IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 21, 2018, accepted June 20, 2018, date of publication July 4, 2018, date of current version July 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2852295

A Study of Robotic Cooperation in Cloud Robotics: Architecture and Challenges

WUHUI CHEN[1], YUICHI YAGUCHI[2], KEITARO NARUSE[2], YUTAKA WATANOBE[2], KEITA NAKAMURA[3], AND JUN OGAWA[3], (Member, IEEE)

[1]School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China
[2]School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Japan
[3]Revitalization Center, The University of Aizu, Aizuwakamatsu 965-8580, Japan

Corresponding author: Wuhui Chen (chenwuh@mail.sysu.edu.cn)

ABSTRACT Networked robotics involves a collection of robots working together to perform complex tasks, such as search and rescue task in disaster management. Because such tasks are beyond the capacity of a single powerful robot, networked robotics has been widely researched. However, the modes of cooperation in traditional networked robotics have been restricted by the inherent physical constraint that all computations are performed in the robotic network, with knowledge sharing being limited to the collective storage in the network. Cloud robotics, which allows robots to benefit from the rich storage, computation, and communication resources of modern data centers, is widely accepted as a promising approach to efficient robot cooperation in applications, such as disaster management. In this paper, we study robotic cooperation in cloud robotics. We first give a conceptual view of the nature of this cooperation. We then propose three novel robotic cooperation frameworks for cloud robotics: robotic knowledge sharing cooperation, robotic physical-task cooperation, and robotic computation task cooperation. Finally, we identify several critical challenges, and illustrate the potential benefits of robotic cooperation in cloud robotics.

INDEX TERMS Cloud robotics, multi-robot system, robot cooperation, quality of service.

I. INTRODUCTION

Networked robotics involves a collection of robots working together to perform complex tasks that are beyond the capacity of a single powerful robot. Because there are many such tasks, networked robotics has been widely researched. In particular, the task of disaster management has become a crucial and urgent research issue [1]–[4]. As one example, Figure 1 shows the networked robotics system that was used at the Fukushima Daiichi nuclear power plant (NPP) after the Great East Japan earthquake disaster [1]. It comprised one relay station car, one operation car, one bulldozer, two dump trucks, two backhoes, and seven camera cars. It was used to remove debris in the interior and exterior areas of the nuclear reactor buildings, to inspect buildings, and also to monitor irradiation levels. Another example is the collaborative mapping of a building damaged by the Great East Japan earthquake at Tohoku university in Sendai city that used a team of networked ground and aerial robots [4]. However, although robots in such a robotic network could share their knowledge and computation workloads with other robots in the same network, the overall effectiveness of the robotic network is still limited to storage space, the processing power, and number and type of sensors and camera in the robots themselves [5]. That is, the robotic cooperation in such a robotic network is restricted by the inherent physical constraints of the collective computation and knowledge sharing capacity of the computation and storage resources of the robotic network.

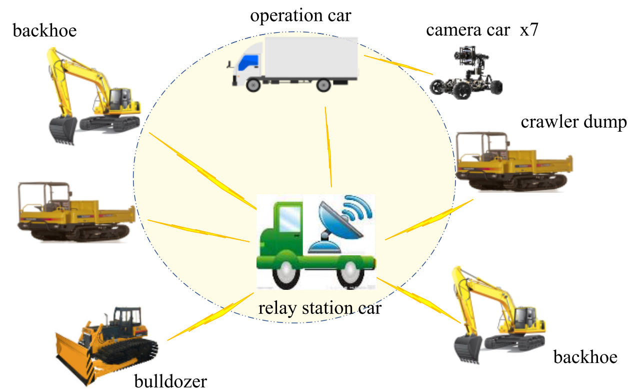


FIGURE 1. A networked robotics system to remove debris at the Fukushima Daiichi nuclear power plant [1].

To overcome the limitations of simple networked robotics, cloud robotics [6], which leverages emerging cloud-computing technologies to extend networked robotics to include the elastic resources of a cloud computing infrastructure, is now widely accepted as a promising approach to improve efficiency in networked robotics. The academic research project RoboEarth [7] has been developed as a giant cloud-based database where various robots can share information about environments, tasks and objects. The data set is generated by both humans and robots in a machine-readable format that includes maps for navigation, object-recognition models, task knowledge, and software components. Another academic research project is RoboBrain [8], which is a large-scale knowledge engine deployed in the cloud that learns and shares representations of knowledge for various robots to enable a variety of tasks. Its knowledge in a graph structure derives from multiple sources. These include the physical interactions that robots encounter while performing diverse tasks (perception, natural language understanding, planning, and control), and large-scale knowledge bases learned from the Internet. As an industrial example of cloud robotics, Google's self-driving cars [9] are one type of cloud-connected vehicles. The autonomous cars access data from Google Maps and images stored in the cloud computing platform to recognize their surroundings (such as road and weather conditions), collect information about road and traffic conditions using cameras, radar, lidar, and a wide range of vehicle sensors, and send the preprocessed information back to the cloud. Another industrial example is Amazon's Kiva Systems robots [10] for warehouse logistics, which interact with central servers to coordinate routing and update the shared information about their environment. However, most current research is focusing on either knowledgebase building and database construction or computation offloading for parallel robotic-task processing. There has been no study of the robot-cooperation problem in cloud robotics for complex tasks such as disaster management.

In this article, we study the application of cloud-robotics concepts to robotic cooperation in complex tasks such as disaster management. To the best of our knowledge, this is the first attempt to study the robot cooperation problem in cloud robotics. We first give a conceptual view of cloud robotics for robot cooperation. We then propose three novel robotic cooperation frameworks for cloud robotics: robotic knowledge sharing cooperation, robotic physical-task cooperation, and robotic computation task cooperation. Finally, we identify several critical challenges, including quality of service issues, communication issues, safety issues, heterogeneity, and security issues, and illustrate the potential benefits of robotic cooperation in cloud robotics.

The remainder of this paper is organized as follows. In Section 2, we introduce a motivating example. In Section 3, we discuss robotic cooperation architecture. In Section 4, we propose three novel robotic cooperation frameworks. The potential benefits and challenges of robotic cooperation in cloud robotics are discussed in Section 5, with Section 6 concluding the article.

II. MOTIVATING EXAMPLE: DISASTER MANAGEMENT

Consider the following scenario: after a large earthquake disaster, one of the most urgent tasks is to search for and rescue survivors in the disaster site within the critical first 48 hours. However, the mechanics of how large structures collapse in a disaster site often prevent rescue workers (however heroic) from searching buildings because of the unacceptable personal risk from further collapse. Moreover, earthquakes often leave damage so widespread that those injured can often be inaccessible by search and rescue teams. An alternative is to use search-and-rescue robots, which have already been proven to play a vital role in getting life-saving aid to those in need [1], [3]. The benefits of rescue robots include reduced fatigue, reduced personnel requirements, and access to human-unreachable areas. For robots to work well in disaster sites, they must satisfy two main requirements. First, they must be well-tested and robust for tasks in disaster sites. Robots that have demonstrated their utility include small unmanned aerial vehicles such as robotic helicopters and quadcopters, "snake" robots capable of slithering through rubble and entering collapsed buildings, and tethered unmanned ground vehicles such as sensor-packed wheeled robots. The second main requirement is that they must be able to move about without impacting on the physical environment, while sensing the surrounding environment in their search for survivors. For example, aerial vehicles could be used for the inspection of the lower altitude checks and upper levels of buildings with onboard cameras and lasers. Snake robots could be sent to collapsed buildings to search for survivors inside with sound sensors, cameras, and lasers. Unmanned ground vehicles could carry cameras in addition to infrared and carbon-dioxide sensors for searching survivors trapped under collapsed buildings.

However, to improve the efficiency of search-and-rescue robot teams, they should be able to mutually cooperate within the disaster site. For example, aerial vehicles could provide 3D maps to snake robots and unmanned ground vehicles for navigation, unmanned ground vehicles could be used to remove debris to enable snake robots to move safely and quickly, and snake robots and unmanned ground vehicles should work cooperatively to search for survivors by sharing information from different regions.

The workflow shown in Figure 2 explains how aerial vehicles can provide 3D maps to snake robots and unmanned

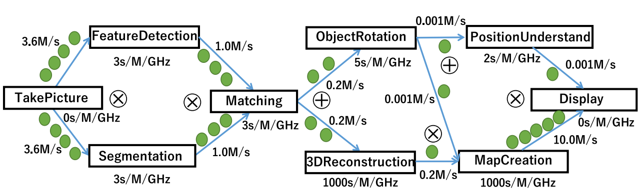


FIGURE 2. A streaming workflow for map construction.

ground vehicles. First, the aerial vehicles acquire picture data in the disaster sites where the snake robots and unmanned ground vehicles are moving. Next, algorithms process the picture data to generate the 3D Map. These processes include *FeatureDetection*, *Segmentation*, *Matching*, *ObjectRotation*, *3DReconstruction*, *PositionUnderstand*, and *MapCreation*. If the algorithms require too much computation power, they could be distributed and executed not only in aerial vehicles, but also in snake robots, unmanned ground vehicles, or the remote cloud. Finally, after the data processing, the 3D Map is sent to the snake robots and unmanned ground vehicles to aid their navigation.

Networked robotics systems have become an active research topic in the recent years. These research projects focus on having a number of robots and cooperating to complete the required task(s). Through robotic cooperation, some goals that were impossible for a single powerful robot to achieve become attainable. A networked robot can share its knowledge and computational workloads with other neighbor robots in the same network, but the overall effectiveness of the robotic network is constrained by its overall storage space, processing power, and number and type of sensors in the network [5], [11]–[13]. Data-processing latency would also affect the quality of robotic cooperation. For example, although aerial vehicles can provide 3D maps to snake robots and unmanned ground vehicles for navigation, if the map construction takes too long, the snake robots and unmanned ground vehicles may not be able to work effectively.

Networked robotics is ultimately a human-oriented system in which all operations are controlled remotely by human operators. However, a lack of operators was one of the most significant problems during the Fukushima accident. There were only about 20 skilled operators for unmanned construction machines throughout Japan, and it was impossible to place all the skilled operators at the Fukushima Daiichi NPP [1]. Moreover, earthquake disaster mitigation requires rapid and efficient search-and-rescue operations. Robotic teams have only 48 hours to search trapped survivors, after which the likelihood of finding alive victims is very small [2]. To mitigate these weaknesses, the use of cloud robotics is considered to offer improvements in robotic cooperation by extending the available computation and communication capacities. Therefore, in the following sections, we discuss robotic cooperation in cloud robotics.

III. CLOUD ROBOTICS COOPERATION ARCHITECTURE

A. SYSTEM OVERVIEW

Fig. 3 illustrates a generic architecture for cloud robotics, comprising a set of robot clusters and a set of infrastructure-based clouds. A robot cluster is formed by a dynamic set of neighboring robots that work cooperatively to accomplish complex tasks. As a robot moves from one environment to another, it can seamlessly join a different cluster of robots and can still benefit from cloud resources. A robot can communicate with its peers using messages transmitted over an ad-hoc wireless network via a local wireless network

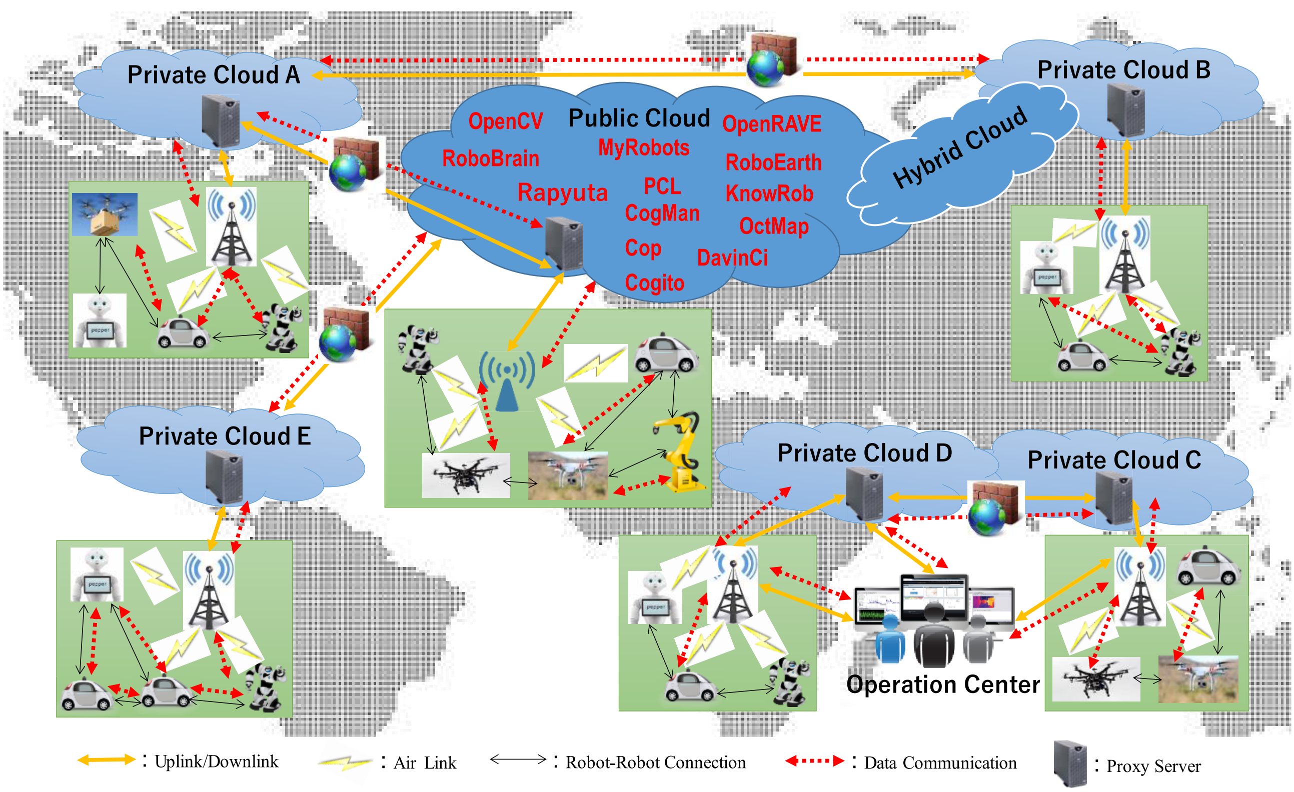


FIGURE 3. A conceptual view of cloud robotics: robots nearby are interconnected as robot cluster and also connected to a remote cloud.

interface (e.g., Bluetooth, 3G, WIMAX, LTE, WiFi, ZigBee). The link between any two peer robots might be direct or indirect. If two peer robots can send and receive messages directly to each other, then there is a direct link between them. Otherwise, intermediary peers are required to forward transmitted messages, leading to an indirect link between the two peer robots. In this case, the intermediary peers act as routers. Many benefits flow from forming a collaborative computing fabric. First, individual robots that are not within communication range of a cloud access point are allowed to send computation offloading requests to the cloud or access information stored in the cloud in a collaborative fashion. Second, among the collaborative robots, information can be shared and exchanged for distributed decision-making in various cooperative tasks. Finally, a virtual ad-hoc cloud can be formed for computation-intensive tasks by pooling the computational capability together from collaborative robots.

Complementing the robot-cluster infrastructure, the cloud infrastructure comprises a pool of shareable computation and storage resources, which are virtualized and provided as service. Figure 3 shows proxy-server components being deployed between the robots and the cloud. To access cloud services, a robot can simply send a query to a nearby proxy server. Because of their limited power supply and capacity for onboard processing and storage, individual robots are constrained by numerous limitations on computational complexity and knowledge sharing. Therefore, performing complex computations or storing large-scale knowledge bases via a remote cloud infrastructure is considered as a promising solution that can take advantage of the powerful cloud resources to help collect, process, and store data for robots, thereby requiring less computing power, memory, and battery capacity on the robot itself. Many benefits flow from using this elastic computing model. First, computation-intensive tasks can be offloaded for remote execution from networked robots, resulting in ''remote-brain'' robots. Second, a large volume of information about the environment (such as map, and object models) stored in cloud can be unified and organized in a format usable by robots to achieve knowledge exchange of higher quality. Finally, an extensive library of behaviors or skills that are related to situational complexities and task requirements can be provided, making it possible to learn from the history of other cloud-enabled robotic activities.

In summary, cloud robotics leverages a combination of different levels of architecture: stand-alone robots, ad-hoc robot clouds formed by a robot cluster, and an infrastructure-based cloud. For tasks that do not require additional robots or complex computation, such as welding, assembly, painting, and packaging, stand-alone robots have been successful applied because of their high precision, speed, and endurance in both static and structured environments. An ad-hoc robot cloud can be a backup solution when a group of robots in a cluster are unable to connect a cloud access point (e.g., robots in areas of damage after a disaster). Even with good network access, an ad-hoc robot cloud may be desirable for tasks with low-latency requirement because transferring large volume of data to a remote cloud could cause a large delay. An infrastructure-based cloud allows robots to benefit from the rich communication, storage, and computation resources of modern data centers.

Besides, all three levels of architecture can collaborate with each other and might not work as effectively if one of them is missing. For example, robotics networks are like stand-alone robots in that they face some inherent physical limitations. Because of the robot's size and other factors, there are obvious limitations to the storage and computation capacity of individual robots. This implies a limited capacity by traditional networked robotics to undertake highly complex processing tasks. Fortunately, an infrastructure-based cloud can solve this problem well by taking advantage of the rich cloud resources to help process, and store large volumes of data for the robots. Secondly, infrastructure-based cloud robotics might become unavailable because of network problems. If a robot were to rely too much on the remote cloud, a fault in the network could leave it ''brainless''. Here, the use of a robot cluster could be a backup solution that shares processing resources across a group of individual robots to solve problems that previously needed help from a remote cloud. It would be independent of connections to an outside network, thereby avoiding possible communication bottlenecks and improving response times. As a result, the ad-hoc virtual cloud provides a solution to sporadic wireless network connectivity. Therefore, by using this computing framework, the stand-alone robots, the ad-hoc virtual cloud, and the infrastructure-based cloud can complement each other in addressing the issues of limited battery power, shortage of processing and storage resources on the robots, and sporadic network connectivity between robots and the remote cloud.

B. TOWARDS HIGHER-QUALITY ROBOTIC COOPERATION: FROM NETWORKED ROBOTICS TO CLOUD ROBOTICS

Cooperation can refer to the situation whereby a number of robots need to interact each other in completing certain tasks, which increases the total utility of the system. Alternatively, cooperation can be considered an interaction between robots, which works together towards a common interest or reward [14]. Cooperating robots have a joint goal, which includes various sub-goals. The advantages of robotic cooperation in robot clusters, such as multi-robot systems and networked robotics, have been studied in [15]. Some representative examples of multi-robot cooperation are as follows: task planning, motion coordination, localization, exploration, search-and-rescue, and object transportation and manipulation [16]. However, although a robot in a robotic network can share its knowledge and computational workloads with other robots in the same network, the overall effectiveness of the robotic network is constrained by the limitation of the storage and computing capacity of the individual robots. This leads to a bottleneck in research and development when considering tasks that involve highly complex processing [5].

With the introduction of cloud robotics, the above constraints can be overcome via the elastic and powerful

TABLE 1. Comparisons of state-of-the-art approaches.

Category	Computing model	Resource level	Latency mainly caused by	Energy mainly caused by	Intelligence Level	Application
Stand-alone robots	On-board computing	Weak	Computing workload	Computing workload	Low	Static and structured tasks
Networked robotic system	Ad-hoc cloud computing	Medium	Computing and communication workloads	Computing and communication workloads	Medium	Real-time processing
Cloud robotic system	Hybrid computing (both cloud computing and ad-hoc cloud computing)	Strong	Communication workload	Communication workload	High	Both real-time processing, and resource-intensive applications

resources of an infrastructure-based cloud, as shown in Table 1. Cloud robotics, which brings ad-hoc cloud computing and centralized cloud computing models together, and combines their advantages to release the resource constraint of traditional networked robotics, is able to effectively support real-time processing and resource-intensive applications for more intelligent and efficient robotic cooperation. It can perform resource-intensive applications using ad-hoc cloud computing for communication-intensive tasks and using centralized cloud computing for computation-intensive tasks. The major advantages of robotic cooperation in cloud robotics are as follows.

- The ability to gather and then share knowledge. In cloud robotics, robots have the potential to instantly share all the skills they have learned from other robots simply by transmitting the information over a network.
- The ability to conquer complex tasks. The cloud (acts as a "brain" for robots to collaborate and achieve complex tasks) can provide both a powerful streamed-data processing capacity to identify the surrounding environment and a large knowledge base to support appropriate decision-making, enabling heterogeneous robots to cooperate with each other effectively.
- Computation offloading. In cloud robotics, robots can migrate computation-intensive tasks from the robot side to an ad-hoc virtual cloud formed by a cluster of robots, or to an infrastructure-based cloud. This is particularly significant for mobile robots that might have low computation, storage and energy capabilities.

Cloud robotics allows robots to benefit from the rich storage, computation, and communication resources of the ad-hoc virtual cloud and infrastructure-based cloud, enabling higher quality robotic cooperation in various applications. However, although the benefits of robotic cooperation in cloud robotics is obviously attractive, robotic cooperation in cloud robotics has yet to be thoroughly studied. Therefore, in the following, we propose three robotic cooperation frameworks for cloud robotics.

IV. FRAMEWORK FOR CLOUD ROBOTICS COOPERATION

In this section, we propose three novel robotic-cooperation frameworks in cloud robotics, namely knowledge-sharing cooperation, physical-task cooperation, and computational-task cooperation.

A. ROBOTIC KNOWLEDGE COOPERATION

Many complex tasks humans would like robots to perform, such as assisting bedridden patients, or packing items in warehouses, are not yet possible because existing robots cannot recognize or easily handle common objects. In general, human beings have no trouble packing items, because we have all gone through "a big-data collection process" called childhood. For robots to perform the same types of routine task, they would also need access to the vast amounts of knowledge associated with grasping and manipulating objects. However, where would all that knowledge come from? It has usually to come via painstaking programming. Ideally, robots could share learned information with each other [7].

Search and rescue robots play a significant role in saving people from natural and man-made disasters such as earthquakes hurricanes, and terrorist attacks. Allowing these rescue robots to cooperate with each other by sharing knowledge and learning from each other can open up a realm of new possibilities.

Consider the following scenario. A search and rescue robot (Rescuer #1) in a disaster site has been programmed to perform a medical service (providing water) to a survivor trapped under a building, so that the survivor is better able to wait for rescue. This task includes locating a bottle of water delivered by an unmanned aerial vehicle in advance, navigating to the bottle's position, grasping, picking up the bottle, locating the survivor trapped under a building, navigating to the survivor, and delivering the bottle to the survivor. Suppose that during task execution, Rescuer #1 monitors and logs its progress, thereby continuously updating and extending its rudimentary, preprogrammed-world model with additional data. It updates and adds the positions of detected objects, evaluates the correspondence between its map and its actual perception, and logs successful and unsuccessful attempts during the performance of its task. If the robot is not able to perform a task, it asks a remote operator for help while storing its newly acquired knowledge. At the end of the search-and-rescue task, the robot shares its acquired knowledge to others by uploading it to a remote distributed database in the cloud.

Sometime later, the similar task is to be executed by a second search-and-rescue robot (Rescuer #2) that has no prior knowledge about how to execute the task. Rescuer #2 queries the database for relevant information and downloads the knowledge previously collected by Rescuer #1. There might be differences between the two robots, such as wear and tear or different robot hardware, or in the environment, such as changed object locations or a different disaster site. This means that the downloaded information by itself may not be sufficient to allow Rescuer #2 to repeat a previously successful task. Nevertheless, this information can provide a useful a-priori starting point. Recognized objects, such as the collapsed building, can now provide rich disaster-site knowledge even for areas not directly observed by Rescuer #2. Detailed object models, such as the bottle of water or a collapsed building in the disaster site, can increase the reliability and speed of Rescuer #2's interactions. Task descriptions of previously successful actions, such as driving around the disaster site and overcoming obstacles on the way, can provide guidance on how Rescuer #2 may be able to successfully perform a given task.

This and other prior knowledge, such as the previous location of the bottle or that the collapsed building is a likely place to find the survivor, can help Rescuer #2's search-and-rescue strategy. In addition, as Rescuers #1 and #2 continue to execute their tasks, gather their data, and share their knowledge to each other, the quality of prior knowledge improves and begins to reveal underlying patterns and hidden correlations about the robots and their environment.

Robotic cooperation on knowledge sharing instantly shares all the skills robots have learned with other robots simply by transmitting the information via the cloud, to speed up robot learning and to allow robots to perform well beyond their preprogrammed behaviors. This is a major extension to the capabilities of robots. With the introduction of large scale knowledge in the cloud, even a small amount of uploaded uncompleted characteristic data can enable matching to a cloud-based scenario. Then the characteristic data for the complete scenario can be downloaded and sent back to the robots for execution of the operation.

Figure 4 shows a conceptual diagram of robotic cooperation in knowledge sharing. The framework has two main elements: a set of groups of robots and a remote centralized cloud. Robots are heterogeneous in nature with a wide variety of hardware, OS, platforms and communication standards. Because modern robots are typically equipped with sensors that are capable of sending environmental information, they can act not only as cloud-service consumers, but also as data sources and producers of cloud-service information. At any one time, some groups of robots may be responsible for collecting data, while other groups are receiving cloud services. On the cloud side, we can identify four layers, as shown in Figure 4 and described below.

1) COOPERATIVE ROBOT COMMUNICATION (CRC) LAYER

The CRC layer is responsible for connecting individual robots with the message broker. Because the remote cloud is designed to serve heterogeneous robots, it needs a "gateway" component to record specific information about the individual robots and the map between message-broker channels and native robotic-data channels. All individual robots are connected through these gateways. Below the gateways are

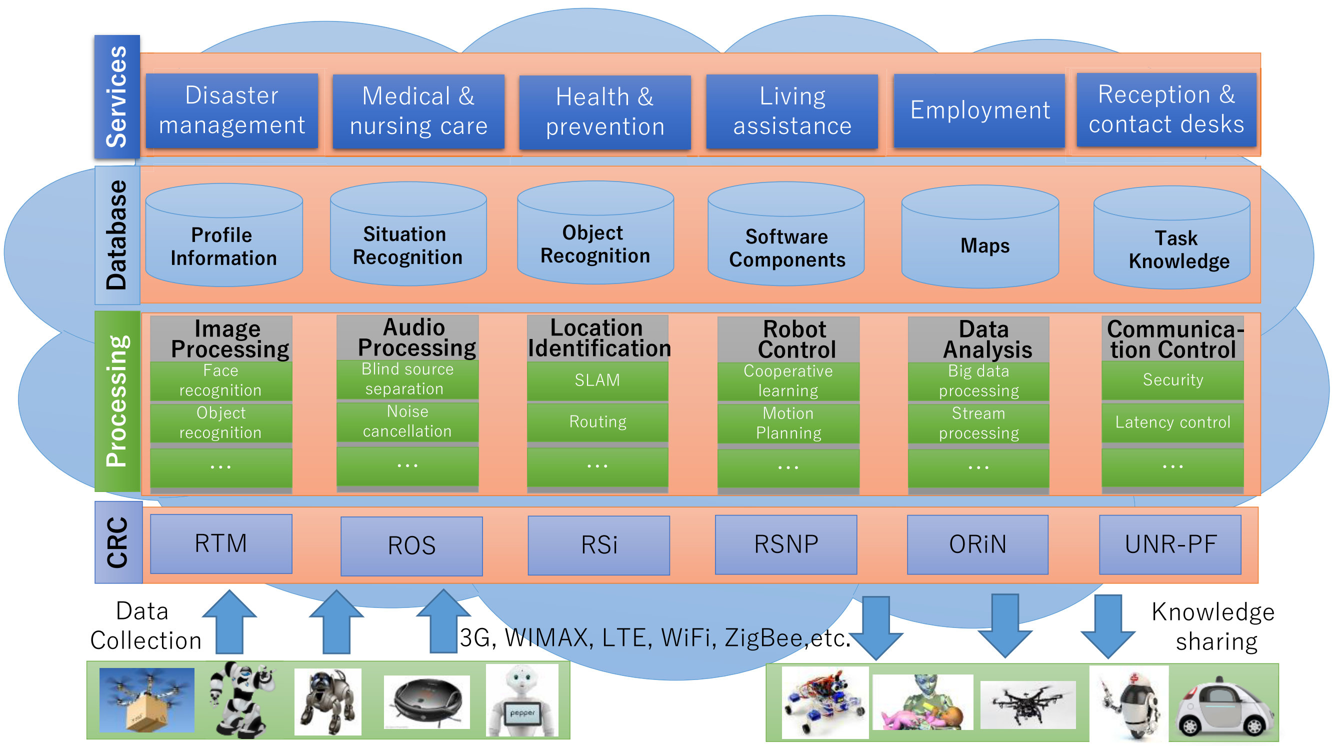


FIGURE 4. A framework of robotic cooperation on knowledge sharing.

remote cloud-robotic drivers that convert robotic data into messages that the cloud services can process. The cloud-robotic drivers first obtain data from individual robots and then use application-programmer interfaces (APIs) to send converted data to the remote cloud. The gateway maintains connections between individual robots and the cloud. Besides, there is a gateway master that coordinates multiple gateways and registers connection information, such as data channel mappings between message brokers and individual robots, enabling the CRC layer to discover robots and provide service entries.

Unlike traditional web applications, stateful communication protocols in the CRC layer, such as the publish/subscribe model, are required to push information asynchronously from the remote cloud to the robots. The cloud-robotic drivers subscribe to robot operating system (ROS) topics to obtain robot-state data, such as odometry data and laser scans. They then convert the data into messages that can be transmitted through a broker to the cloud. After the data-processing procedures, the remote cloud sends back velocity commands through the message broker. The remote cloud driver converts each velocity command message into a ROS message that is then sent to the correct ROS topic to enable the robot to be controlled. To distinguish the messages sent from different robots, a unique robot ID generated by the cloud driver is attached to the message. For different types of robots, the cloud drivers are different, but for the same type of robots, the same cloud driver can be used, and it only needs to spawn a new driver instance for each individual robot. Using the robot ID attached in the message, the gateway creates a command queue for each cloud driver instance, and sends each message to the correct queue for the target robot.

2) PROCESSING LAYER

It is increasingly preferred for information from various sources to be integrated into a single common shared repository called a ''data lake''. The job of the processing layer is to process this lake of big data, handling issues of volume, velocity, and variety, by providing a rich set of features for preparing and enriching data before storing or publishing them. The data-processing layer is used to filter, cluster, and process the collected data from robots, and to prepare the required data for storage and analysis. Its main components are as follows.

- Real-time streamed-data processing. This component receives streamed data (camera data and sensor data from robots) in small chunks (a few rows per second/minute interval), and reacts to the streamed data as it arrives with low latency. The reaction is often the result of applying a set of rules to the arriving streamed data, in the context of data that has been precomputed in the data store. Streamed data in this component is characterized by incremental changes or events.
- Batch data processing. This component transforms data (such as aggregation, mapping, calculating, filtering, or joining) on a periodic basis. It is responsible for heavy-duty or expensive processing, creating data in a format useful for downstream users of the data (robots or remote operators). Its operations are fairly complex in nature (characterized by re-computations or complex aggregations) and typically take place over a very large and growing set of records.
- Data enrichment processing. This component makes use of data from multiple sources in different robots, and combines the data after the integration phase (entity recognition, duplication removal, and matching), resulting in enriched data. Data enrichment may be performed by combining data from multiple sources, including public data sources and data stored in the database layer on the cloud side.

3) DATABASE LAYER

The database layer's data lake stores a global world model, including information on software components, environments (such as object locations, and maps for navigation), object recognition models (such as point clouds, images, and models), and task knowledge (such as action recipes and skills, and manipulation strategies). The database layer provides a key resource for enabling robots to store and share information collaboratively. Such knowledge bases in the cloud enable robots to cope with the complexities of human environments and offer a powerful approach to increase the speed of learning by leveraging the experience of other robots. The Willow Garage Household Objects Database, the MIT KIT object data set, and the Columbia Grasp data set are available online and have been widely applied to evaluate different aspects of grasping algorithms, including robust grasping, grasp stability, and scene understanding [11]. RoboBrain [8] is a large-scale robotic knowledge system, which learns and shares knowledge representations from real-life robot trials, computer simulations, and publicly available Internet resources, to enable robots to perform a variety of tasks. RoboEarth [7] is a cloud-based knowledge base designed for the share and exchange of knowledge among robots, by encoding the object descriptions, environment information, task structure, and commonsense knowledge in a formal knowledge base. By storing and sharing knowledge about software components, maps, objects, and actions in a common framework, RoboEarth enables different robots to exchange and collectively improve their knowledge.

4) SERVICE LAYER

Human beings share their experiences and expertise through language and demonstration, whereas robots have the potential to instantly share all their learned skills with other robots via robotic knowledge base in the cloud, so that other robots could increase the speed of leaning by reusing the shared knowledge. Consider a robot that has found an object that it has never encountered before. The robot could simply send an image of the unknown object to knowledge base in the cloud via a API and receive the object's information, 3D model, and detailed instructions on how to handle it from the API. The service layer provides a mechanism for enabling any robot with a network connection to generate, reuse, and share information in different domains (such as disaster management, medical and nursing care, and living assistance) through a set of standardized APIs. Typical everyday manipulation tasks are vaguely specified, and the robot must therefore infer by itself how to carry out the appropriate actions on objects in an appropriate manner to accomplish these tasks. These inferences are only feasible if the robot has access to the necessary knowledge base through APIs provided in service layer, including precise information about an object's model, properties, location, and what might happen if particular actions are performed on it. Such knowledge acquired through APIs in service layer could reduce the need for meticulous reprogramming, and it could enable robots to adapt quickly and accurately when facing with an unfamiliar setting or a new task by the sharing, reuse, and learning of knowledge from others' experiences.

Based on the concepts of robotic cooperation in knowledge sharing, knowledge-based solutions such as RoboBrain and RoboEarth are becoming popular research tools. Although these solutions have obtained some positive results for smarter robots, there remain some challenging research problems in this area to be addressed.

First, not all the knowledge robots could potentially learn is easily exchangeable via a joint knowledge repository. Raw trajectory data or actuator and sensor parameters are often too hardware-specific to be shared successfully. Standardization is therefore one widespread challenge to create outward homogenization of heterogeneous entities to enhance quality of service (QoS). Robotic middleware (such as OpenRTM, and ROS) could interpose an abstraction layer and act as an arbiter to smooth out the impact of heterogeneity for knowledge sharing, enabling communication between heterogeneous robotic systems. However, contemporary middleware requires considerable research and development before becoming an appropriate heterogeneity-handling technique in cloud robotics. In addition, a bridge component for different middleware, such as the OpenRTM–ROS bridge component, is needed to build interoperability and to enable seamless communication among heterogeneous robotic middleware.

A second challenge involves semantic technology. This is another distinctive research field for bridging and interconnecting heterogeneous data, applications, and processes by encoding meanings and providing abstraction layers [17]. Semantic systems based on logic and knowledge can surpass human deductive abilities. With a powerful ability to create semantic links between mined data, such a system can present logical results from big-data warehouses [18]. Therefore, semantic technology in cloud robotics has the potential to provide context-awareness, data portability, interoperability, flexibility, and scalability across heterogeneous environments and robotic systems.

Finally, the complicated and nuanced nature of human environments cannot be summarized within a limited set of specifications, but will require diverse robots to systematically share data and build on each other's experiences. Today, thousands of robotic systems solve the same core problems over and over again. We believe that the benefits of storing, reusing, and sharing information in robotic systems will enable the robots to perform successfully in increasingly new and complex tasks and in unstructured environments.

B. ROBOTIC PHYSICAL-TASK COOPERATION

The complexity of many environments may require teams of possibly heterogeneous robots to work together to carry out a mission that no individual robot can accomplish alone. However, it is not yet possible for robot teams to be aware of their dynamically changing environments cooperatively, and to make appropriate real-time decision and reaction seamlessly. This is because robots find it difficult to recognize and deal with unexpected and unfamiliar situations. Human beings usually have no trouble controlling their hands and legs to accomplish complex tasks in complex environments because the powerful human brain can process streamed environmental data and make quick decisions about cooperation between hands, legs, and other body parts. For cooperating robots to perform as well, they will also need a powerful ''brain'' that can process data in real time and have access to rich knowledge. What might constitute that ''brain''? The cloud can be considered a remote robot ''brain'' that can enable heterogeneous robots to cooperate with each other effectively by offering a powerful computation capacity and a large knowledge-storage capacity.

On March 11, 2011, at 2:46 p.m. local time, a magnitude-9.0 earthquake struck off the East coast of Japan and an ensuing huge tsunami hit the Fukushima Daiichi NPP. This resulted in a hydrogen explosion, which affected a very wide area in Fukushima prefecture. Light debris was scattered along the roadway surrounding the power plant far from the buildings. Pieces of radioactive debris varying in size and shape were also piled on top of trailers and cars around the turbine and reactor buildings, which hampered access to the reactor buildings by fire-extinguishing vehicles and concrete pumping trucks. Moreover, it was strictly prohibited for workers to enter the buildings because of the high radioactivity. As a result, the amount of damage within the buildings was unknown.

To recover from such an urgent accident, it is essential to access the buildings and investigate the type and amount of damage. Therefore, the mobilization of many different types of tele-operated robotic systems was urgently needed in this case. For example, just after the accident occurred, a remotely operated system comprising an operation car, a relay station car, a monitoring camera car, and a concrete pumping track were brought into operation to pour water into the spent nuclear fuel pool. This was considered one of the most urgent and critical tasks immediately after the accident occurred, because if all the water in the pool had evaporated, most of the fuel in the pool would have melted down, producing levels of radioactivity so high that recovery would have been impossible. Another remotely operated system, available from the beginning of April 2011, was used for the removal of radioactive debris outside the buildings, which resulted not only in clear access to the buildings, but also decreased the level of airborne radiation. This system comprised the unmanned construction machinery shown in Figure 1, namely a relay station car, an operation car, a bulldozer, two dump trucks, two backhoes, and seven camera cars. Again, in April 2011, some U.S. military robots were specially modified for the Fukushima Daiichi NPP accident and were used to investigate the severity of the damage and to monitor the airborne radiation.

The networked robot technologies applied to this disaster achieved certain results. The results and trials of the practical applications, however, revealed new issues regarding the management of these robot technologies. First, there was a high usage threshold. With only about 20 skilled operators for unmanned construction machines in all of Japan, it was impossible to place sufficient operators at the Fukushima Daiichi NPP, therefore the lack of skilled operators was one of the most significant problems during the accident. Second, there was a low safety guarantee. The average worker's external exposure value for 3,765 people during March 2011 was high at 13.81 mSv. Even in April 2012, the average value for 5,128 people was 1.07 mSv/month, this is approximately 10 times the normal annual average-exposure radiation level of 1.4 mSv/y for workers at the Fukushima Daiichi NPP. Finally, low efficiency is also a problem with traditional networked robotic systems. For example, the robot group designed for the removal of debris required about six months to remove the outdoor debris at the Fukushima Daiichi NPP.

Fortunately, with the help of cloud robotics, robots can use the cloud computing platform as a common medium of collaboration in undertaking such common and challenging tasks. Therefore, it is important to study robotic physical-task cooperation using cloud robotics.

First, it is necessary to process continuous streamed data in a real-time manner for robots to identify their surrounding environment, such as scene understanding, path planning, object labeling, robot localization, and object affordances [19]. For robots to operate autonomously, they should be able to perceive their environments, manipulate objects, plan paths and interact with our human beings. If a large number of individual robots are connected to the cloud, massive amounts of real-time streamed data from these individual robots needs to be analyzed and processed quickly before they can be recorded in the database and processed offline by the cloud [20]. Data from individual robots or databases can be injected into the data-processing engine as streams and the computing logic running in the engine will continuously process the data and then emit the results. In some scenarios such as robot control, the streamed data from individual robots will have to be processed and fed back in real time. These time-critical tasks require the system to respond sufficiently and rapidly [21].

Second, it is necessary to have access to a large-scale knowledge base for robots to be able to carry out a variety of cooperative tasks [22], [23]. To perform such a task, robots require access to a wide range of information that includes fine details about how to perform the tasks of perception, control, planning, and natural language understanding [24]. For example, when asked to serve a cup of green tea, the robot would have to access to large-scale knowledge about grounding the language symbols into physical entities, the knowledge that green tea can either be in a fridge or on a table, and appropriate knowledge about inferring the detailed plans for grasping and manipulating the green tea cup. We can log the entire body of information that is relevant for achieving robot manipulation tasks including the images interpreted by the robot perception system, pose data, and other sensor and control signal streams, into large scale big-data knowledge bases and annotate them with semantic-indexing and well-organized structures that are generated automatically by the interpreter of the robotic systems. These "episodic memories" enable the robots to answer queries about what they did, how they did it, why they did it, what happened when they did it, what they saw when they did it, and how to cooperate with other robots.

Robotic cooperation on physical tasks aims to enable heterogeneous robots to cooperate with each other effectively, using the cloud to provide both a powerful streamed-data processing capacity to identify the surrounding environment and a large knowledge base to support appropriate decision-making for better robot task cooperation. A robotic cooperation system involves robot clusters, in which heterogeneous individual robots are connected and cooperate to perform complex tasks that no single robot can accomplish. A robot cluster has two major functions. First, it dispatches smaller sub-problems to individual robots in the group and allows them to interact with each other to find solutions to complex problems. Second, and more importantly, it connects to the cloud to enable the real-time processing of continuous streamed data to model the surrounding environment and to access a large-scale knowledge base about appropriate reactions to the environment and identified conditions. By using the remote cloud, the robot cluster can become smarter and improve the cooperation between its robots in carrying out a variety of cooperative tasks.

Figure 5 shows a conceptual diagram of robotic cooperation on physical tasks.

Firstly, the stream-processing layer handles the processing of real-time streamed data, enabling the robots to analyze streamed data continuously as it is captured in a real-time manner and take immediate action, which can be significant in the control of robots. Streamed-data processing applications require a continuous stream of often unstructured data (such as audio, video, or pictures) to be processed. Therefore, data should be continuously analyzed and processed in memory before it is stored on disk. In this layer, Apache Storm [25] can be used as the computation engine, which is a distributed processing system for real-time streams that can

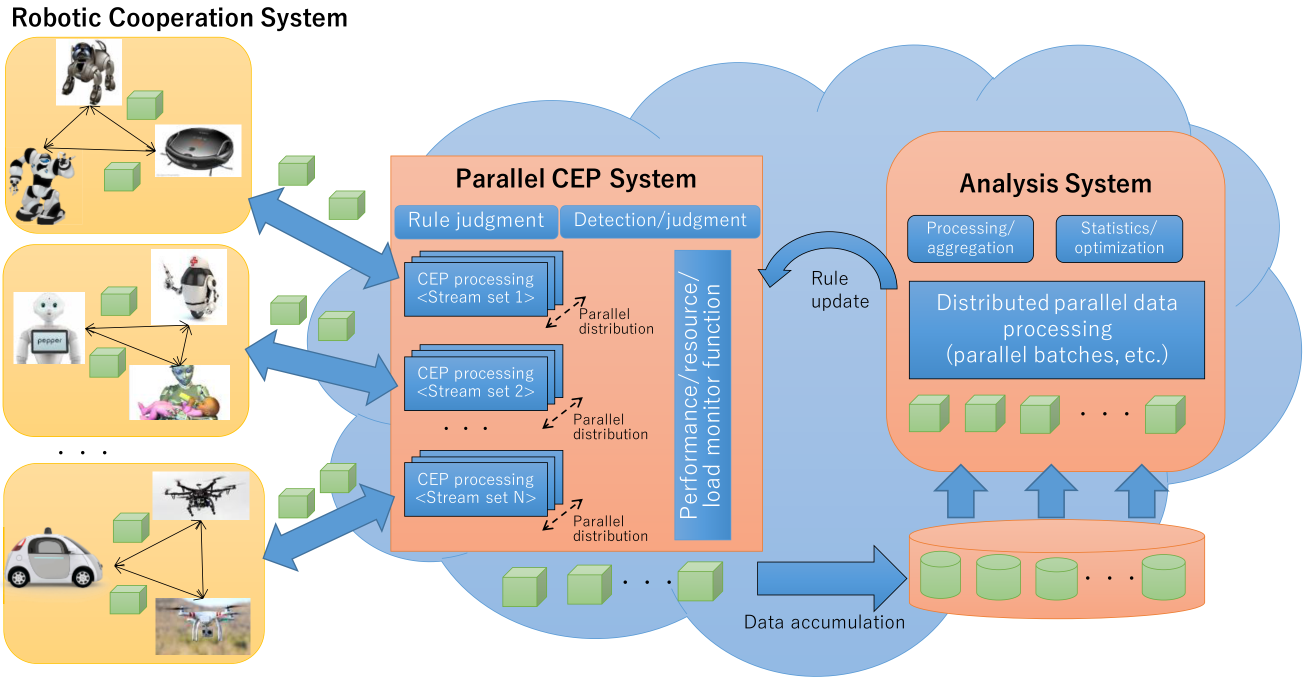


FIGURE 5. **A framework of robotic cooperation on physical-task cooperation.**

process up to one million "tuples" (the data type processed in Storm) per second. It is well suited to processing streamed data from numerous individual robots. In general, a real-time application running in a distributed streamed-data framework can be modeled as a directed graph with processing tasks defining the nodes and streams defining the edges. A stream is a continuous and unbounded sequence of events flowing through the edges of the graph and each such event comprises a chunk of data with a predefined window size. The processing tasks at the nodes continuously consume input streams and produce output streams. To use the real-time stream-processing service in this layer, data from individual robots should be sent to the correct message broker channels, which are connected to the input of a real-time application via a gateway application that connects to the robot's data stream. Then, by subscribing to the channels that connect to the output of the real-time application, results can be fetched in real time. Such a data processing paradigm is well suited to robot control. By deploying multiple instances of the application or increasing the number of computation nodes for the application, the data processing ability can be scaled appropriately.

Secondly, the batch/storage layer (see Figure 5) stores data from the stream-processing middle layer and provides batch processing/data mining services for static data from various distributed databases. First, it splits the file data into manageable chunks or blocks and distributes them across multiple nodes for parallel processing. The data being organized, which may include images, videos, human-language data, and even physical-interaction data, can take several forms, such as column-oriented data, semi-structured XML data, relational-database items, document-database items, key/value pairs, or a raw format. This means that several techniques and tools need to be employed for the effective retrieval and organization of the data. Second, the batch/storage layer extracts knowledge by advanced analysis using mining techniques. This involves constructing analytical models, finding patterns and associations, and performing classifications and predictions. With potentially large volumes of stored data, scalable distributed parallel processing is used. The batch layer uses a batch-processing model because huge amounts of data have already been collected before the data processing is undertaken. It targets the production of accurate results by performing statistical analyses of tens of terabytes or even petabytes of the stored data. In contrast to the stream-processing layer, the batch/storage layer would usually have a large latency, in the order of minutes to hours. Known as an effective solution for the batch/storage layer, Davinci [26] makes use of the Hadoop Map/Reduce framework for the batch processing of visual information and sensor data to learn from many types of learning signals and from a variety of multimodal data. Rapyuta [27] is another open-source cloud-robotics framework that allows diverse robots to offload their heavy computations to secure computing environments in the cloud computing platform.

With the help of cloud robotics, robots can cooperate in more complex tasks than can be undertaken by traditional multi-robot systems. One of the challenging issues is to reduce the delay incurred during big-data processing. latency is becoming a more critical concern for robotic cooperation in cloud robotics. For example, a self-driving car in high speed must make snap decisions. Even slight delays in updating weather and road conditions could mean longer travel times or dangerous errors (such as collision between two autonomous cars). The new 5G network [28] can be considered as a potential solution to delay control by reducing latency from 40-60ms to 1-10ms. Especially, low latency and low mobility interruption gab together with reliability are of the highest importance in the 5G agreed scenarios for cloud robotics, e.g. in order to connect cars, drones and other mobile robots to the cloud. However, there remain many issues related to the network architecture and network access protocols. Even with current 5G methods, cellular-network bandwidth may never be low-cost, making bandwidth utilization and optimization an important concern.

Robot mobility is another challenge for robotic cooperation on physical tasks. Individual robots in cloud robotics establish connections with others, even using fixed nodes in a wired network, as they move about. As an individual robot moves from one environment to another, it may connect to a new cloud proxy or join a new cluster of robots. Therefore, a new prediction strategy and adaptation policy should be developed to ensure that they can still benefit from computation and storage-power enhancement seamlessly. This might involve dynamic application offloading or virtual machine (VM) migration techniques. Moreover, because of the mobility of individual robots, the communication network associated with a robot cluster will change over time and space, bringing some challenging issues to QoS guarantees. Therefore, cloud robotics should consider a dynamic robotic-cooperation strategy when aiming to improve the quality of robotic cooperation.

Finally, context awareness is another difficult challenge for robotic cooperation on physical tasks. Context awareness in an application model refers to its ability of discovering and reacting to changes in its physical environment [29]–[31]. For example, it is critical for service robots to be aware of the dynamic characteristics of information about a person's activities, such as survivors' movements in a disaster site. To achieve a ubiquitous robotized environment, the capability of being sensitive to changes in context is required to adapt the system's reaction to the movement intentions of the people involved. Context awareness has been studied in the domain of mobility systems, usually with respect to personal devices such as smartphones and PDAs, focusing on how they adapt to changes in network connections [29]. The application of context awareness in robotics will require the integration of many more information sources than the ones mentioned in these examples. However, the particular case of sharing context information between distinct and heterogeneous networks without having to redevelop the entire system from scratch remains a challenge.

C. ROBOTIC COMPUTATION COOPERATION

Resource-intensive applications, such as those involving streamed video and sensors mounted on robots, produce continuous streams of data about the robot's environment. They are increasingly being deployed in robotic computing. However, the built-in processing capacity of individual robots will not be able to keep pace with the increasing need for resource-intensive processing. Resource limitation means that individual robots will not be able to support resource-intensive features such as artificial vision and object tracking. CPU performance, battery life and memory are some of the non-scalable resources in individual robots. Motivated by this problem, cloud-robotic cooperation on computation addresses the limitations of individual robots both by resource sharing among robots and by computation offloading to the resource-abundant cloud. Computation-intensive applications can be partitioned, with components migrated to the robot cluster or to the remote cloud in a cooperative manner. After the intensive computations are executed either by robot–robot or by robot–cloud cooperation, the processed results are sent back to the robots for incorporation into the overall computation.

Consider the following scenario, a set of rescue robots enter a disaster area, aiming to observe the condition and location of the survivors, the interior of the rubble, and any potential dangers. This will involve video streams captured by cameras. The core function in this application is object recognition from sequences of video frames. A rescue robot's onboard processor would usually execute this function periodically as it moves about, aiming to identify features of the environment in real time. The execution time for such a recognition function on mainstream hardware (1.7 GHz, 4-Core CPU, and 2 GB RAM) has been investigated [32]. It requires at least 60 seconds to process one $1{,}000 \times 800$-pixel frame in the video stream. Higher resolution implies a longer time and more energy consumption. However, the rescue robot could offload this task, by sending the streamed data to the cloud, where a cloud-based application performs object recognition, and receiving back the processed results. In this way, the energy consumption and the execution time could be reduced.

In reality, however, a rescue robot in a disaster area is likely to encounter bandwidth fluctuations or even disconnections. This may involve a long wait for the results to become available. In such cases, a better solution would be to execute the application within a robot cluster by cooperating with the computation resources of neighboring robots. In fact, by enabling both robot–robot and robot–cloud cooperation, the robotic system can adapt to changes in its surrounding environment such as network bandwidth fluctuations, enabling complex robotic computations to be more effective, saving execution time and energy consumption.

Robot cooperation in computation involves not only robot–robot cooperation but also robot–cloud cooperation, and aims to enable computation-intensive or energy-intensive workloads via the distributed execution of robot-cloud applications. This is achieved by migrating a portion of the complex application state from the individual robots to remote cloud-based computation resources. Robot–robot cooperation can be supported by an ad-hoc network formed by a cluster of neighboring robots that work cooperatively to achieve application offloading. However, robot–cloud cooperation is driven by cloud clones and remote execution engines, where resources are elastic and available on demand.

Figure 6 shows the components of an computation offloading system, which is composed of a client module running on the individual robots, and a server module running in the ad-hoc network formed by a cluster of heterogeneous robots nearby that either share their computation resources among the cluster to quickly solve computationally hard problems, or communicate with the remote infrastructure-base cloud, which enables the robots to benefit from the powerful storage, computation, and communications resources of modern data centers. The client module has three major functions. First, it monitors and predicts the network and processing performance of the individual mobile robots. Second, it tracks and predicts the execution requirements for resource-intensive applications in terms of execution time on both the individual robot and the cloud and input/output data requirements. Third, by using this information, the client module chooses some portions of the resource-intensive applications to execute in the cloud to minimize the total energy consumption and execution time. Alternatively, the server module executes these offloaded portions immediately in the cloud after receiving them and returns the processed results back to the client module, after which the resource-intensive application can be resumed on the individual robots.

To enable distributed execution of a resource-intensive application, the computation offloading framework must determine the efficient partition of the given application for scheduling on individual robots and cloud servers. This is the job of the task-offloading decision engine, which may be present either in the individual robot or in a predefined server in remote cloud. The task-offloading decision engine must effectively identify the most energy or computation intensive tasks in the given application to migrate some portions of the running applications. Using information about the running application and the available resources in the cloud-robotic system, it selects program components to be offloaded for remote execution in ad-hoc virtual cloud and remote cloud. The decision about where to place the execution should be made by considering the relative amounts of computation and communication resources required by these portions of the running application. Intuitively, a task involving little communication combined with a large amount of computation should point to remote execution, whereas much communication combined with little computation should indicate local execution of the task.

However, when considering the delay constraints and the extra costs of data transmission and remote computation, it is not trivial to make optimized decisions. First, the computational offloading strategy should consider various factors such as offloading granularity, robot capacities, network delays, local and remote execution overheads, and the delay deadline for task completion. Particular consideration should be given to the amounts of data that must be exchanged and bandwidth limitations between individual robots and servers in the remote cloud. Second, given the presence and dynamics of cloud resources and computation requirements, the offloading strategy should also dynamically consider the relative advantages of executing the task within the group of

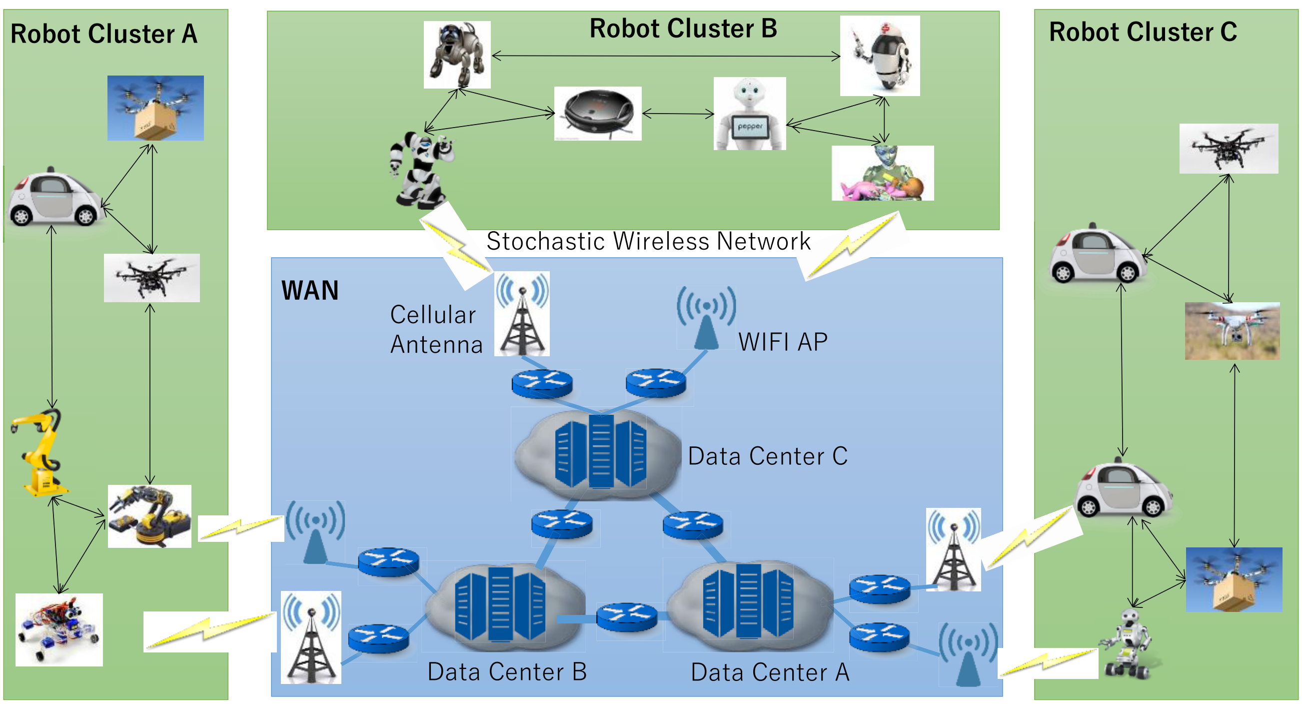


FIGURE 6. A framework of robotic cooperation on computation cooperation.

networked robots or to the remote cloud. In general, task component offloading can be either static or dynamic. In static computation offloading, the programmers predetermine the task components that can be offloaded to the ad-hoc virtual cloud or the remote cloud. However, this solution may not be very effective, as many issues may affect the performance of computation offloading. Alternatively, in dynamic computation offloading, the execution location of the task components is not predetermined, and the offloading decisions are intelligently made by analyzing contextual information, such as, robotic computation and communication resources, bandwidth, energy, latency, and available resources in the cloud.

Computation offloading first requires the data to be transmitted and processed before receiving back the results, which can be time consuming. Although several approaches [33]–[36] can facilitate and promote mobile cloud-based computational offloading, research and development in this domain remains a high priority. There are two main issues, which we can identify as follows.

First, the partition of robotic applications (or task graphs) is critical for computation offloading in cloud robotics. However, identifying resource-intensive task components in robotic task graphs is a challenging task because there are few hard rules that indicate a task component's intensity. For instance, a task component might be resource-intensive (in terms of computation time) for a robot with low processing power, but not for a robot with high computation capacity. Even if intensity is defined in terms of computational complexity (the number of instructions per unit of processing data), partitioning robotic applications remains an issue. For example, static application partitioning and decisions regarding the task components' execution location would not be a foolproof solution and might fail in a number of scenarios. Even though context-aware dynamic application partitioning has an edge over static application partitioning, it still requires timely repartitioning of task graphs to accommodate changes caused by the inconsistently available cloud-based resources and mobile robots' dynamic environments.

Second, computation offloading in cloud robotics always trades off communication costs for computation gain. Previous systems [37]–[40] usually assume adequate mobile cloud-based computation resources and stable network connectivity. However, in cloud robotics environments, an individual mobile robot may experience poor or even intermittent connectivity, while cloud services may be busy or even temporarily unavailable. This would lead to high data transmission latency and low energy efficiency for computation offloading. In such cases, it might not be beneficial for the individual robots to offload computational tasks to the remote cloud. Besides, achieving efficient computation offloading coordination among individual robots is still a challenging task. If too many individual robots choose to simultaneously offload computational tasks to the cloud via wireless access, this might generate severe mutual interference, which would reduce the data rates for computation data transmission [41] and therefore increase the latency of data transfer. Here, the communication cost may be higher and the computation gain may be lower. Moreover, the network and execution predictions may be inaccurate, causing the overall performance of the cloud robotic systems to be degraded.

V. APPLICATIONS AND CHALLENGES

A. APPLICATIONS

After the Great East Japan earthquake disaster, we started a cloud-robotic cooperation project sponsored by the Fukushima prefecture. The project aims to build a cloud-robotic cooperation system to help future disaster management, as explained in Section 2, and the challenging demolition task at the Fukushima Daiichi NPP.

Figure 7 shows the framework of our cloud-robotic cooperation system. First, there are cloud-based data centers, with all robots being able to connect to the cloud. This enables the robots to benefit from the rich storage, computation, and communications resources of the cloud data centers. Second, a group of robots can communicate with each other, learn from each other effectively, and ask for help in performing a required operation with the support of the cloud. A group of robots working as cooperative agents with the help of the cloud can significantly improve their overall efficiency and reduce the burden on the human controllers.

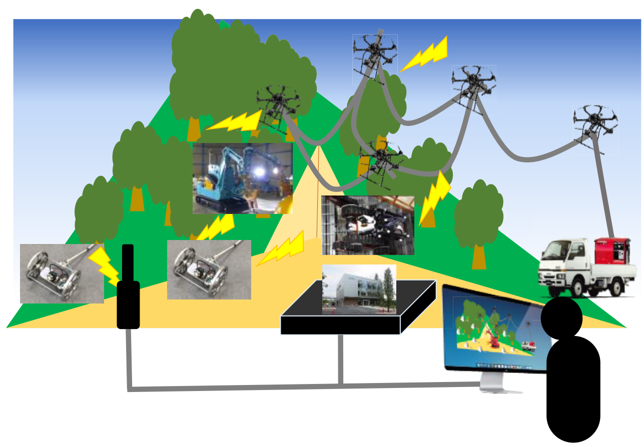


FIGURE 7. The framework of our cloud robotic cooperation system.

In our cloud-robotic cooperation system, a robot can call for help from other neighboring robots in the network. The robot can send its navigation direction, exact GPS location, and other relevant information (such as an unusual obstacle in the way or a useful detour) that it has learned during its travels. When new robots arrive, they can collaborate with each other in the rescue operation, which will then be much more effective. With our cooperative network, robots can avoid redundant operations and can make intelligent plans. After a robot has completed its search of an area, it can mark that area as covered and notify other neighboring robots not to waste time searching the area. Moreover, if a device fails or power/battery problems occur, it can notify other neighboring robots to take its place. We now summarize the characteristics of our cloud-robotic cooperation system in terms of the aspects of robotic cooperation discussed in Section 3.

1) APPLYING ROBOTIC KNOWLEDGE-SHARING

By using robotic knowledge-sharing, our cloud-robotic cooperation system can learn from individual-robot experiences, leading to greatly enhanced robotic capabilities. The cloud provides a common medium for the cooperative robots to share gathered information and to learn new knowledge and skills from each other. The cloud can host a knowledge base or a library of behaviors and skills that map to different environmental complexities and task requirements. One representative application for robotic knowledge-sharing is determining the optimal way to grasp an object in a disaster site, such as debris or an injured survivor trapped under a building. In general, if the full 3D model of the object is precisely known, then one of the various standard methods can be used to synthesize the object grasping procedure. However, if the object is not precisely known or completely unknown, the problem becomes much more challenging, and will need much data preprocessing and a large amount of computation. Recently, data-driven or information-based object grasping methods [22] have been developed to enable robotic grasping for any object and any robot hand. But all these methods require access to large databases. Cloud-based resources can facilitate incremental learning of appropriate object grasping strategies, by matching sensor data against 3D CAD models stored in the cloud in advance. Examples of sensor data include 3D point clouds, 3D features, and 2D image features, in addition to demonstrations [11]. With the support of cloud-based robotic knowledge-sharing, a mechanical hand can send featured data obtained from a small number of sensors to the cloud using a specific data format. The cloud processes the featured data and performs model matching using a knowledge base stored in the cloud, and returns back data for a set of candidate objects, each with grasping options. The robot itself compares the received 3D CAD models from the cloud with the detected point cloud to refine its identification and to perform pose estimation, and finally selects an appropriate grasping solution. Moreover, model knowledge with respect to new objects learned by various robots in disaster sites can be added to the knowledge base in the cloud and become available for future reference by other robots in similar cases of similar need. Figure 8 is an example of debris grasping using robotic knowledge-sharing, where the grasping options were acquired from different robots in domains other than disaster management.

2) APPLYING ROBOTIC PHYSICAL-TASK COOPERATION

The complexity of many environments and missions in disaster sites may require teams of possibly heterogeneous robots that can collaborate with each other. Fortunately, robotic physical-task cooperation can mimic the abilities of the human brain by providing rapid-response streamed-data processing and knowledge about task execution to provide high-quality heterogeneous-robot cooperation. A representative application for physical-task cooperation is robot navigation in disaster sites [42]. In a disaster scenario, the environment after widespread damage is completely unknown and the task of robot navigation is to determine the robot's own position with respect to a certain reference point and then plan a path to reach a desired location [43]. There has been extensive research, including both map-based approaches and map-less approaches [44], [45]. However, because of limited onboard processing and storage resources, these methods usually suffer from reliability problems. The map-less approach can involve large-scale computing tasks (processing observations from perception sensors) and the map-based approach can involve large-scale storage issues (the process of building and searching the map for navigation routes). Robotic physical-task cooperation provides a very promising solution for future cloud-enabled navigation that overcomes these challenges. It can not only provide sufficient storage platform to store the large-scale knowledge for mapping, but also provide enough computing capacity to quickly facilitate the searching and building of the map by processing streamed big data in real time. Figure 9 shows an example of navigation in a disaster site using robotic physical-task cooperation. The robots in the disaster site not only send streamed camera data about their environment continuously to the cloud for real-time 3D map construction [46], but

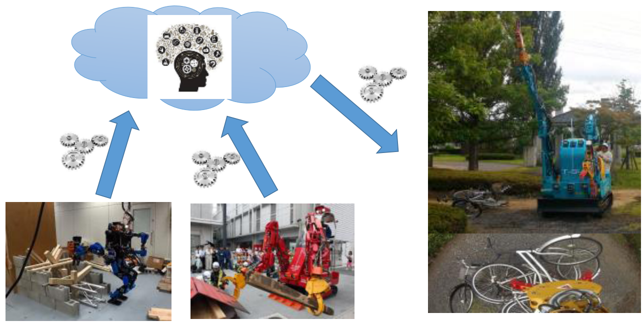


FIGURE 8. Debris grasping using robotic knowledge sharing.

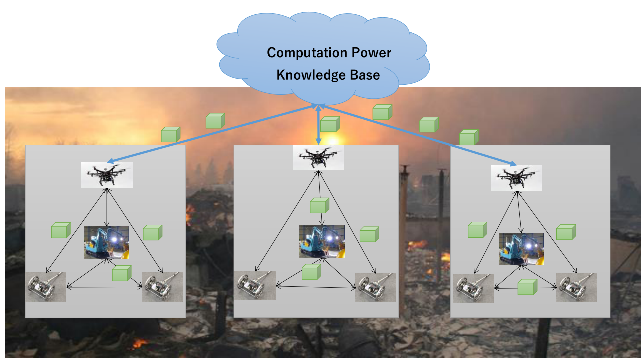


FIGURE 9. Navigation in a disaster site using robotic physical-task cooperation.

also share with each other their exact GPS locations and navigation directions [47], together with other relevant information that they have learned along the way, such as unusual obstacles or useful detours.

3) APPLYING ROBOTIC COMPUTATION COOPERATION

The basic idea of robotic computation cooperation is to enable individual robots to leverage the computation and energy in robot clusters and cloud servers cooperatively to execute computational tasks that require heavy use of computing and/or network resources. An individual robot in our robotic cooperation system only has to support its cameras, other sensors, and actuators with basic processing power. Despite its limited processing power, it can be involved in computationally intensive actions. A representative application for robotic computation cooperation is simultaneous localization and mapping (SLAM) in disaster sites. This is a technique enabling an autonomous vehicle or a mobile robot to build a map of its environment without any priori knowledge, and to localize itself simultaneously in the completely unknown environment [48]. Although SLAM-based algorithms are becoming more and more accurate, the superiority of the current algorithms for large scale maps cannot make up for the limited storage and computational resources on board individual robots. Therefore, the formation of large-scale maps often required excessive time before the advent of cloud robotics, well outside the needs of real applications. The emergence of cloud-based technologies, together with robotic computation cooperation, can remove the bottleneck caused by limited onboard storage and computing equipment by offloading tasks such as map filtering and fusion for state estimation to the cloud. For example, offloading the visual SLAM (VSLAM) processing to systems running in a cloud-based deployment of the ROS has been proposed as a method for managing increasing processing constraints [49]. The FastSLAM algorithm has been implemented in Map/Reduce and experimental results show significant execution-time gains when building a map of a large area [26]. As demonstrated in [50], the cloud can substantially improve the processing speed of SLAM. Figure 10 is an example of SLAM in a disaster site using robotic computation cooperation. Tasks such as *TakePicture* and *Segmentation* are executed in an ad-hoc network formed by a group of neighboring robots that work cooperatively. Other tasks, including *FeatureDetection*, *Matching*, *ObjectRotation*, *3DReconstruction*, *PositionUnderstand*, and *MapCreation*, are offloaded to the remote cloud's powerful computation resources. Finally, the received map can be displayed and used by various robots in the disaster site, having been produced with low latency and high energy efficiency.

B. CHALLENGES

In this subsection, we discuss some specific research challenges to robotic cooperation in cloud robotics that should be addressed.

1) QoS ISSUES

The QoS for robotic cooperation in cloud robotics refers to its performance in satisfying its users' preferences [51], [52]. In particular, two QoS factors, namely latency and energy consumption, are discussed below.

a: LATENCY ISSUES

Cloud robotics needs to collect and analyze data in real time and make quick decisions. Such systems are sensitive to latency [53]. In cloud robotics, latency is defined as the time involved in acquiring the necessary knowledge or offloading the computational tasks and retrieving the results from the cloud. It involves many factors, such as processing time on individual robots, processing time in the cloud, the input/output data sizes, data transport time, and network latency. On the one hand, to reduce latency, a task application must be aware of the task's computation time on both the robot side and the cloud side for latency estimation. The challenge here is that estimating the computation time of task components in an application is a complex problem. First, the robots have heterogeneous hardware specifications, thus there can be no predefined computation time for task components in an application. Second, the type of coding and data input size used for instructions are much important factors that can affect a task component's computation time. This may be partially resolved by profiling robot and cloud-based component executions, however, the overhead of

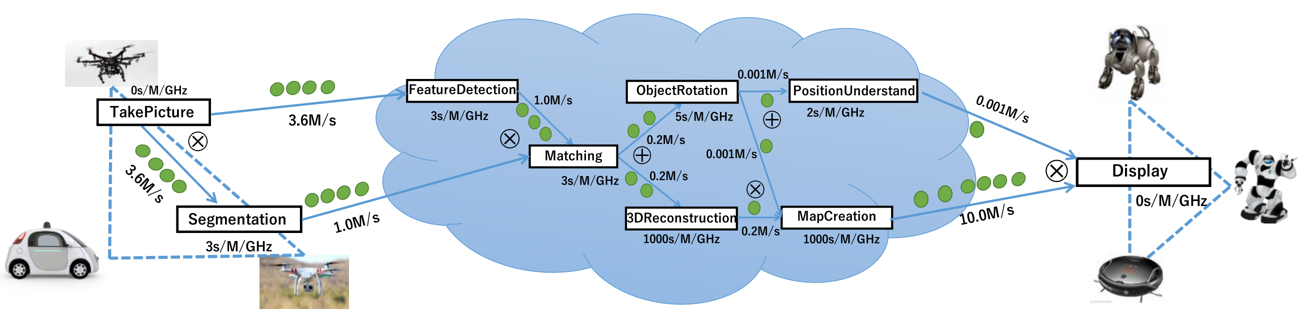


FIGURE 10. SLAM in a disaster site using robotic computation cooperation.

profiling every execution of a task component against variable input-data size should be considered.

On the other hand, to reduce interaction latency, proposals in [33]–[36] aim to create a proximate cloud to access nearby remote resources. However, security and trust matters are the most important factors hampering the proposed solution. Further research is therefore needed to develop systems that are trustworthy and offer a rapid response. In particular, latency-aware global resource-scheduling approaches, such as algorithms [54], [55] for service placement, data placement, task placement, and their joint optimization, are needed to allocate tasks to cloud-robotic systems optimally for latency reduction. Besides, latency-aware communication protocols should also be developed to support communication in cloud robotics that offers further latency reduction.

b: ENERGY EFFICIENCY ISSUES

Battery is one of the most precious resource for mobile robots. The key of energy-efficiency cloud robotics is to seek better tradeoff between the transmission energy consumption and computation energy consumption. To make robotic applications energy efficient, the applications must be aware of the amount of energy required for computation offloading to the remote cloud and robot-based application execution. Otherwise, bad offloading decisions could overwhelm the mobile robots' energy capacity. However, monitoring the energy consumption of task components in an application is a challenging task. First, the amount of energy consumed by an task component depends on the robotic model (its hardware specifications). For example, different amounts of energy will be consumed by two robots with different types of processors, such as quad-core versus single-core. Second, a task component's energy consumption can vary depending on the CPU frequency and utilization level. The problem is that robots do not provide low-level energy information about their communications and computations.

Besides, energy-efficient communications are critical when applying the cloud to extend the capabilities of individual robots. The stochastic characteristics of wireless networks may lead to unpredictable energy consumption for communications between individual robots and the remote cloud. For example, the network capacity and availability, including the signal strength and bandwidth at access points, can vary from place to place. Moreover, there are uplink and downlink bandwidth fluctuations caused by, for example, mobility issues, building shields, flash crowds, and the weather. Measurement studies [56] have shown that the energy consumption for transmitting a fixed amount of data is inversely proportional to the available bandwidth. This implies that frequent transmission during bad connectivity may consume excessive energy, making the computation offloading unattractive. A recent practical solution called eTime [57] adaptively and aggressively uses periods of good connectivity to prefetch frequently used data while deferring delay-tolerant data during bad connectivity. However, it is a challenge to estimate network connectivity both quickly accurately. In addition, more energy-efficient communication protocols such as the HpFP protocols [59] are needed to support mobile communications with high energy efficiency for cloud robotics. Finally, edge computing [60] which has been considered as a potential solution to address the concerns of energy-efficiency and latency by processing the data at the edge of the network, needs further research for cloud robotics.

2) COMMUNICATION ISSUES

Cloud robotics should ensure that there is fast and continuous Internet connectivity between robots and the remote cloud. The individual robots are always linked to the remote cloud from any time and place for task execution [58]. However, when compared with wired networks, wireless networks are characterized by intermittent, low-bandwidth, and less-reliable transmission. Establishing and maintaining seamless sessions between the remote cloud and individual robots using wireless networks are critical to fully realizing the power of cloud robotics. Intermittent connectivity leads to several challenges such as the excessive consumption of limited robotic resources, dismissal of always-on connectivity, and disproportionate delaying of application execution that can sharply degrade the overall performance of robotic cooperation. Therefore, during the development of the cloud-robotic cooperation models, much efforts must be made to address these challenges.

First, communication issues in extremely bad environments such as high radiation levels and signal black-spot areas need to be investigated [61]–[63]. For example, the radiation tolerance of the electronic components in Quince robots were checked by means of gamma-ray irradiation tests. It was found that the high-power wireless communication devices previously used in Quince robots could not work in the Fukushima Daiichi NPP [64]. In addition, a wireless relay-communication technology that uses other robots to control robots beyond the line of sight has been developed [65]. This aims to maintain the control link continuously when the relay route changes as the robots move about and to support stable operation of remotely controlled robots in a poor radio-propagation environment. The robots cooperate with each other to maintain a continuous wireless control link, even in the presence of obstacles such as thick walls, buildings, trees, or mountains. Second, the communications associated with real-time big-data processing also raises challenges. HpFP [59] was developed for high-quality streamed-data transfer. It outperforms UDP and TCP protocols in terms of latency and tolerance to packet loss. Finally, existing approaches [66]–[70] are further compared and summarized in Table 2.

3) SAFETY ISSUES

Safety is essential for robots, given the mission-critical deployment of many cloud-robotic applications such as air and ground transportation systems, disaster monitoring and warning systems, and medical and healthcare systems [71]. It is therefore important to ensure the overall stability of

TABLE 2. Comparisons of different communication models.

Literature	Network model	Real-time transmission	Reliability	Cost efficiency	Coverage	Rapid deployment	Self-organization	Challenges
[65], [66]	Mobile ad-hoc networks	Good	Fair	Good	Fair	Good	Good	End-to-End communication
[67], [68]	Delay tolerance networks	Poor	Fair	Good	Good	Fair	Fair	Long delay
[69], [70]	Movable base stations	Good	Good	Poor	Good	Poor	Fair	Distribution and adjustment is difficult

physical systems to avoid potential dangers such as imminent collision. ISO 60601 defines safety as the avoidance of hazards to the physical environment during the operation of a medical device under normal or single-fault conditions [72]. We believe that this definition of safety can also be applied to nonmedical domains such as cloud robotics by broadening the scope of the hazards considered, including radiation leaks, faulty operation of the computation unit, thermal effects, software failures, biocompatibility issues, electrical hazards, and mechanical hazards. However, because there are various fault sources in the physical, networking, sensing, computing, and actuation domains that can make systems behave anomalously, it is challenging to achieve this goal in cloud robotics. Many uncertainties exist in their environment and physical systems. Any one of a variety of types of failure can occur at any place and at any time in cloud-robotic systems [73].

More work is needed on interaction safety [72]. There are two main cases. The first is when the cyber-physical interaction between the computation units in two different individual robots may affect either one's operation in hazardous ways. Second, the cyber-physical interaction between the computation units and the physical environment may have harmful effects on the physical environment. The nature of the physical environment may hinder the operation of the computation unit in the second case. For example, tissue growth around implanted sensors could reduce communication and sensing capabilities.

Finally, from the social perspective of the robotic community, work is needed to enhance the safety of cloud robotics. First, laws, regulations, and social structures such as insurance should be in place [61]. Second, high-quality simulators are needed and robotic test fields should be constructed. Third, robotic systems should be continuously tested, and exercised at real scenarios (such as disaster sites) and in simulated mock-ups. Fourth, there should be active user communities to enable information exchange and user collaboration. In other words, sufficiently robust technology must be adopted for robotic systems developed for use at real scenarios (such as disaster sites). Otherwise, it would be unclear if the cloud robotic systems could work well in real scenarios, for example. If a robot becomes inoperable on an access path, the robot itself would become an obstacle to other robots [1].

4) HETEROGENEITY

Heterogeneity is a major concern for robotic service provisioning. There are main areas of heterogeneity in cloud robotics.

First, heterogeneity among individual robots arises from technological variation in terms of OS, software, hardware architecture, features, platform, and the communication medium. Bridging this heterogeneity to provide a homogenous processing solution for distributed applications remains a very challenging issue for cloud robotic systems. For example, interoperability is one of the most significant technological and standards challenges [74]. Existing hardware, platform, operating system, API, and feature heterogeneities among the multitude of robot types necessitate standardization.

Second, heterogeneity in cloud systems arises because there are numerous cloud providers in the market, each providing a variety of cloud services using their own customary policies. This leads to cloud heterogeneity as these vendors develop their respective infrastructures, platforms, and APIs, which leads to interoperability and portability challenges.

Third, there is heterogeneity in the wireless networks. In cloud robotics, wireless networking is the major communication medium, which can be cellular, WiMAX, GPRS, WLAN, CDMA2000, WCDMA, or satellite-based. Heterogeneous networks operate with different connection protocols and technologies and network traffic is time-varying and uneven [75]. This variation affects the mobility, augmentation, and usability of individual robots. It raises the problem of managing the wireless connections in addition to addressing cloud-robotic needs such as being energy-efficient on the robot side, always being connected, and achieving scalability of on-demand wireless connectivity [76].

Therefore, during the development of cloud-robotic cooperation systems, work on middleware such as OpenRTM and ROS must be undertaken to address these challenges.

5) BIG-DATA PROCESSING

Because cloud cooperation services usually deal with large-scale data sensed by individual robots, the overhead of data transmission and processing is a key challenging issue.

First, a variety of types of sensor on individual robots will be widely deployed and robots in the same region may therefore generate duplicated data. An in-network processing mechanism is needed for network components to aggregate data from downstream nodes and filter duplicated and noisy data.

Second, unstructured or semi-structured information sensed from different sources may be correlated. Seeking efficient algorithms for in-network processing and compression, such as compressive sensing, to reduce the cost of data transmission is a popular recent research topic [76].

Cloud robotics provides massive storage and computing resources for storing and processing large volumes of data, which is hosted in large data centers and accessed through the cloud on demand. However, analyzing, indexing, and querying large volumes of spatial and high-dimensional big data, when the data may be unstructured or semi-structured, is difficult to perform at high speed. Moreover, software solutions and algorithms for high-performance distributed big-data computing on a large scale is a nontrivial challenge. Cloud robotics demands highly distributed big-data processing techniques with low communication and computing requirements. In particular, processing large amounts of real-time data, while aiming to make quick decisions and transmit the relevant instructions back to the sensing robots in a timely manner, is now a major challenge in both research and application development [77].

6) PRIVACY AND SECURITY ISSUES

Because individual robots upload their personal private data for storage and processing by a remote cloud, there is a major concern about privacy and the leakage of private information [78]. A particular privacy issue for mobile robots is the leakage of their private location information in location-based services. To solve this problem, a method called ''location cloaking'' makes users' location data slightly imprecise before being uploaded to the remote cloud [79]. Sometimes, however, imprecise data could not provide satisfactory or relevant results for some certain applications. Therefore, location cloaking needs to be adaptively tuned to balance the trade-off between result accuracy and privacy for cloud robotics.

Security issues arise from every aspect of cloud robotics. This includes security for individual robots (e.g., eliminating the threat of worms and viruses), security in the cloud data-center nodes (e.g., preventing unauthorized access to personal data stored in the remote cloud), and security for data transmission over networks (e.g., encrypting communication protocols). First, security mechanisms should be lightweight, without involving much computation power and energy consumption by the robots. Second, cloud clones should be trusted. The robot should be able to check the identity of the cloud clone based on trust measurements (i.e., reputation-based trust [80]) or to identify a trusted cloud clone by itself (i.e., trust establishment [81]). Third, storage and computation services provided by the cloud computing platform must be trusted. Finally, there is an urgent demand for technologies that endeavor to enforce security and privacy in data transmission when moving crowdsourced data to cloud data centers [82]. Blockchains [83]–[85] which has been considered as a potential solution to address the concerns of vulnerabilities, potential threats, and attacks, needs further research for cloud robotics.

VI. CONCLUSIONS

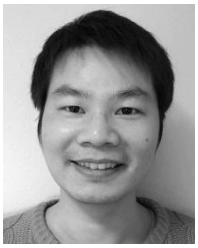
Cloud robotics is a frontier interdisciplinary area that combines robotics and computer science to investigate the huge expansion in the capability of robotic systems. Computer science's role is to make robots smarter by introducing cloud computing, big-data science, and machine learning in the near future. Robotic cooperation will become a major trend and even more popular in cloud-robotics research as increasingly complex tasks are undertaken by a group of robots. In this paper, we have proposed three novel robotic-cooperation frameworks for cloud robotics to enable improved robotic cooperation. This will open new horizons in the domain of robotics that we believe will lead to wide-ranging research and development initiatives. The challenges emerging from these three robotic cooperation frameworks were discussed. We introduced our ongoing cloud-robotic cooperation system using the proposed robotic cooperation frameworks in terms of disaster management. Some of the challenges for robotic cooperation in cloud robotics were identified and discussed. These include QoS, communication, safety, heterogeneity, big-data processing, and privacy and security issues. In future work, we will continue to develop our cloud-robotic cooperation system, aiming to improve robotic cooperation for disaster management such as the challenging demolition task at the Fukushima Daiichi NPP. We also aim to address the various identified challenges. We plan to test our cloud-robotic system in the Fukushima robot test field being built in in the Fukushima prefecture.

REFERENCES

[1] H. Osumi, ''Application of robot technologies to the disaster sites,'' in *Report of JSME Research Committee on the Great East Japan Earthquake Disaster*. Tokyo, Japan: Jpn. Soc. Mech. Eng., Feb. 2014, pp. 58–73.
[2] S. Kawatsuma, M. Fukushima, and T. Okada, ''Emergency response by robots to Fukushima-Daiichi accident: Summary and lessons learned,'' *Ind. Robot, Int. J.*, vol. 39, no. 5, pp. 428–435, 2012.
[3] F. Matsuno, N. Sato, K. Kon, H. Igarashi, T. Kimura, and R. Murphy, ''Utilization of robot systems in disaster sites of the Great Eastern Japan earthquake,'' in *Proc. Field Service Robot. (FSR)*, 2012, pp. 1–17.
[4] N. Michael *et al.*, ''Collaborative mapping of an earthquake-damaged building via ground and aerial robots,'' *J. Field Robot.*, vol. 29, no. 5, pp. 832–841, 2012.
[5] G. Hu, W. P. Tay, and Y. Wen, ''Cloud robotics: Architecture, challenges and applications,'' *IEEE Netw.*, vol. 26, no. 3, pp. 21–28, May/Jun. 2012.
[6] J. J. Kuffner, ''Cloud-enabled robots,'' in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, Nashville, TN, USA, Nov. 2010, pp. 176–181.
[7] M. Waibel *et al.*, ''RoboEarth,'' *IEEE Robot. Automat. Mag.*, vol. 18, no. 2, pp. 69–82, Jun. 2011.
[8] A. Saxena, A. Jain, O. Sener, A. Jami, D. K. Misra, and H. S. Koppula. (2014). ''RoboBrain: Large-scale knowledge engine for robots.'' [Online]. Available: https://arxiv.org/abs/1412.0691
[9] (2018). *Google Self-Driving Cars*. [Online]. Available: https://waymo.com/
[10] (2018). *Amazon Robotics*. [Online]. Available: https://www.amazonrobotics.com/#/
[11] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, ''A survey of research on cloud robotics and automation,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 398–409, Apr. 2015.

[12] J. Wan, S. Tang, H. Yan, D. Li, S. Wang, and A. Vasilakos, "Cloud robotics: Current status and open issues," *IEEE Access*, vol. 4, pp. 2797–2807, Jun. 2016.
[13] P. P. Ray, "Internet of robotic things: Concept, technologies, and challenges," *IEEE Access*, vol. 4, pp. 9489–9500, 2016.
[14] L. E. Parker, "Current research in multirobot systems," *Artif. Life Robot.*, vol. 7, nos. 1–2, pp. 1–5, Mar. 2003.
[15] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 399, pp. 1–18, 2013.
[16] A. Gautam and S. Mohan, "A review of research in multi-robot systems," in *Proc. IEEE 7th Int. Conf. Ind. Inf. Syst. (ICIIS)*, Chennai, India, Aug. 2012, pp. 1–5.
[17] L. Riazuelo *et al.*, "RoboEarth semantic mapping: A cloud enabled knowledge-based approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 432–443, Apr. 2015.
[18] M. Chen, Y. Qian, Y. Hao, Y. Li, and J. Song, "Data-driven computing and caching in 5G networks: Architecture and delay analysis," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 70–75, Feb. 2018.
[19] H. He, S. Kamburugamuve, G. C. Fox, and W. Zhao, "Cloud based real-time multi-robot collision avoidance for swarm robotics," *Int. J. Grid Distrib. Comput.*, vol. 9, no. 6, pp. 339–358, Sep. 2016.
[20] G. Mohanarajah, V. Usenko, M. Singh, R. D'Andrea, and M. Waibel, "Cloud-based collaborative 3D mapping in real-time with low-cost robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 423–431, Apr. 2015.
[21] C. E. Agüero *et al.*, "Inside the virtual robotics challenge: Simulating real-time robotic disaster response," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 494–506, Apr. 2015.
[22] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the Google object recognition engine," in *Proc. IEEE Int. Conf. Robot. Automat.*, Karlsruhe, Germany, May 2013, pp. 4263–4270.
[23] M. Tenorth, K. Kamei, S. Satake, T. Miyashita, and N. Hagita, "Building knowledge-enabled cloud robotics applications using the ubiquitous network robot platform," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 2013, pp. 5716–5721.
[24] M. Beetz *et al.*, "Open robotics research using Web-based knowledge services," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 5380–5387.
[25] (2018). *Apache Storm*. [Online]. Available: http://storm.apache.org/
[26] R. Arumugam *et al.*, "DAvinCi: A cloud computing framework for service robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2010, pp. 3084–3089.
[27] G. Mohanarajah, D. Hunziker,, R. D'Andrea, and M. Waibel, "Rapyuta: A cloud robotics platform," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 481–493, Apr. 2015.
[28] M. Chen, Y. Hao, M. Qiu, J. Song, D. Wu, and I. Humar, "Mobility-aware caching and computation offloading in 5G ultra-dense cellular networks," *Sensors*, vol. 16, no. 7, pp. 974–986, 2016.
[29] N. Kumar, M. Singh, S. Zeadally, J. J. P. C. Rodrigues, and S. Rho, "Cloud-assisted context-aware vehicular cyber-physical system for PHEVs in smart grid," *IEEE Syst. J.*, vol. 11, no. 1, pp. 140–151, Mar. 2017.
[30] M. Chen, W. Li, Y. Hao, Y. Qian, and I. Humar, "Edge cognitive computing based smart healthcare system," *Future Gener. Comput. Syst.*, vol. 86, pp. 403–411, Sep. 2018, doi: 10.1016/j.future.2018.03.054.
[31] A. Rahman, J. Jin, A. Cricenti, A. Rahman, and D. Yuan, "A cloud robotics framework of optimal task offloading for smart city applications," in *Proc. IEEE GLOBECOM*, Washington, DC, USA, Dec. 2016, pp. 1–7.
[32] L. Yang, J. Cao, S. Tang, D. Han, and N. Suri, "Run time application repartitioning in dynamic mobile cloud environments," *IEEE Trans. Cloud Comput.*, vol. 4, no. 3, pp. 336–348, Jul./Sep. 2016.
[33] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, Apr. 2018.
[34] H. Zhou, H. Wang, X. Li, and V. Leung, "A survey on mobile data offloading technologies," *IEEE Access.*, vol. 6, no. 1, pp. 5101–5111, Jan. 2018.
[35] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.
[36] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.
[37] M. Chen, Y. Hao, L. Hu, K. Huang, and V. Lau, "Green and mobility-aware caching in 5G networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 8347–8361, Dec. 2017.
[38] W. Chen, Y. Yaguchi, K. Naruse, Y. Watanobe, and K. Nakamura, "QoS-aware robotic streaming workflow allocation in cloud robotics systems," *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2018.2803826.2018.
[39] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018, doi: 10.1109/JSAC.2018.2815360.
[40] Z. Cheng, P. Li, J. Wang, and S. Guo, "Just-in-time code offloading for wearable computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 1, pp. 74–83, Mar. 2015.
[41] P. Pandey, D. Pompili, and J. Yi, "Dynamic collaboration between networked robots and clouds in resource-constrained environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 471–480, Apr. 2015.
[42] M. J. Meena, S. S. Prabha, and S. Pandian, "A cloud-based mobile robotic system for environmental monitoring," in *Proc. Asia–Pacific Conf. Comput. Aided Syst. Eng. (APCASE)*, Feb. 2014, pp. 122–126.
[43] M.-L. Lam and K.-Y. Lam, "Path planning as a service PPaaS: Cloud-based robotic path planning," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2014, pp. 1839–1844.
[44] O. Arslan, D. P. Guralnik, and D. E. Koditschek, "Coordinated robot navigation via hierarchical clustering," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 352–371, Apr. 2016.
[45] J. Salmerón-García, P. Íñigo-Blasco, F. Díaz-del-Río, and D. Cagigas-Muñiz, "A tradeoff analysis of a cloud-based robot navigation assistant using stereo image processing," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 444–454, Apr. 2015.
[46] R. G. Lins, S. N. Givigi, and P. G. Kurka, "Vision-based measurement for localization of objects in 3-D for robotic applications," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 11, pp. 2950–2958, Nov. 2015.
[47] G. Silano and L. Iannelli, "An educational simulation platform for GPS-denied unmanned aerial vehicles aimed to the detection and tracking of moving objects," in *Proc. IEEE Conf. Control Appl. (CCA)*, Buenos Aires, Argentina, Sep. 2016, pp. 1018–1023.
[48] B. D. Gouveia, D. Portugal, D. C. Silva, and L. Marques, "Computation sharing in distributed robotic systems: A case study on SLAM," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 410–422, Apr. 2015.
[49] P. Benavidez, M. Muppidi, P. Rad, J. J. Prevost, M. Jamshidi, and L. Brown, "Cloud-based realtime robotic visual SLAM," in *Proc. Annu. IEEE Syst. Conf. (SysCon)*, Vancouver, BC, Canada, Apr. 2015, pp. 773–777.
[50] A. Marjovi, S. Choobdar, and L. Marques, "Robotic clusters: Multi-robot systems as computer clusters: A topological map merging demonstration," *Robot. Auton. Syst.*, vol. 60, no. 9, pp. 1191–1204, 2012.
[51] Y. Li, H. Wang, B. Ding, P. Shi, and X. Liu, "Toward QoS-aware cloud robotic applications: A hybrid architecture and its implementation," in *Proc. Int. IEEE Conf. Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People, Smart World Congr.*, Toulouse, France, Jul. 2016, pp. 33–40.
[52] L. Wang, M. Liu, and M. Q.-H. Meng, "Real-time multisensor data retrieval for cloud robotic systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 507–518, Apr. 2015.
[53] D. M. Lofaro, A. Asokan, and E. M. Roderick, "Feasibility of cloud enabled humanoid robots: Development of low latency geographically adjacent real-time cloud control," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2015, pp. 519–526.
[54] L. Wang, M. Liu, and M. Q.-H. Meng, "A hierarchical auction-based mechanism for real-time resource allocation in cloud robotic systems," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 473–484, Feb. 2017.
[55] D.-H. Lee, S. A. Zaheer, and J. H. Kim, "Ad hoc network-based task allocation with resource-aware cost generation for multirobot systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6871–6881, Dec. 2014.
[56] H. Zhang, F. Fang, J. Cheng, K. Long, W. Wang, and V. C. M. Leung, "Energy-efficient resource allocation in NOMA heterogeneous networks," *IEEE Wireless Commun.*, vol. 25, no. 2, pp. 48–53, Apr. 2018.
[57] P. Shu *et al.*, "eTime: Energy-efficient transmission between cloud and mobile devices," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 195–199.
[58] R. Doriya, S. Mishra, and S. Gupta, "A brief survey and analysis of multi-robot communication and coordination," in *Proc. Int. Conf. Comput., Commun. Automat.*, May 2015, pp. 1014–1021.
[59] (2018). *High-Performance and Flexible Protocol (HpFP)*. [Online]. Available: http://hpfp.nict.go.jp/en/

[60] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
[61] S. Tadokoro, "Challenge of disaster robotics," in *Proc. 34th Chin. Control Conf. (CCC)*, Hangzhou, China, Jul. 2015, pp. 21–22.
[62] Y. Kakiuchi *et al.*, "Development of humanoid robot system for disaster response through team NEDO-JSK's approach to DARPA robotics challenge finals," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Nov. 2015, pp. 805–810.
[63] M. Okugawa *et al.*, "Proposal of inspection and rescue tasks for tunnel disasters—Task development of Japan virtual robotics challenge," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Oct. 2015, pp. 1–2.
[64] K. Nagatani *et al.*, "Emergency response to the nuclear accident at the Fukushima Daiichi nuclear power plants using mobile rescue robots," *J. Field Robot*, vol. 30, no. 1, pp. 44–63, Jan. 2013.
[65] (2018). *Layer 2-Based Wireless Relay Network for Robot Control Under Non Line-of-Sight Environments*. [Online]. Available: https://www.nict.go.jp/en/press/2016/09/16-1.html
[66] X. Wang, Y. Zhang, V. C. M. Leung, N. Guizani, and T. Jiang, "D2D big data: Content deliveries over wireless device-to-device sharing in large-scale mobile networks," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 32–38, Feb. 2018.
[67] K. Wei, X. Liang, and K. Xu, "A survey of social-aware routing protocols in delay tolerant networks: Applications, taxonomy and design-related issues," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 556–578, 1st Quart., 2014.
[68] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.
[69] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D Placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 434–437, Aug. 2017.
[70] J. Lu, S. Wan, X. Chen, Z. Chen, P. Fan, and K. B. Letaief, "Beyond empirical models: Pattern formation driven placement of UAV base stations," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3641–3655, Jun. 2018, doi: 10.1109/TWC.2018.2812167.2018.
[71] T. S. Tadele, T. de Vries, and S. Stramigioli, "The safety of domestic robotics: A survey of various safety-related publications," *IEEE Robot. Autom. Mag.*, vol. 21, no. 3, pp. 134–142, Sep. 2014.
[72] A. Banerjee, K. K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta, "Ensuring safety, security, and sustainability of mission-critical cyber–physical systems," *Proc. IEEE*, vol. 100, no. 1, pp. 283–299, Jan. 2012.
[73] M. Y. Jung and P. Kazanzides, "An architectural approach to safety of component-based robotic systems," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Stockholm, Sweden, May 2016, pp. 3360–3366.
[74] I. Yaqoob, E. Ahmed, A. Gani, S. Mokhtar, and M. Imran, "Heterogeneity-aware task allocation in mobile ad hoc cloud," *IEEE Access*, vol. 5, pp. 1779–1795, Feb. 2017.
[75] W. Xiao, W. Bao, X. Zhu, W. Zhou, and P. Luy, "Improving the performance of data sharing in dynamic peer-to-peer mobile cloud," in *Proc. IEEE Int. Conf. Parallel Distrib. Syst.*, Dec. 2016, pp. 743–752.
[76] W. Li, Y. Zhao, S. Lu, and D. Chen, "Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 89–97, Mar. 2015.
[77] W. Yu *et al.*, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.
[78] J. Mahler, B. Hou, S. Niyaz, F. T. Pokorny, R. Chandra, and K. Goldberg, "Privacy-preserving grasp planning in the cloud," in *Proc. IEEE Int. Conf. Automat. Sci. Eng.*, Aug. 2016, pp. 468–475.
[79] F. Liu *et al.*, "Gearing resource-poor mobile devices with powerful clouds: Architectures, challenges, and applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 14–22, Jun. 2013.
[80] T. H. Noor, Q. Z. Sheng, L. Yao, S. Dustdar, and A. H. H. Ngu, "CloudArmor: Supporting reputation-based trust management for cloud services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 2, pp. 367–380, Feb. 2016.
[81] J. Jiang, G. Han, C. Zhu, S. Chan, and J. J. P. C. Rodrigues, "A trust cloud model for underwater wireless sensor networks," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 110–116, Mar. 2017.
[82] C. Zhu, J. J. P. C. Rodrigues, V. C. M. Leung, L. Shu, and L. T. Yang, "Trust-based communication for the industrial Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 16–22, Feb. 2018.
[83] E. C. Ferrer. (2016). "The blockchain: A new framework for robotic swarm systems." [Online]. Available: https://arxiv.org/abs/1608.00695
[84] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3154–3164, Dec. 2017.
[85] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.



WUHUI CHEN received the bachelor's degree from the Software College, Northeast University, China, in 2008, and the master's and Ph.D. degrees from the School of Computer Science and Engineering, The University of Aizu, Japan, in 2011 and 2014, respectively. He was a Researcher with the Revitalization Center, The University of Aizu. He is currently an Associate Professor with Sun Yat-sen University, China. His research interests include services computing, cloud computing, and cloud robotics.

YUICHI YAGUCHI is currently an Associate Professor with the School of Computer Science and Engineering, The University of Aizu, Japan. His research interests include drone formation flight, cloud robotics, SLAM, 3-D reconstruction from aerial image, and drone automatic control.

KEITARO NARUSE is currently a Full Professor with the School of Computer Science and Engineering, The University of Aizu, Japan. His research interests include swarm robots and its application to agricultural robotic systems, and interface system for disaster response robots.

YUTAKA WATANOBE is currently a Senior Associate Professor with the School of Computer Science and Engineering, The University of Aizu, Japan. His research interests include visual programming language, data mining, and cloud robotics.

KEITA NAKAMURA is currently an Associate Professor with the Revitalization Center, The University of Aizu, Japan. His research interests include robotic simulation, swarm robots and its applications, and cloud robotics.

JUN OGAWA is currently an Associate Professor with the Revitalization Center, The University of Aizu, Japan. His research interests include evolutionary soft robotics, development of generic physics engine for soft robot simulator, and soft-bodied agent simulation.