Smartphone-Oriented Development of Video Data Based Services

Nikita A. Bazhenov, Dmitry G. Korzun, Sergey I. Balandin
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{bazhenov, dkorzun}@cs.karelia.ru
sergey.balandin@fruct.org

***Abstract*—The massive introduction of video capturing devices in Internet of Things (IoT) environments leads to development of various video data based services. In this paper, we consider the need and background on the video data based services in IoT environments. Based on the smart spaces approach, we introduce the architecture and distributed configurations to construct such services using primarily local devices and to deliver such services using smartphones. We discuss possible data models that can be used on such mediatory components as a local video server and a semantic information broker.**

I. Introduction

The mobile and Internet of Things (IoT) technologies enable development of advanced digital services, e.g., see [1], [2]. The IoT supports deliver of such services as personalized information assistance and online recommendations in our daily life [3], [4]. Mobile services provide the end-user with ubiquitous access to the Big Data sensed in current IoT environment and further processed and accumulated as valued information resources in the global Internet [5], [6]. Video capture data form a valuable class of IoT sensed data.

This paper considers the development problem of video data based services, where the video data are captured in local IoT environment. We expect that an essential part of data can be processed locally and nearby using appropriate devices. Our approach is supported with the two emerging IoT-enabled paradigms: Edge-centric Computing [7] and Fog Computing [8]. In particular, such services are in demand for mobile health systems used in everyday patient life for health monitoring, prevention, and assistance [9], [10].

The basic processing capacity for local construction of video data based services is provided by everyday personal mobile devices (smartphone, tablet). Modern smartphones are relatively powerful and IoT-friendly, making them reasonable devices for data processing and service construction [22]. Many smartphones are based on open platforms (e.g., Sailfish OS or Tizen OS), which supports easier programming. We argue in this paper that existing desktop and mobile technologies for programming surrounding IoT devices allow creating a powerful distributed system for processing video data in the local IoT environment.

In our previous work [11], we studied the basic development opportunities of connecting and interacting video cameras with other IoT devices. In particular, we analyze development of a video surveillance system, where cameras provide video data for further processing on user's smartphone. In this paper, we continue our study and propose a distributed architecture deployable in IoT environments with presence of many personal mobile devices. We introduce possible device configurations in respect to what devices can be involved into the distributed system. The proposed configurations show how to advance the system with more device classes to take data processing tasks.

The rest of the paper is organized as follows. Section II introduces preliminaries on the video data based services and technologies for IoT environments. Section III describes the architecture and possible device configurations to deploy the service-oriented system in IoT environment. Section IV considers the enabler desktop and mobile technologies that can be used for video data based services. Section V presents our data models for video data processing in the proposed architecture. Finally, Section VI concludes the paper.

II. Preliminaries

Now we observe the massive emergency of many video capturing devices [11]. Smartphones and tablets have digital cameras. Urban and rural areas have cameras embedded statically with a rotary function. Online daily use cameras appear at many homes and public areas.

Various digital services can be constructed based on the video capturing. The most topical case relies on online processing, when Internet is used to transfer the video data between the capture point and the service clients. The simplest case is when a pure video recorder is used; it transfers the recorded video further to processing and consuming point (e.g., for persistent storage at a remote server or for live watching by clients). More advanced case is when so-called "smart" devices are employed [12]. They use recorded video as a form of sensed data. The device performs own data processing to understand the current situation, to extract valued information, and to provide the result (as a service) to clients.

In addition to video capturing devices, some additional (mediatory) computers can be introduced to perform data processing. Video server is a computer that in some way processes and/or stores information from a video camera (storing records and pictures, processing and recognizing images). In some cases, the video server can be located on the same device that is recording. For instance, smartphone seems a promising devices for this purpose. This approach follows the vision of fog computing [8] and edge-centric computing [7],

Fig. 1. Application problem domains for video data based services

when appropriate local and mediatory devices are involved to perform data processing near the data sensing place.

In an IoT environment, video capturing devices generate many video flows. With every flow certain semantic information can be associated to describe the activity on the data (e.g., who and how made video data processing) and the obtained results (e.g., interpretations and conclusions from video data). Interrelation of flows as well as relation with other context information provide semantics that can be used in services. According to the smart spaces approach [1], this task is delegated to semantic information broker (SIB), which creates a shared view on available resources of the IoT environment. SIB runs on some available and relatively powerful device. The SIB task is to make linking for dynamically appearing information fragments (e.g., links to the corresponding videos, their description, and interpretations). This information is represented using the model of a semantic network [5]. Video data processing tasks are performed by agents running on some devices. The agents make semantics extraction and publish the obtained information in the semantic network via the SIB.

Video data processing is based on the automated acquisition of various image data sequences from video capture devices. Many algorithms exist for this type of processing, e.g., see the HOG method (Histogram of Directed Gradients), which is widely accepted for object recognition [13], [14]. Video surveillance systems implement processing of live data streams from video cameras to reason about the real environment [15], [16]. Topical problem domains for considered video data based services are depicted in Fig. 1.

- Smart safety and security [15], [17]: geo-spatial territory monitoring for protection. Video security is necessary at the world level. Many crimes, thefts and disasters are noticed thanks to video capturing. Moreover any security system must have “remote eyes” based on at least one video camera.
- Ambient Assisted Living [18], [16]: smart assistance to patients in their everyday life, including social and health care aspects. Elderly people need assistance because of the reduction in motor and cognitive functions. Video data processing supports monitoring the state of these human functions.
- Smart homes [19], [4]: digital assistance to people in their everyday processes at home (a kind of smartification of our houses and other buildings). The popularity of smart homes is increasing; they aim at improving the quality of people’s lives and saving time for many “everyday” activities. Also, home safety in such systems is of high priority. Video data processing supports monitoring the state and context of those in-home activities.
- Event detection [20], [21]: monitoring of production equipment and service personnel during the operation. Video data processing identifies events and analyze the current situation in the context of previous actions. Video based event detection can be used in other problem domains, e.g., for personalized assistance of the participants during their collaborative work [3].

The progress in IoT technologies has led to such paradigms as edge-centric computing and fog computing [6], [8], [7], as evolution of cloud computing. The following examples from the recent literature show the applicability of these paradigms for creating video data based services in emerging IoT environments.

The ePrivateEye platform [13] is an edge-enabled version of PrivateEye. The original PrivateEye defines a generic two-dimensional special shape that is easy for users to draw, e.g., on a piece of paper, on a whiteboard, or within a projected presentation. Marker-recognition is performed locally on a recording mobile device in real-time. The idea of ePrivateEye is offloading the marker detection to an edge server. The video data processing delegation, despite small losses in the response (delay), leads to more effective computation than the mobile device can perform, e.g., analyze 50% more frames.

The LAVEA platform [14] considers the tradeoff problem between network core (where powerful computation resources are available) and the edge (where most of the data is produced). Data processing at the edge reduces response time, improves bandwidth usage, and increases energy efficiency. LAVEA offloads computation between clients and edge nodes, collaborates nearby edge nodes. As a result, low-latency video analytics is provided at places closer to the users.

III. Smartphone-Oriented Video Data Based Service System Design

Consider the proposed architecture and system design solutions for deploying video data based services in IoT environment, following the smart spaces approach [1]. The basic architecture of the system is derived from our previous work [11], see Fig. 2. The system supports participation (in the local wireless network) of many video capture devices (cameras, either embedded or mobile) and mobile clients (running on smartphones, tablets, laptops, etc.). Each participating device acts as a knowledge processors (KP), representing computational resources of the edge. SIB is used for collecting the semantics and its sharing among the participating KPs.

A. IoT Environment Configurations

The proposed architecture supports the following functions needed in construction of video data based services: (i) video data capture using multiple devices in the IoT environment, (ii) local data processing using edge IoT devices, and (iii) semantic data mining in available information. The studied

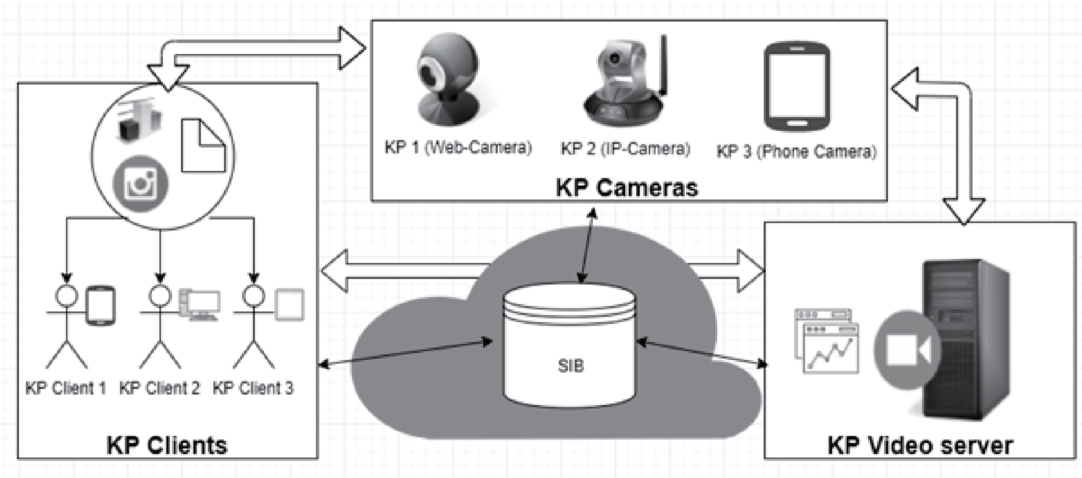

Fig. 2. Distributed architecture for using semantics in video data: the semantics are discovered by KPs, collected at SIB, and applied in service construction

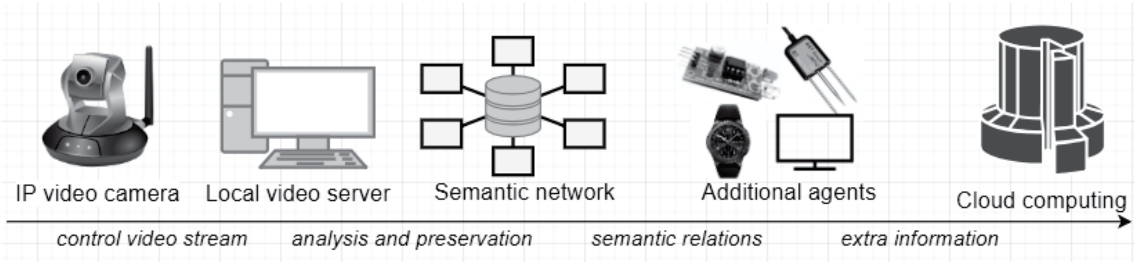

Fig. 3. Extensibility vision on IoT environment configurations in respect to involved resources in construction of video data based services

configurations of IoT environment are shown in Fig. 3. All the configurations assume that user's smartphone (or similar personal mobile device) is the primary access point (client) for the user to consume constructed services [22].

The basic configuration corresponds to "smartphone–camera" interaction in service construction and delivery. This configuration can be extended with more resources as follows.

- Local video server is employed for video data processing and storage.
- SIB for collecting results of video data processing (e.g., results from the video server) and other context into the semantic network.
- Additional KP agents for video data processing and data mining in addition to the video server.
- High-performance computing resources for advanced data processing (e.g., made on such remote computing facilities as cloud systems and data centers).

Video capture is oriented to "everyday" cameras, which now are widely distributed in human surroundings. Examples include webcams on laptop and desktop computers, static installed cameras in public areas, rotary IP cameras for in-door environments. Such a camera is connected to the Internet using local wireless network (typical case is Wi-Fi), either directly (the device has own IP address) or through the host computer. To construct services in the smartphone–camera configuration, the smartphone needs to connect to one or more cameras in the local network, then to receive and process the video streams.

When the smartphone computing capacity is not enough the configuration is extended with local video server, i.e., an additional relatively powerful edge device is involved into service construction. For example, such a video server stores information about most important pictures and video records (made by the camera) or calculates the usage statistics (connection history to the camera). In this configuration, the video server performs the essential part of video data processing, and clients (on smartphones) consume the achieved results of data mining in the processed video streams. Each KP client may compose results related to several individual streams or to distant time periods by own resources, i.e., making certain contribution to the service construction.

The above type of composition can be delegated to an additional intermediary when the configuration is extended with SIB. The latter is typically installed on a local computer, i.e., similar to the local video server. An additional layer is provided on top of the processed video streams, which is forms a shared view on available resources [3]. A simple example of this "semantic role" is that SIB keeps the access information about online cameras and their properties. Any client uses this rendezvous mechanism to access the appropriate camera. Otherwise, the client has to independently solve the complicated camera discovery task in the local network.

Additional KP agents act on behalf of various smart devices, including sensors, boards, smartphones of other users.

TABLE I. PERFORMANCE REQUIREMENTS

Problem domain	Smartphone	Data processing	Semantics
At-Home Lab	low	medium	medium
Face recognition	medium	high	high
Equipment monitoring	low	medium	high

low, medium, high: the level of required computations

The role is in augmentation of captured video data with available context information and in provision of auxiliary computational resources to data processing. In particular, such agents can relate recognized objects in video streams with the temperature in the location of those objects or with other information on the observed situation.

When complicated data mining is needed then the function of video server and SIB can be (partially) delegated to the remote facilities. They implement known big-data oriented algorithms Important results may return to the local environment and be shared using the in the local video server and/or SIB.

B. Capabilities for Service Construction

Consider the capabilities the introduced IoT environment configurations provide in respect to particular problem domains. The computing resources are used for video data processing and mining, including semantics discovery and analysis algorithms. Table I shows the performance requirements for deploying video data based services of the selected problem domains.

- At-Home Lab [9], [11]: AAL-enabled personalized healthcare services provided at home settings, where everyday cameras are used for video capture. The services analyze the human motor activity in "natural" everyday life conditions. The cost is the reduced analysis result precision compared with medical and healthcare laboratories having professional equipment.
- Face recognition [23], [24]: A person can be identified and verified based on her/his digital image in video snapshots or streams. The video data are coming from multiple cameras installed in a spatial area (building, out-door, public space). The cameras are typically of heterogeneous characteristics.
- Equipment monitoring [25], [20]: Detection of deviations in equipment operation. In addition to cameras installed near the equipments, the personnel may use smartphone camera to observe the current operation situation.

The Sailfish OS currently provides an efficient and fast mobile OS IoT-friendly ecosystem. The system is based on Linux kernel and includes most of required basic primitives for accessing low-level functions and interfaces. The open architecture allows developing and integrating the missing primitives to the system core. As a result, the Sailfish OS based smartphones are the best candidates for use in the studied IoT environment configurations [22]. Another property of Sailfish OS is its priority to the privacy and usability. This priority is a valuable bonus feature that the mobile OS can deliver to video data based services.

TABLE II. DESKTOP PLATFORMS

	Windows 7	Windows 10	Windows 8.1	MAC OS	Linux
Market %	**42.39%**	34.29%	5.56%	4.46%	1.31%
Release year	2009	2015	2013	2001	**1991**
Accessibility	**Very high**	High	High	Low	Medium
Access to low-level functions	Medium	Medium	Medium	Low	**High**
Complexity of programming	Medium	Medium	Medium	Medium	**Easy**
Processing speed + complexity	**High**	Low	Medium	Low	High
Support for various libraries	High	**High**	High	Low	Medium
Prevalence	Medium	**Very high**	High	Medium	Low

TABLE III. MOBILE PLATFORMS

	Android (4.0-6.0)	Android (7.0+)	iOS	Windows Phone	Sailfish OS
Market %	**51.54%**	34.36%	14%	0.1%	0.1%
Release year	2011	**2016**	2007	2012	2013
Accessibility	**Very high**	High	High	Low	High
Access to low-level functions	High	High	Low	Medium	**High**
Complexity of programming	Medium	Medium	**Low**	Medium	Low
Processing speed + complexity	Medium	High	High	Low	**Very High**
Support for various libraries	High	**High**	Medium	Low	High
Prevalence	High	**Very high**	High	Very low	Very high

IV. CLIENT PLATFORMS

Let us analyze mobile and desktop technologies available now on the market. Our analysis provides insights on selecting platforms for developing clients of the video data based services.

A. Technology Parameters

Consider the following parameters that influence the development of service clients.

1) Market share (%)
2) Release year
3) Accessibility for end-users
4) Access to low-level functions
5) Complexity of programming
6) Processing speed and computation complexity
7) Support for various libraries
8) Prevalence

We summarized the above parameters in Tables II and III the analytical reviews from the following Internet resources.

- COMSS.ONE for Russian market, www.comss.ru.
- NetMarketShare for global market, netmarketshare.com.

B. Desktop Platforms

Table II shows available desktop platforms for development of service clients. The Windows platform is the main competitor on this market. It has many versions (Windows 7, 10, 8.1 are now most popular). The forecast is that the number of Windows 7 users is decreasing since they move to Windows 10. The same trend is the Windows 8 decrease. Likely that MAC OS and Linux will occupy a similar position on the market in near future.

The advantage of the Linux platform is the possibility to access low-level functions. Also, Linux provides a relatively flexible way for programming, since implementation of many functions and libraries is similar to the architecture of this platform. However, the flexibility can lead to ambiguity, which makes programming more difficult in some cases.

In terms of the processing speed, the key competitors are Windows 7 and Linux. If we assume that major data processing is not on the client then the difference between the target platforms becomes negligible.

C. Mobile Platforms

Table III shows possible mobile platforms. The two main leaders are clear: Android devices (Samsung, Huawei, Xiaomi) and Apple devices Apple (on iOS). The share of Windows Phone, Sailfish OS, and Tizen is essentially smaller.

Advanced versions of Android have been released since 2016. They are supported by many applications. iOS is in a similar situation; new versions are released even more frequently than Android. Nevertheless, the transition to the new version of iOS is sharper; many functions may stop working. Sailfish OS seems interesting from research point of view, since it provides easy ways for programming and piloting innovative development.

Although Android version numbers came out some time ago, smartphones with Android 4.x–6.x are still among the most popular devices, as well as being relatively cheap. Many Android devices implement the accessibility to low-level features. In contrast, iOS has almost no access to low-level primitives or the access is limited.

The widely accepted opinion is that iOS provides an easy way for programming. User interface is friendly and simple, leading to intuitive applications. Nevertheless, the Android platform shows fast progress in advancing the programming techniques and user interface mechanisms.

V. Concept Data Models

Consider the data models for using at primary data processing components of the proposed architecture (see Fig. 2 above): video server (data processing) and SIB (semantics and data mining). The presented high-level data modeling solutions follow the requirements introduced in our previous study [11].

A. Data Processing

In the basic configuration, the client on smartphone connects to one or more cameras and performs video data processing of the incoming stream. In this case, KP client also implements the service construction. Due to the low capacity, complicated data processing and data mining are not possible. Thus, the case is limited with simple services, when the KP client implements lightweight video analytics algorithms. In fact, this approach is close to so-called Internet of Video Things [26], i.e., smart activity of cameras is observable as services for clients.

In construction of more advanced services, the major part of data processing has to be performed on a local video server or on remote facilities. The server (KP Video server) receives, processes, and stores video data as well as context information. The data are received from the following participants: KP Clients, KP Sensors, and KP Cameras.

Video data processing is based on existing algorithms and their implementations. In particular, for face recognition, KP Video server receives a stream of images from the camera. Faces in the incoming images can be recognized using improved Haar Cascade Filter [27]. Recognized images are augmented with special visual effects, e.g., an oval surrounds any recognized face on every image. Consequently, the initial stream is modified and can be forwarded to appropriate clients (as a service) or stored at the server for later use.

Therefore, the basic data model is fusing of several data streams (video and other data) into a single video data stream. This stream is subject to:

- direct service provision to clients, which visualize/play the stream to the user or react on detected events,
- collection of data processing results (discovered information) for later use, e.g., offline video or statistics.

This type of video server model is suitable for the needs of At-Home Lab, face recognition, and equipment monitoring, as we discussed in Section III.

B. Semantics for Data Mining

The video server makes data processing based on analysis of individual data flows. The derived results also form some sequences (e.g., video streams, events flow, time-period statistics). On top of this information, the semantic layer can be constructed using the model of semantic network. A node of the semantic network correspond to problem domain flow-based data (e.g., video stream), to a meaningful element in these data (e.g., recognized face), or to a participant (e.g., camera).

The semantic network construction is based on activity of interacting KPs, as illustrated in Fig. 4. Such KPs discover new information and integrate it into the semantic network using SIB. KP cameras and KP Video server provide information discovered in source video data. KP sensors provide context measurements (as analysis results of time series). KP clients provide control information (control of the camera or server storage, processing requests). The collected semantics are shared and further used by KP clients to receive services. For instance, clients can find access details of an appropriate camera for direct communication or detect events for notifying the user.

The following information assigned with the participants can be represented in the semantic network.

- *Client:* identifier (e.g., login name), statistic on previously used cameras, statistics on performed actions (service usage).

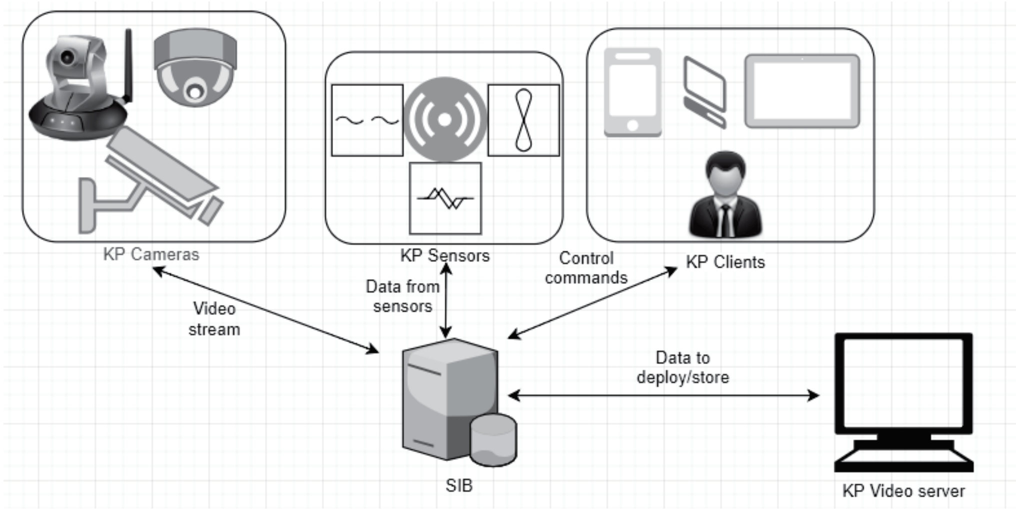

Fig. 4. SIB supports the service construction as connecting resources with their consumers

- *Camera:* network access information, current state (e.g., monitored area), current observations (e.g., recognized people) control queue (commands to execute: turn, adjust brightness, etc.).
- *Sensor:* aggregated characterization of measurements.
- *Video server:* summary characteristics of the video data collection, online video streams.

Therefore, the basic data model is relation of available resources. Using this semantic vision, each client can consume a video data based service from one or more cameras (and augment the video data with sensor measurements), either directly accessing the cameras or indirectly via access of processed information from the video server. In both cases, the SIB implements the rendezvous function when the service is constructed as connecting resources with their consumers.

VI. Conclusion

This paper considered video data based services and the problem of their development for IoT environments. We discussed the possible architecture and its extendible configurations depending on the classes of participating computing devices. We showed potential desktop and mobile platforms available for service clients. We presented concept data models for video processing implemented using a local video server and for data mining implemented using a semantic information broker.

Acknowledgment

The research was financially supported by the Ministry of Education and Science of Russia within project # 2.5124.2017/8.9 of the basic part of state research assignment for 2017–2019. The reported study was funded from Russian Fund for Basic Research according to research project # 16-07-01289. The results were implemented by the Government Program of Flagship University Development for Petrozavodsk State University in 2017–2021.

References

[1] D. Korzun, S. Balandin, and A. Gurtov, "Deployment of Smart Spaces in Internet of Things: Overview of the design challenges," in *Internet of Things, Smart Spaces, and Next Generation Networking*, ser. Lecture Notes in Computer Science, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds., vol. 8121. Springer, Aug. 2013, pp. 48–59.

[2] L. Roffia, F. Morandi, J. Kiljander, A. D. Elia, F. Vergari, F. Viola, L. Bononi, and T. Cinotti, "A semantic publish-subscribe architecture for the Internet of Things," *IEEE Internet of Things Journal*, vol. PP, no. 99, 2016.

[3] D. G. Korzun, S. I. Balandin, A. M. Kashevnik, A. V. Smirnov, and A. V. Gurtov, "Smart spaces-based application development: M3 architecture, design principles, use cases, and evaluation," *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, vol. 8, no. 2, pp. 66–100, 2017.

[4] S. Wu, J. B. Rendall, M. J. Smith, S. Zhu, J. Xu, H. Wang, Q. Yang, and P. Qin, "Survey on prediction algorithms in smart homes," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 636–644, June 2017.

[5] D. Korzun, "On the smart spaces approach to semantic-driven design of service-oriented information systems," in *Proc. 12th Int'l Baltic Conf. on Databases and Information Systems (DB&IS 2016)*. Springer International Publishing, Jul. 2016, pp. 181–195.

[6] M. Gusev and S. Dustdar, "Going back to the roots–the evolution of edge computing, an IoT perspective," *IEEE Internet Computing*, vol. 22, no. 2, pp. 5–15, Mar 2018.

[7] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, Sep. 2015.

[8] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[9] A. Meigal, K. Prokhorov, N. Bazhenov, L. Gerasimova-Meigal, and D. Korzun, "Towards a personal at-home lab for motion video tracking in patients with Parkinson's disease," in *Proc. 21st Conf. Open Innovations Association FRUCT*, Nov. 2017, pp. 231–237.

[10] L. Köping, K. Shirahama, and M. Grzegorzek, "A general framework for sensor-based human activity recognition," *Computers in Biology and Medicine*, vol. 95, pp. 248–260, 2018.

[11] N. A. Bazhenov and D. G. Korzun, "Use of everyday mobile video cameras in IoT applications," in *Proc. 22nd Conf. Open Innovations Association FRUCT*, May 2018, pp. 344–350.

[12] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the Internet of Things," *IEEE Internet Computing*, vol. 14, no. 1, pp. 44–51, Jan. 2010.

[13] C. Streiffer, A. Srivastava, V. Orlikowski, Y. Velasco, V. Martin, N. Raval, A. Machanavajjhala, and L. P. Cox, “ePrivateEye: To the edge and beyond!” in *Proc. 2nd ACM/IEEE Symposium on Edge Computing (SEC ’17)*. New York, NY, USA: ACM, 2017, pp. 18:1–18:13.

[14] S. Yi, Z. Hao, Q. Zhang, W. Shi, and Q. Li, “LAVEA: latency-aware video analytics on edge computing platform,” in *Proc. IEEE 37th Int’l Conf. on Distributed Computing Systems (ICDCS)*, June 2017, pp. 2573–2574.

[15] G.-W. Kim and J.-W. Han, “Security model for video surveillance system,” in *Proc. Int’l Conf. on ICT Convergence (ICTC)*, Oct. 2012, pp. 100–104.

[16] L. Meinel, M. Findeisen, M. He?, A. Apitzsch, and G. Hirtz, “Automated real-time surveillance for ambient assisted living using an omnidirectional camera,” in *2014 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2014, pp. 396–399.

[17] A. Gurtov, M. Liyanage, and D. Korzun, “Secure communication and data processing challenges in the Industrial Internet,” *Baltic Journal of Modern Computing*, vol. 4, no. 4, pp. 1058–1073, 2016.

[18] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos, “A survey on ambient intelligence in healthcare,” *Proceedings of the IEEE*, vol. 101, pp. 2470–2494, Dec. 2013.

[19] J. Bugeja, A. Jacobsson, and P. Davidsson, “On privacy and security challenges in smart connected homes,” in *2016 European Intelligence and Security Informatics Conference (EISIC)*, Aug 2016, pp. 172–175.

[20] D. L. Xu, W. He, and S. Li, “Internet of Things in industries: A survey,” *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[21] T. Marciniak, A. Chmielewska, R. Weychan, and A. Dabrowski, “Fast prototyping of automatic real-time event detection facilities for video monitoring using DSP module,” in *Proc. 20th Int’l Conf. Mixed Design of Integrated Circuits and Systems (MIXDES 2013)*, June 2013, pp. 473–478.

[22] D. Korzun and S. Balandin, “Personalizing the internet of things using mobile information services,” in *UBICOMM 2016 : The Tenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2016, pp. 184–189.

[23] A. S. Dhavalikar and R. K. Kulkarni, “Face detection and facial expression recognition system,” in *2014 International Conference on Electronics and Communication Systems (ICECS)*, Feb 2014, pp. 1–7.

[24] J. Wang, B. Amos, A. Das, P. Pillai, N. Sadeh, and M. Satyanarayanan, “A scalable and privacy-aware IoT service for live video analytics,” in *Proc. 8th ACM on Multimedia Systems Conference (MMSys’17)*. New York, NY, USA: ACM, 2017, pp. 38–49.

[25] V. Goyal and K. S. Umadevi, “Simple architecture for smart home using Internet of Things,” in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, April 2017, pp. 1–4.

[26] A. Mohan, K. Gauen, Y. H. Lu, W. W. Li, and X. Chen, “Internet of video things in 2030: A world with many cameras,” in *Proc. IEEE Int’l Symp. on Circuits and Systems (ISCAS 2017)*, May 2017, pp. 1–4.

[27] L. Cuimei, Q. Zhiliang, J. Nan, and W. Jianhua, “Human face detection algorithm via Haar cascade classifier combined with three additional classifiers,” in *2017 13th IEEE International Conference on Electronic Measurement Instruments (ICEMI)*, Oct 2017, pp. 483–487.