficients of target templates. The optimal state x_t^i satisfies:

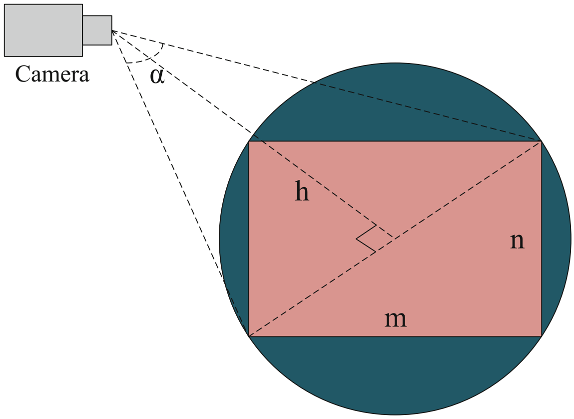
$$x_t^i = \arg\max_{x_t^i \in S_t}\ p(y_t|x_t^i) \tag{15}$$

B. Speed Calculation Algorithm

Leveraging the moving target tracking algorithm discussed above, the position of a vehicle in consecutive frames can be identified using a bounding box to lock the moving vehicle on the road of interests. If the real distance that the vehicle has traveled is available, the speed can be calculated easily.

As shown in Fig. 4, the pink rectangle represents one video frame whose resolution is $m \times n$, and the dark blue circle represents actual field taken by the drone camera. α is the field of view of the drone camera and the dash line h is the height from the camera to the ground. Knowing height h and the field of view α, the longest distance of the field monitored by the drone camera, which is the diameter of the dark blue circle as well, can be:

Fig. 4 Field of view



$$d = 2h \times tan\frac{\alpha}{2} \tag{16}$$

According to the way a camera works, the image plane is also a circle that is circumscribed of the CCD (Charge Coupled Device) plate in the camera. We assume that each pixel represents nearly equally length, so the actual length that one pixel represents in an image can be obtained by:

$$l = \frac{d}{\sqrt{m^2 + n^2}} \tag{17}$$

The l_1 tracker algorithm indicates that the position of the moving vehicle frame by frame. Assume in two consecutive video frames, the vehicle position changes from (x_1, y_1) to (x_2, y_2). The pixel number that the vehicle moves across in the frame is:

$$n = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{18}$$

Then, $R = l \times n$ is the actual range that the vehicle moves between two consecutive video frames and combined with the time interval, we can calculate the speed v:

$$v = \frac{R}{t} \tag{19}$$

4.3 Experimental Results

We have evaluated the proposed Fog Computing based speeding vehicle detection scheme. The experimental results are reported in this section.

A prototype has been built to validate the feasibility of Fog Computing based smart urban surveillance. Two DJI Phantom 3 Professional drones are used to monitor the traffic on road. Two Google Nexus 9 tablets are connected to the remote controller of the drones such that the collected video streams are played to the operators in real-time. One laptop functions as a Fog Computing node, which configuration is Intel core i7-2720QM 2.20 GHz and the RAM memory is 8.00 GB. The given FOV (Field of View) of the camera on the drone is 94°. However, in real camera engineering, manufacturers would always make the image plane not perfectly circumscribed with the CCD plate. Therefore, the diameter would be a little larger than the actual length represented by the diagonal of the image. We have taken a photo of an object whose dimensions are known to calibrate FOV. The calibrated FOV is 89.39°.

It is a challenge to track an object in a noisy environment, such as the shadows on the road, multiple similar targets on the road or even dark video frames because of the night. It is critical to ensure that we will not lose the track of the vehicle of interest.

Figure 5 shows the results of target tracking in day time, with tree occlusions on the road from the roadside. The test video stream was taken above a road on Binghamton University campus. The vertical height from the camera on our drone to the ground is 140.0 m. The black car is the target and a white bounding box is on its body in the image for locking. The tracking algorithm has successfully tracked that black car all the time and never lost it.

Then some more challenging scenarios were considered. Figure 6 shows the tracking results with multiple vehicles moving on the road. This video stream was taken above a local freeway and the vertical height is 262.5 m. The results verified that the proposed Fog based surveillance scheme is able to track the target coexisting with multiple similar objects.

Fig. 5 Tracking test sequence 1

Fig. 6 Tracking results with multiply similar objects

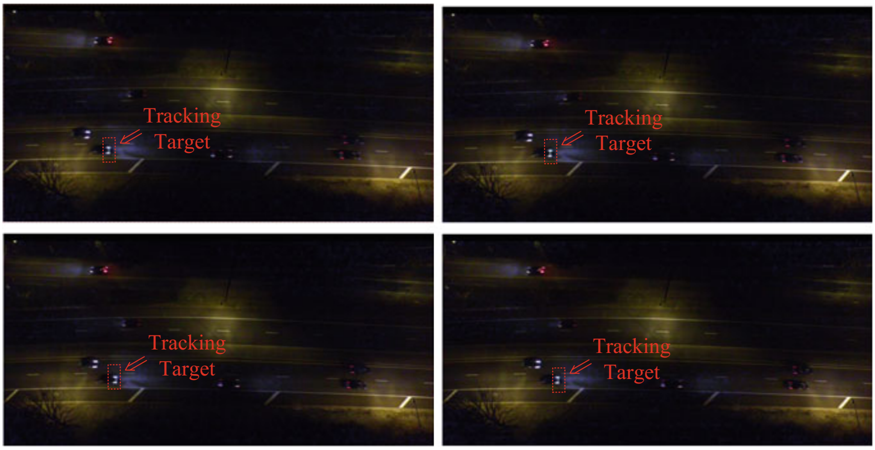


Fig. 7 Tracking results in dark night

Also, the capability of tracking in dark is highly desired by city administration since there is less police force on road in evening times and drivers are more likely to drive faster. At night it is too dark to distinguish the body of cars from the background. Our strategy is to lock the headlights of a moving vehicle. The results shown in Fig. 7 verified the effectiveness of the tracking scheme in dark night.

Fig. 8 Speed calculation test scenario

5 Speeding Vehicle Detection

The first experiment examined the accuracy of speed evaluation. The video is taken above a campus road of our university, and the vertical height of the drone is 140.0 m. As shown in Fig. 8, a black Toyota Camry is moving on the road with the constant speed of 27 mile per hour. There are 160 frames in the video stream processed with the time interval of 100 *ms* between two consecutive frames. The speed of tested vehicle was calculated every ten frames, and the speed estimation error is defined as follows:

$$error = \frac{|estimation - actual|}{actual} \tag{20}$$

Figure 9 presents the estimated driving speed of the vehicle. It is close to the actual driving speed, 27 mile per hour, varying from 26.2358 to 28.7736. The estimation errors are shown in Fig. 10, which is small (lower than 10 %). This experiment has proved that the proposed scheme can efficiently estimate the speed of vehicles driving on road with an acceptable accuracy.

Then, we have applied our system to monitor real traffic on highway and caught speeding vehicles successfully. The video is taken on the highway I81-N, where the speed limit is 65 mile per hour. Figure 11 shows that a red freight truck at the left lane is speeding up to pass a white freight truck in the right lane.

The speed estimation results are depicted in Fig. 12. The red line represents the speed of the vehicle on the left lane and the blue line represents the slower one on the right lane. The red line is above the blue line almost all the time, which is consistent with the observation in the video. The ranges of the speed over the limit of these two vehicles are depicted in Fig. 13. The faster vehicle is speeding and its speed was

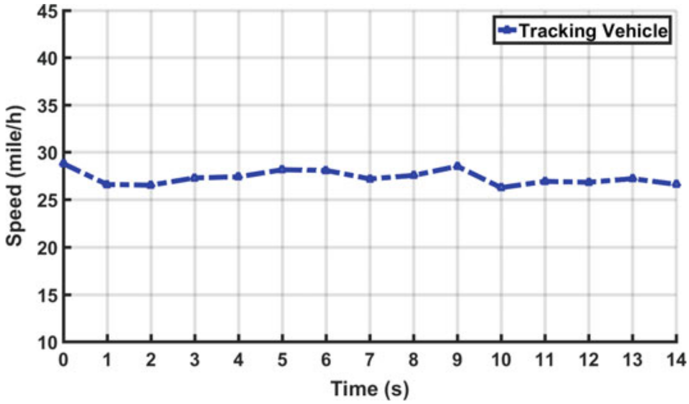


Fig. 9 Speed calculation results

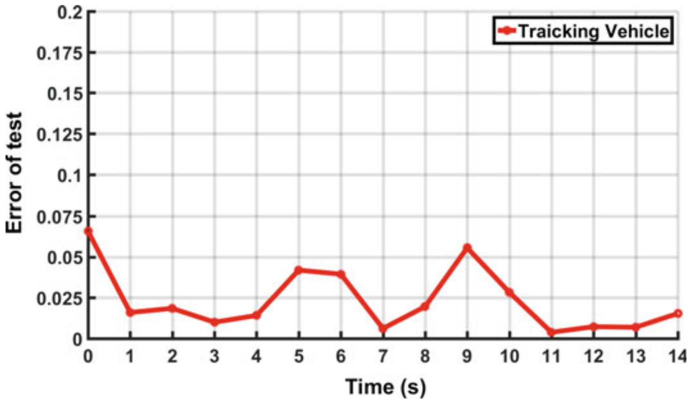


Fig. 10 Speed estimation errors

Fig. 11 Test sequence on highway

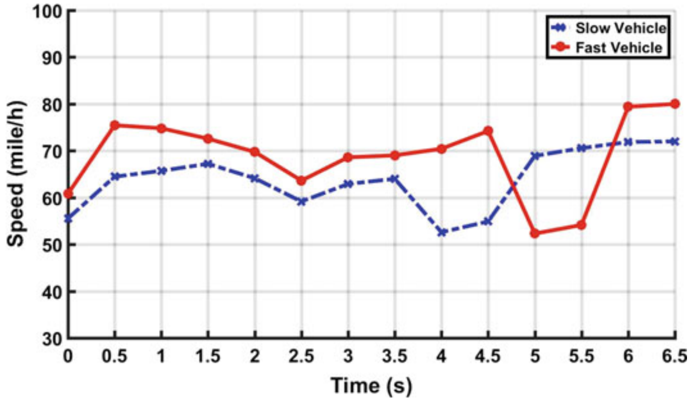


Fig. 12 Highway speed calculation

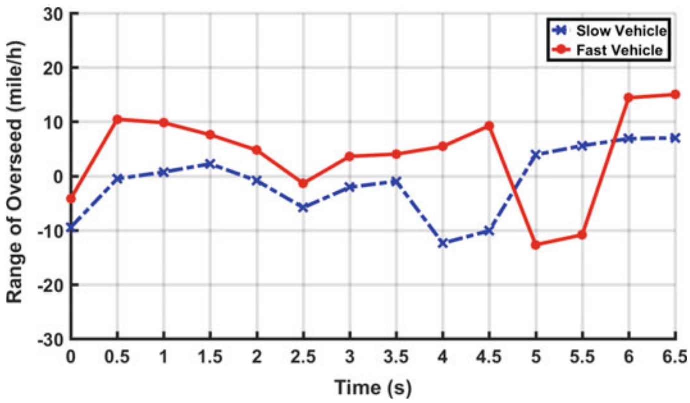


Fig. 13 Range of overspeed

10 miles higher than the limit. In particular, after 6 s, it became 15 miles faster than the speed limit. In contrast, the blue line shows that the speed of the slower vehicle keeps around the speed limit pretty well.

6 Summary

The emergence of Internet of Things and advanced computing architecture are motivating us to implement the concept of smart city. The focus of smart city is to make citizens happier, management more efficient and business more prosperous by utilizing innovative information and communication technologies. However, implementing a smart city still faces some challenges. The information fusion from heterogeneous layered sensors, efficiently and correctively analysis of large scale dynamic data are difficult. Situational awareness, which is an essential part of urban surveillance, requires short processing delay and quick decision making which would be hard for existing methods.

Cloud Computing is recognized as a promising solution for smart city. But it cannot meet the requirements from mission critical applications. The latency is the main obstacle to implement Cloud Computing based urban surveillance. Fog Computing can enable computation directly at the edge of network which is close to end users providing storage and processing schemes. Such that Fog Computing is potentially the ideal tool to be utilized in situational awareness tasks.

In this chapter, we proposed to apply Fog Computing in smart urban surveillance and validated the feasibility using speeding vehicle detection as a case study. Leveraging the Fog Computing paradigm, a conceptual proof prototype has been built, in which DJI Phantom 3 Professional drones are used for real-time surveillance video collection, tablets and laptops serve as Fog Computing nodes. It allows users monitor the traffic and track the speeding vehicles effectively. Field experimental studies have been conducted and the results show that the proposed scheme can efficiently track the target and catch speeding vehicles in real-time. This work has validated the feasibility of applying the Fog Computing paradigm to make the city smarter.

References

1. Arribas-Bel, D.: Accidental, open and everywhere: emerging data sources for the understanding of cities. Appl. Geogr. **49**, 45–53 (2014)
2. Batty, M.: Smart cities, big data. Environ. Plann. Part B **39**(2), 191 (2012)
3. Zheng, Y., Capra, L., Wolfson, O., Yang, H.: Urban computing: concepts, methodologies, and applications. ACM Trans. Intell. Syst. Technol. (TIST) **5**(3), 38 (2014)
4. U. Nations, World Urbanization Prospects 2014: Highlights. United Nations Publications (2014)
5. T. D. of Transportation, Texas motor vehicle crash statistics (2014). http://www.txdot.gov/government/enforcement/annual-summary.html. Accessed 01 Nov 2015
6. Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M.: Internet of things for smart cities. IEEE Internet of Things J. **1**(1), 22–32 (2014)
7. Chen, M., Mao, S., Liu, Y.: Big data: a survey. Mob. Netw. Appl. **19**(2), 171–209 (2014)
8. Yin, C., Xiong, Z., Chen, H., Wang, J., Cooper, D., David, B.: A literature survey on smart cities. Sci. China Inf. Sci. **58**(10), 1–18 (2015)
9. Blasch, E., Seetharaman, G., Suddarth, S., Palaniappan, K., Chen, G., Ling, H., Basharat, A.: Summary of methods in wide-area motion imagery (wami). In: SPIE Defense + Security. International Society for Optics and Photonics, pp. 90 890C (2014)
10. Chen, Y., Blasch, E., Chen, N., Deng, A., Ling, H., Chen, G.: Real-time wami streaming target tracking in fog. In: the 2016 SPIE Defense, Security, and Sensing (DSS) (2016)
11. Wu, R., Chen, Y., Blasch, E., Liu, B., Chen, G., Shen, D.: A container-based elastic cloud architecture for real-time full-motion video (fmv) target tracking. In: Applied Imagery Pattern Recognition Workshop (AIPR), 2014 IEEE, pp. 1–8. IEEE (2014)
12. Wu, R., Liu, B., Chen, Y., Blasch, E., Ling, H., Chen, G.: Pseudo-real-time wide area motion imagery (wami) processing for dynamic feature detection. In: 2015 18th International Conference on Information Fusion (Fusion), pp. 1962–1969. IEEE (2015)

13. Bonomi, F., Milito, R., Natarajan, P., Zhu, J.: Fog computing: a platform for internet of things and analytics. In: Big Data and Internet of Things: A Roadmap for Smart Environments, pp. 169–186. Springer (2014)
14. Stojmenovic, I., Wen, S.: The fog computing paradigm: scenarios and security issues. In: 2014 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1–8. IEEE (2014)
15. Yi, S., Li, C., Li, Q.: A survey of fog computing: Concepts, applications and issues (2015)
16. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
17. Stantchev, V., Barnawi, A., Ghulam, S., Schubert, J., Tamm, G.: Smart items, fog and cloud computing as enablers of servitization in healthcare. Sensors Transducers (1726-5479) **185**(2) (2015)
18. Buch, N., Velastin, S., Orwell, J., et al.: A review of computer vision techniques for the analysis of urban traffic. IEEE Trans. Intell. Transp. Syst. **12**(3), 920–939 (2011)
19. Kitchin, R.: The real-time city? big data and smart urbanism. GeoJournal **79**(1), 1–14 (2014)
20. Megalingam, R.K., Mohan, V., Leons, P., Shooja, R., Ajay, M.: Smart traffic controller using wireless sensor network for dynamic traffic routing and over speed detection. In: Global Humanitarian Technology Conference (GHTC), 2011 IEEE, pp. 528–533. IEEE (2011)
21. Sarowar, S.S., Shende, S.M.: Overspeed vehicular monitoring and control by using zigbee
22. Srinivasan, S., Latchman, H., Shea, J., Wong, T., McNair, J.: Airborne traffic surveillance systems: video surveillance of highway traffic. In: Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks, pp. 131–135. ACM (2004)
23. N. Chen, Y. Chen, Y. You, H. Ling, and R. Zimmermann, "Dynamic urban surveillance video stream processing using fog computing," in *the 2nd IEEE International Conference on Multimedia Big Data (BigMM 2016)*
24. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1830–1837. IEEE (2012)
25. Mei, X., Ling, H.: Robust visual tracking using l_1 minimization. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1436–1443. IEEE (2009)
26. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. SIAM J. Optim. (2008)