Article

Smart Resources for Smart Cities: Distributed Architecture to Integrate Sensor Information

Jose-Luis Poza-Lujan [1,†*], Juan-Luis Posadas-Yagüe [1,†], Jose E. Simó[1,†] and Francisco Blanes [1,†]

[1] University Institute of Control Systems and Industrial Computing (ai2). Universitat Politècnica de València (UPV) Camino de Vera, s/n. 46022 Valencia (Spain); jopolu@ai2.upv.es, jposadas@ai2.upv.es, jsimo@ai2.upv.es, fblanes@ai2.upv.es

* Correspondence: jopolu@ai2.upv.es; Tel.: +34-963877000 (J.P.)

† These authors contributed equally to this work.

Abstract: Objects recognition is a necessary task in smart city environments. This recognition can be used in processes such as the reconstruction of the environment map or the intelligent navigation of vehicles. This paper proposes an architecture that integrates heterogeneous distributed information to recognize objects in intelligent environments. The architecture is based on the IoT / Industry 4.0 model to interconnect the devices, called Smart Resources. Smart Resources can process local sensor data and send information to other devices. These other devices can be located in the same operating range, the Edge, in the same intranet, the Fog, or on the Internet, the Cloud. Smart Resources must have an intelligent layer in order to be able to process the information. A system with two Smart Resources equipped with different image sensors has been implemented to validate the architecture. Experiments show that the integration of information increases the certainty in the recognition of objects between 2% and 4%. Consequently, in intelligent environments, it seems appropriate to provide the devices with intelligence, but also capabilities to collaborate closely with other devices.

Keywords: Smart Environment, Smart Sensors, Distributed Architectures, Object Detection, Information Integration

---

1. Introduction

The growth of cities has given rise to an environment populated by increasingly intelligent and more connected devices. Practically all of them have sensors and actuators with very different capacities.

Classically, each one of these devices with sensors and actuators can be considered as a control node. However, because of devices are connected between them and they can communicate to share their resources, the concept of control node can change to the concept of intelligent resource that provides services to the rest of the devices [1]. In addition, heterogeneous devices provide complementary information that can be used to enrich the knowledge. Consequently, a distributed system in which the devices are intelligent can be defined as an intelligent system or Smart System [2]. Smart systems perform their tasks in dynamic environments with multiple features and changing conditions. Urban environments are an example of dynamic systems, not predictable, in order to apply intelligent systems.

Therefore, the continuous and accurate knowledge of the environment is necessary to provide autonomy and interact. Subsystems such as robots or vehicles navigation need to know the environment to perform their tasks such as the planning of trajectories or the execution of missions [3]. Most urban systems are based on devices with many sensors. A device with many sensors may

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

be overloaded, or its sensors can be used very sporadically, with the consequent loss of efficiency. To reduce the load on each device, it is important to be able to exchange sensory information between them. For this, devices have to communicate not only with the devices of their own semantic level, but with any element of the system. It is for this reason that devices in smart cities could be decoupled from any hierarchy. Consequently, smart city architectures can be organised according to the amount of information and frequency of exchange of such information. Therefore, interacting at the information level instead of at the sensor level facilitates the intelligence of the system.

The fact that the elements of the system can exchange information between them, decoupling their location in the hierarchy of the architecture, has led to the emergence of new paradigms (Figure 1). These paradigms and standard architectures organise the components depending on dimensions according to the amount of data with which they work, to the geographical scope, or to the immediacy (real time) of the messages and responses of control [4]. Based on the dimensions proposed in [4], the Industry 4.0 or Internet of Things (IoT) models stand out as very suitable for the design of any system that provides intelligence to a city. The Reference Architectural Model Industry 4.0 (RAMI 4.0) [5] is based on "smart products", which the IoT [6] model places in a layer called "Edge". Examples of these products range from robots in an assembly line to smart street lamps that optimise energy consumption. These devices have a scope of sensorization and action of the order of a few meters, and a very fast reaction. When several devices of the Edge level communicate or interact on a wider level, spatially or temporarily, it is called the smart factory or platform level. For example, when monitoring the performance of all robots in an assembly line or when managing the lighting of an entire city. Finally, when the devices are connected to exchange large amounts of data or there is a longer-term reaction, it is called a connected world, the business level or the well-known concept of "cloud". The control architectures of smart cities fit perfectly into these models [7].

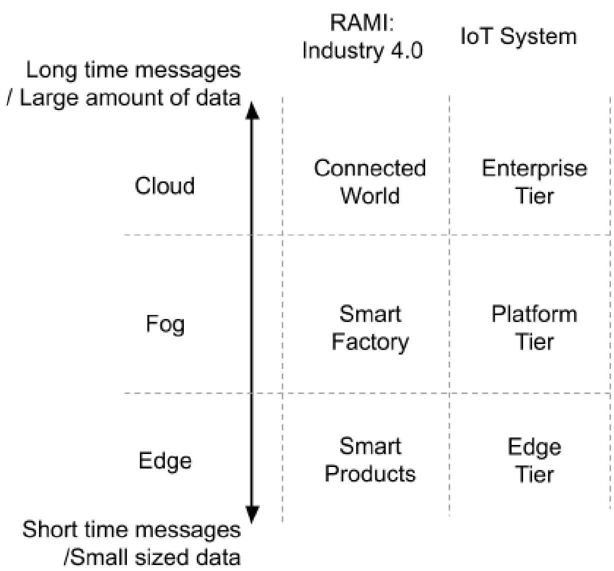

Figure 1. Architecture models in smart cities

The paradigms shown in Figure 1 coincide in organizing the elements according to the level of information or the frequency of access, rather than a dependence on the physical or logical topology. Therefore, this article proposes a collaboration model for object recognition whose location and models are in the Edge layer.

The recognition of objects is one of the usual functionalities required by the elements of an intelligent city. For example, vehicles need to recognise traffic signals for autonomous driving [8]. Other applications of the detection of objects in cities are the detection of people [9] or the detection of vehicles [10], generally to improve road safety or the comfort of citizens.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

Sensors can help to know the environment by detecting objects and some of their characteristics. However, when the objects detected have to be classified and recognised, a set of patterns with which to compare [11] is necessary. For example, the shape of a box can be detected by means of a 3D sensor, but the same box can have different textures, so it is also necessary other type of sensor, for example an RGB sensor, to recognise what type the box is. Therefore, using heterogeneous sensors to detect and recognise the objects placed in an environment can increase the probability of success recognising the right object. When working with heterogeneous sensors, their information must be merged, usually remotely creating sensor networks [12]. As a summary, acquiring characteristics of the environment to associate them with specific objects implies a sequence of actions shown in the Figure 2.

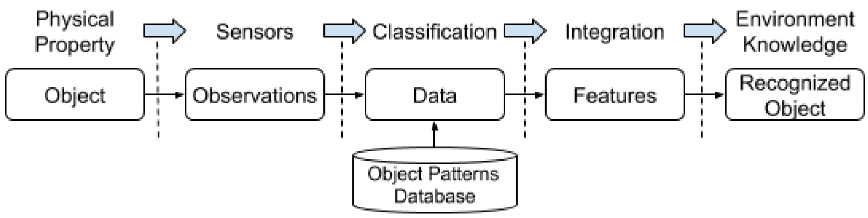

Figure 2. Overview of the components of the recognition process in the integration of sensory information

The inclusion of object detection in the environment map adds a difficulty and force the use of advanced sensors. Consequently, when there are many sensors the data fusion depends on the fusion mechanism [13]. Once a certain precision in the detection of the object and its characteristics has been achieved, it should be possible to classify the object [14]. The classification of the object requires the use of patterns in order to compare the percentage of similarity [15]. Therefore, in an object recognition system, a database of patterns is necessary. In the current models presented in Figure 1, the different components of the process presented in Figure 1 can be located at any level. The sensors belong to the Edge, while classification and integration are processes that can be on the local device (Edge), on any nearby device (Fog) or on dedicated servers (Cloud).

The addition of micro-controllers and micro-processors to the sensors devices, has increased the information capacity that sensors can provide. These devices are usually called respectively smart, or intelligent sensors [16]. When the sensor includes some advanced processing and, in some occasions, actuators, some authors call them smart devices [17]. Adding communication interfaces allows smart devices to share information and, consequently increase the knowledge of the environment. The use of smart devices is growing from the environment like Smart cities [18] to the concept of Smart Objects when these devices are included into the daily life of people [19]

Consequently, the current situation is that sensors can send processed information rather than raw data. The result is that sensor networks become into distributed systems that integrate sensor information in order to take advantage of the processed information [20]. When there are different distributed devices, there are some interesting problems to solve. One of the problems is to achieve a balance between the number of devices used and the correct use of their sensors. That is, when a new device is introduced, its sensors should help increase the probability of success when detecting and recognising an object. Consequently, the composition and connection between the devices will determine how to recognise the objects. For example, two devices with an RGB sensor can recognise the same texture with a similar probability. However, the probability of success could increase by using another type of sensor which reinforces the process, for example, a thermal camera that can distinguish between different ink types. The diversity of available sensors is especially relevant in smart cities. In the same street, it is possible to find traffic cameras at fixed points, but also navigation cameras in vehicles. Both types of cameras can cooperate for the identification of objects on the tracks. In this way, it is possible to access a system that allows to distinguish between authorised objects (such

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

as cleaning and work temporal signals) and objects not allowed (such as garbage or even potentially dangerous objects).

To study how different types of devices can cooperate, the Smart Resource model used in previous researches has been used [21]. Smart resources allow high connection flexibility since the features are offered as services. Services offered depend on the available sensors and the computing capacity of each smart resource. Clients determine the necessary services, establishing a connection topology depending on their needs. For example, in the case of a smart resource that detects and identifies an object with a high probability, more information to corroborate the identified object could not be required. However, if the probability is low, the smart resource will need other measurements from other sensors that allow to increase the probability to identify successfully the object.

Figure 3 shows an example in which various devices have to detect objects in an urban environment. If both vehicles have cameras, they will be able to detect objects based on the patterns they have and the type of camera. When the vehicles are in a nearby environment, they will be able to dialogue in order to increase the certainty of their observations. In this way, if the driver of the bike is interested in looking for a waster bin and the electric scooter has recognised the object with more certainty, the electric scooter will be able to facilitate the location of the waster bin to the driver of the bike.

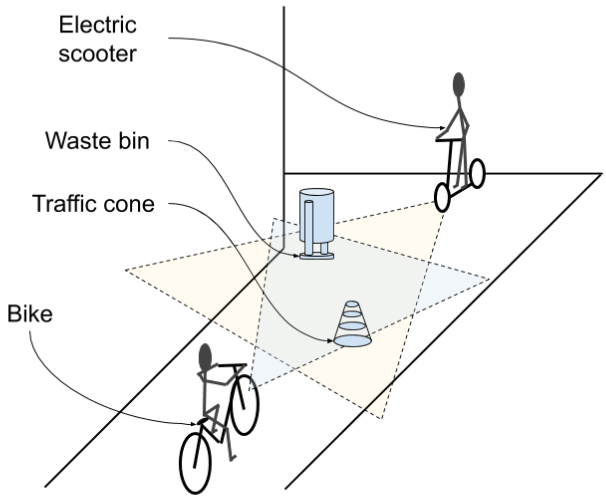

Figure 3. Urban environment in which various smart devices can collaborate in order to detect objects more accurately

Integration of sensory information is produced along with all levels of the architecture and provides a layer over sensory fusion to enrich the semantic meaning of the information provided to other components. Research into the integration of sensory information is based on the enrichment of the semantic meaning of the information depending on the architecture level where is integrated. The more architecture level, the more semantic meaning. For example, in the lower level of the architecture could be interesting to provide all the values of a temperature sensor but in a higher level the interesting information could be the average of these values and its comparison with the values of other temperature sensors. Sensory information is shared by the components of the architecture transparently to their location. Components could be close or located in the cloud. Therefore, it is interesting to study how the functionalities offered by the system devices improve through the integration of sensory information.

The aim of this paper is to present the study and implementation of a solution to integrate sensory information in order to increase certainty in the object recognition process. Currently, the elements of smart cities are increasing their capacity of processing and communication. Therefore, it is interesting to study how the efficiency of the systems can improve when their heterogeneous elements collaborate in order to integrate the information they have.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

Paper has been organised as follow. Once the aim of the investigation has been contextualised, section 2, materials and methods, describes the proposed architecture and its classification according to the models presented above. Next, the system implemented with their sensors and methods is presented. The section 3 presents the scenario on which the system has been tested and the results of the experiments performed for the recognition of two different objects. The results obtained verify how the integration of information from the services provided by the smart resources improves the accuracy to detect objects. Next, section 4 of Discussion analyses the results and the repercussions of the proposed architecture. Finally, the conclusions are drawn and some of the future lines to be developed are presented.

2. Materials and Methods

2.1. System Architecture

According to the concept of cloud, fog and edge computing, and the need to have elements capable of interacting between all layers, an architecture has been designed whose components are based on the Smart Resources (Figure 4).

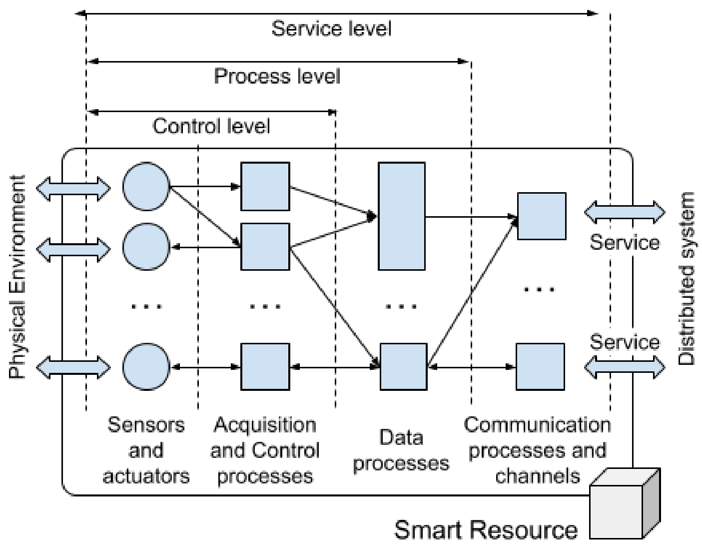

Figure 4. Concept and components of an Smart Resource. From interaction with the physical world (left) to interaction with the rest of the system (right)

An intelligent resource is defined as an element of intelligent control that offers its capabilities for interaction with the environment through services. As an intelligent control element, it has a direct connection to the physical environment through a set of sensors and actuators. In order to carry out the control actions, the smart resource has the functions of acquisition, reactive processing and action. Up to this point, a smart resource does not differ from a control node. For example, a traffic light with a VGA camera, a set of relays to control the light, and an Arduino Microcontroller with network connection constitute a control node. The role of the microcontroller is to acquire and transmit images, and to receive orders to turn on or off the lights. However, depending on the processing capacity of the smart resource and the functionalities it offers, there will be a set of processes with a higher level of intelligence. If the previous device was provided with a more powerful microprocessor that allows, for example, to store a history to infer the evolution of traffic, or to detect the number of vehicles waiting, it would be a Smart Resource. These advanced features are offered through services to other Smart Resources or system elements.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

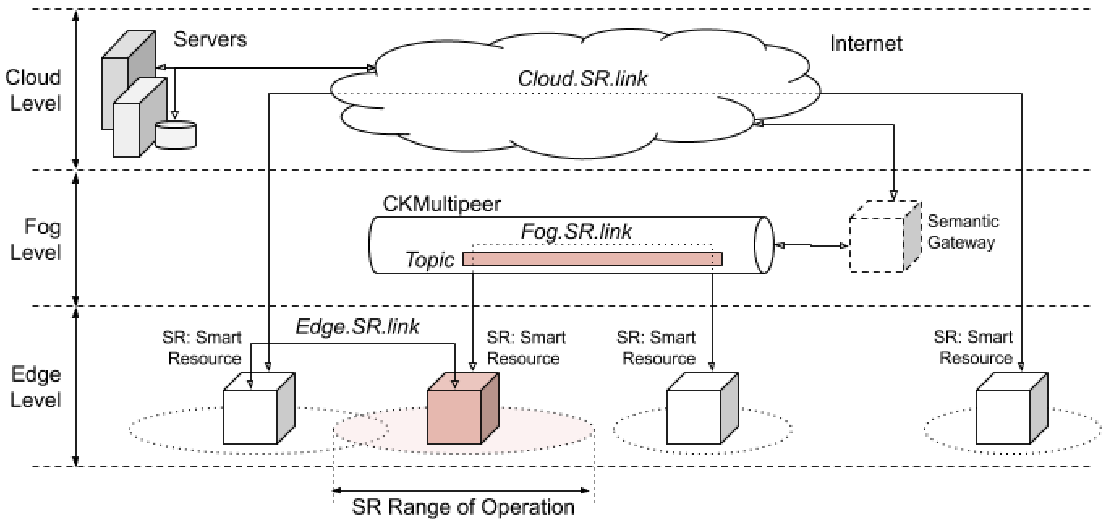

Figure 5. Location of the Smart Resources in the architecture. Edge interaction is possible when smart resources are physically in contact, that is, they overlap their operating ranges. The interaction in the Fog allows communication with real-time restrictions between Smart Resources without the need to share the physical space. The cloud interaction allows connection to other components and data servers without real-time restrictions.

One of the characteristics of smart city architectures is the interconnection capacity between elements or components. Connections between elements must be transparent to their location in the fog or in the cloud. A Smart Resource can have a communications system that allows connections at both the cloud and fog levels, or even at the edge level. In this last case, smart resources are physically very close (direct contact on the Edge) and the communication channels used are specific, such as Bluetooth or direct connection such as I2C.

Previously, has been discussed the importance of having mechanisms for semantic information conversion. For example, when a big number of calculations are required on a historical archive of temperature samples to obtain an daily average or predict a trend with a time horizon of one day. These semantic conversions require knowledge (for example the temperatures history found in the Cloud), and some data found directly within the smart resources. The element that allows semantic conversions between Fog and Cloud, is called the Semantic Gateway. The Semantic Gateway acts as a broker that can provide information to the Edge.

Connections in the Edge are possible when two smart resources are in the same physical space or operating range. This operating range is defined as the physical space where the sensors and actuators of an Smart Resource interact. For example, when a person rides on a bicycle, the sensors of the bicycle can connect and collaborate with the sensors of the person. The position sensors of a mobile device of the cyclist can collaborate with a camera installed on the bicycle to transmit the route or recognise objects. This collaboration is interesting because the same device can collaborate with other devices during different intervals of time. Therefore, a communication protocol that allows the collaboration between heterogeneous devices is necessary. Figure 6 shows the proposed protocol oriented to services.

Peer-reviewed version available at *Sensors* **2019**, *20*, 112; doi:10.3390/s20010112

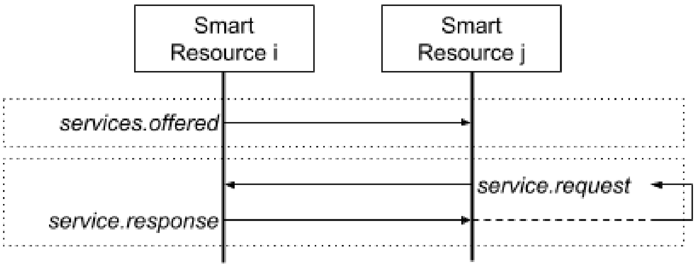

Figure 6. Communication protocol diagram between two Smart Resources at the Edge level (Edge.SR.link).

The diagram in figure 6 is placed into the application level. When two smart resources have been connected, both offer their services by exchanging a JSON message. A smart resource i offers its services to another smart resource j. When the smart resource j requires a service of the smart resource i, it will request it and the smart resource i will generate a response with the result or information provided by the service. That response could include an action, too.

At the Fog level, a channel with connection management is necessary, which allows communication between elements independent their location. The Publish-Subscribe [22] paradigm is one of the most suitable since it allows to uncouple physically the devices involved in the communication and connect each device to the information that they are interested in. Communications system used is the CKMultiPeer, described in [23], CKMultiPeer is based in the Data Distribution Service (DDS) model, widely used in critical distributed systems [24]. Communication through CKMultipeer is done through Topics. A topic is a common communication space in which information can be published. The element that publishes the information writes the data in a specific Topic. The elements that wish to know the information of the Topic, subscribe to this Topic and, when the information changes, they receive a message with the new data.

The communication protocol between two smart resources using the CKMultipeer at the Fog level is shown in detail in the figure 7

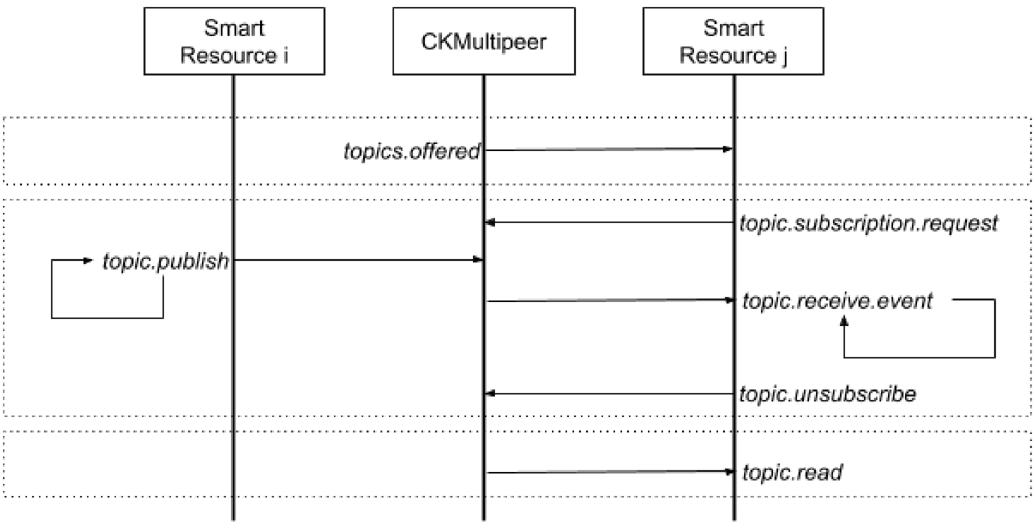

Figure 7. Communication protocol diagram between the CKMultipeer broker and the Smart Resources at the Fog level (Fog.SR.link).

The communication protocol at the Fog level is based on the publish-subscriber paradigm. The exchange of services is carried out through topics and the Smart Resources interact decoupled. First, the Smart Resources request at the Fog the list of available Topics through the broker called CKMultipeer. As a result, they receive a JSON file with the list of services offered, features and service quality

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

parameters [25]. When the Smart Resource j requires a service, it subscribes to the associated topic through the CKMultipeer broker. CKMultipeer is responsible to send automatically to subscribers all information published in the topics. When the smart resource, that provides the service required (smart resource i, in the figure 7), publishes new information in the associated topic, the CKMultipeer notifies and sends the new information to all smart resources subscribed. Subscribers read asynchronously the information, that is, by means of a notifications model based on events. When the smart resource j does not need to receive more information, it can unsubscribe from the corresponding topic. Additionally, Smart Resources can read any topic synchronously without any subscription.

2.2. Smart resource-based object recognition

As described in the previous section, the classification and integration steps imply the use of patterns and the decision is based on the probabilities of each pattern provided from the different sensors. The process is described in Figure 8.

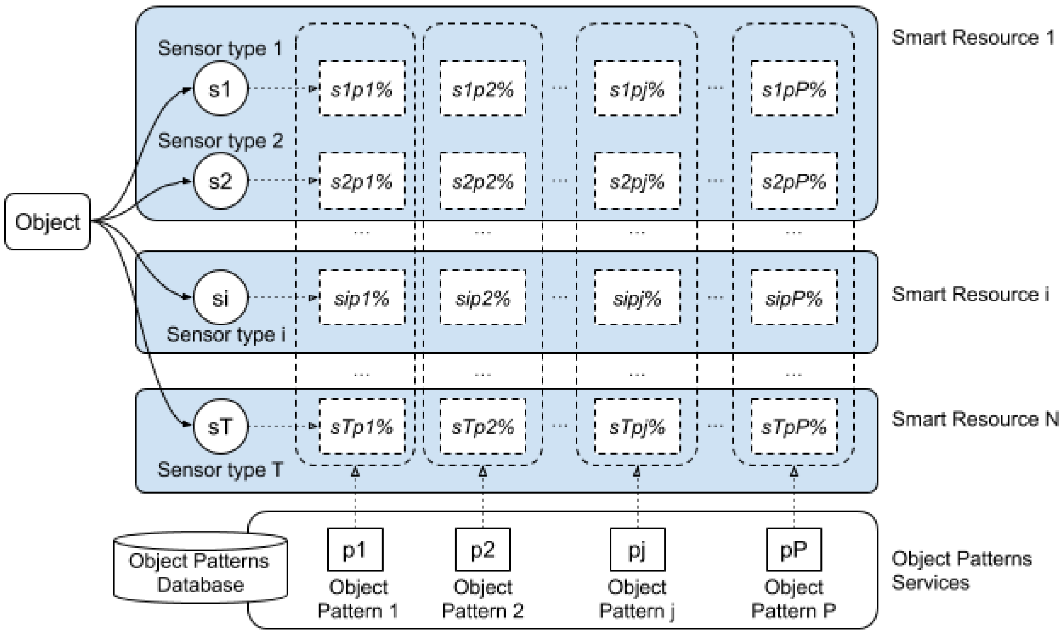

Figure 8. Smart Resources scheme and connection with the patterns for object recognition based on the analysis of the similarity between measurements and patterns.

Taking into account that each object j has a specific pattern from each type of sensor i. A pattern is built with the object characteristics detected from a type of sensor. If the whole process was centralised, each device should have access to as many sensors as possible, and the patterns to compare with those sensors. The storage load of all the patterns and the processing load of all the sensors in the same device could be too much. In addition, in the case that a sensor obtains a very high probability with a specific object pattern, it would be not necessary to continue processing more sensors, unless a 100 % certainty was required. Therefore, a distributed system can be an adequate and efficient solution [26]. In this system, only when a device has a low certainty in the recognition of an object, it should request more results from other devices that have recognised that object. So, a device A needs to have only the patterns of the sensors used. The device A should be able to consult a device B about an object, in order to reinforce the probability of the recognised object. In order to distribute the object recognition process, a system based on distributed intelligent devices, called Smart Resources, has been developed. The smart resource model is described in [27]. A smart resource is an extension of a smart device, that offers services to the others system elements. For example, in the system described in the next

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1
Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

section, smart resources offer as a services, among others, their position in the map, and the probability detected for each pattern.

In order to communicate the devices, a communication system is needed that allows the subscription to specific services, offering a balanced network load. For example, in the Figure 8, the object of pattern 1 may have associated the sensors type 1, 2, and 3. But if there is a device that only has a sensor type 1 and 2, and another device with the type 3 sensor, it is convenient for both devices to send and receive information of a type of pattern and not of a specific device.

The communication system allows a device to connect to a source of information (the pattern of a specific object) from which to obtain data that can reinforce the identification of a specific object, fulfilling the requirements mentioned above.

2.3. System implemented

To validate the architecture and test the operation of smart resources a system with two smart resources has been implemented.

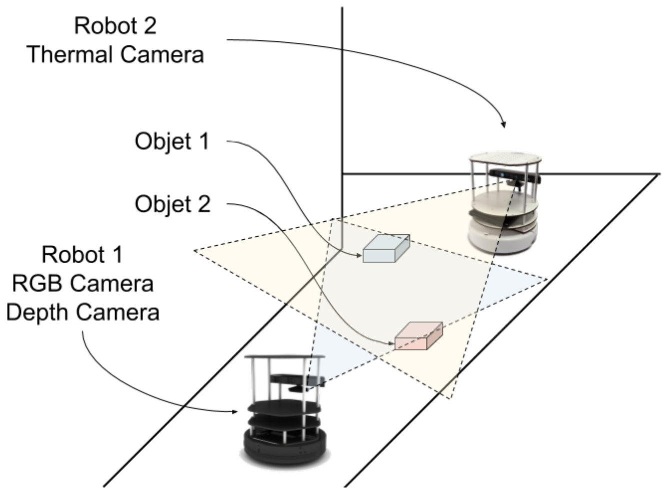

Figure 9. The case study used in the experiments. The vehicles are replaced by autonomous robots and street objects replaced by boxes. The robots can be controlled better than a real bike or scooter, this allows the experiment to be replicated.

System is shown in figure 10. Two robots Turtlebot [28] carry on the smart resources. Any smart resource is composed by one BeagleBone [29], the corresponding sensors and one IEEE802.11 interface to communicate between them. In the experiments performed, real-world vehicles are replaced by robots. Using well known and controllable robots, the experiments can be replicated with identical vehicles vehaviour and movements. Turtlebot 1 carry on the Smart Resource 1 and Turtlebot carry on Smart Resource 2. The first smart resource used, has two sensors, a deep camera to detect the geometry and an RGB camera to detect the texture. The second smart resource has only a sensor, a thermal camera that produces an RGB associated to the colour reflected. The colour of the image depends on the temperature and is directly associated with the ink composition.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

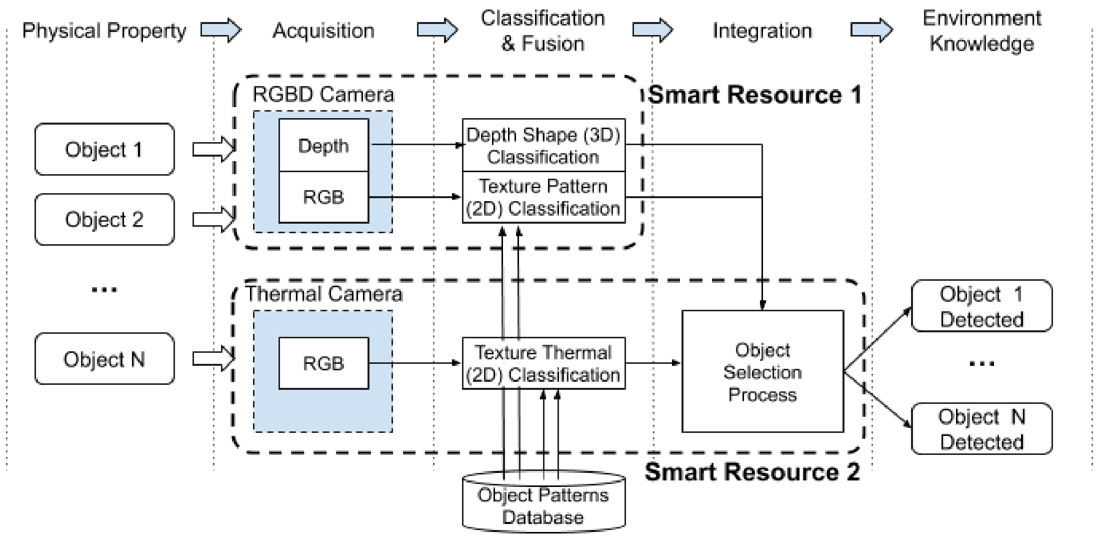

Figure 10. Details of the system implemented to perform the experiments and the corresponding step associated with the component (top of the figure).

The reason for using a different RGB sensor is to be able to use the same recognition algorithms (2D image), but with different patterns of the same object.

2.4. Scenario and experiment performed

The objective of the experiment is to characterise the performance of the presented architecture. The experiments that were performed evaluate the obtained results in both single and multi-robot approaches. While the single robot experiments offer information for characterising the access to on-board smart devices, the multi-robot approach shows how to deal with spatially decoupled sensor. In order to provide these statistical values, a set of environment features are recognised and integrated as environment knowledge. This set includes heterogeneous object samples 11, which are representative for testing.

Figure 11. Objects (boxes) used in the experiments. Other objects are used to measure the false positives rate.

In the proposed system shown in the Figure 11, two specific objects are proposed to be detected and recognised by means of two different smart resources. Both objects have the same geometry (boxes) but different textures (the box of a Xtion and the box of a BeagleBone).

The patterns used to recognize the textures have been the images of the six sides of each face of the boxes to be detected. Both smart resources contain both the texture (images) and geometry

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

(3D shape) patterns of the two boxes. Therefore, the matching pattern-sensor measurement will be performed in both smart resources. Figure 12 shows the processes that will be carried out based on smart resources and boxes.

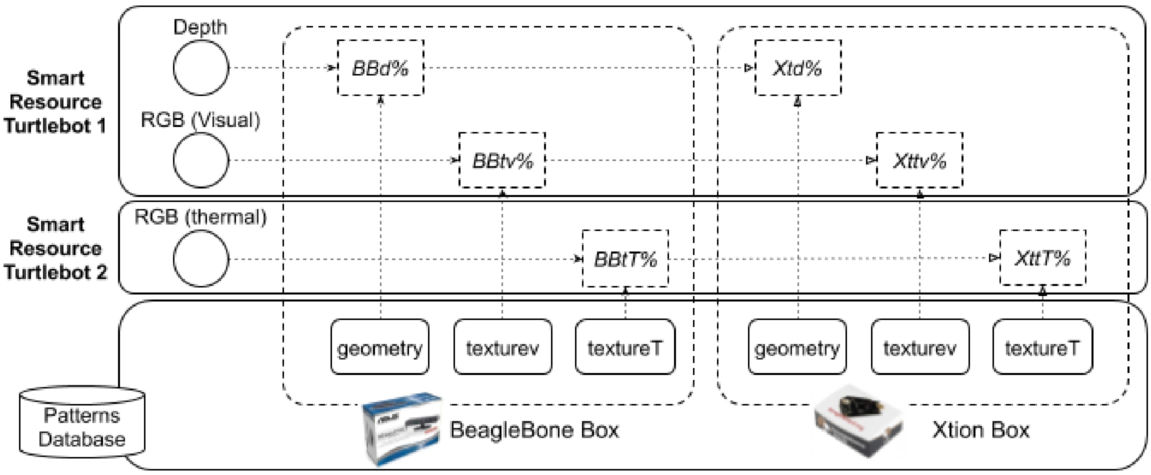

Figure 12. Objects (boxes) used in the experiments, sensor into the smart resources, and patters used to recognise the boxes.

The experiment starts when the two robots find the box. First was tested with the Beaglebone box, and after the Xtion box. When Smart Resource 1 detects a box with a reasonable prospect of certainty (upper than 0.500) publish the estimated box position, time and certainly value in the topic 'BBBox' or 'XtionBox'. A topic is a common space to share data in a Publish-subscribe system. Smart Resource 2, receives the data of the certainly of both boxes and integrates the information with the data obtained from their sensors. To check the integration efficiency compensates the transmission time, all data path between both smart resources must be checked (Figure 13)

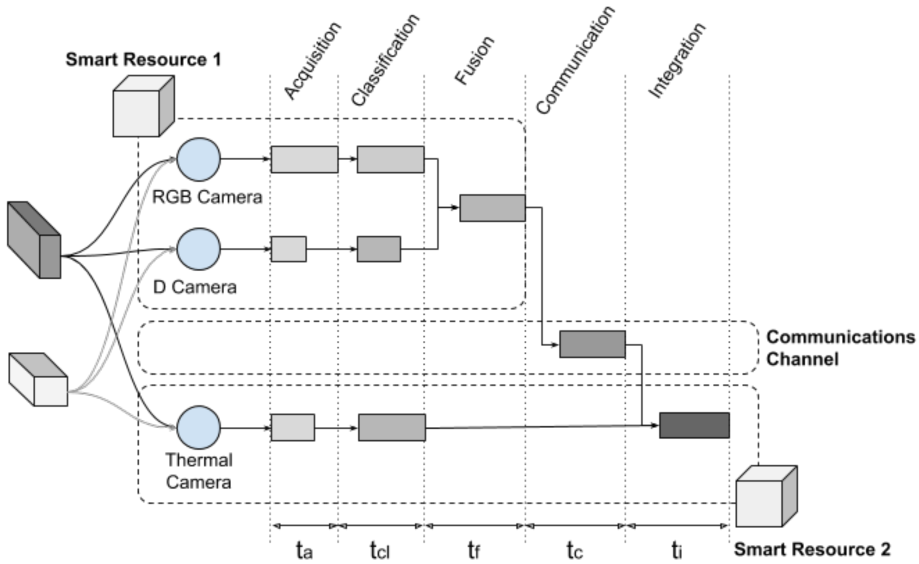

Figure 13. Data path of the experiment performed. From the data adquisition (left) by means of the smart resources sensors to the result obtained with the integration of the information (right).

At the bottom of Figure 13 the times used by each smart resource to classify an object are shown. The time t_a refers to the acquisition time of the images by the cameras used. In the experiments it remains constant depending on the sensor used.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

The time t_cl is the time it takes to classify the images according to the available patterns. A Reliability-Based Particle Filter was used for classification, which is shown in [30]. In order not to alter the experiments, the patterns were already preloaded in each intelligent resource. When the pattern is not available, the Smart Resource has to request it from the fog through the CKMultipeer or even from the cloud.

The time t_f is the time it takes to fusion various results. An Extended Kalman Filter (EKF) very used in similar environments was used [31]. It is important that, once the object detected by the camera is classified, the percentage of certainty is used as the inverse error (the greater certainty, the less error) in the measurement of a sensor. That is, when comparing an image with a pattern, for example the BBBox box, it is considered a BBBox box sensor with a specific percentage of certainty. The integration must provide an increase in certainty in the recognition of an object. But for integration, the smart resources must communicate between them. The time t_c is the communications time. CKMultipeer was used on a 54Mpps WiFi network.

The time t_i is the time of the integration of the results. A summary of the average times obtained is shown in the results section.

3. Results

3.1. Latency time

Experiments that are detailed in this section make use of the robotic platform Turtlebot 2 [32]. Two Turtlebot robots with heterogeneous sensor configuration have been designed for these experiments. Every sensor has been integrated as a part of a different Smart Resource. Therefore, sensory information and its management are always accessed through the distributed services that are provided by the related Smart Resource. First Turtlebot configuration implements an Asus Xtion Pro(depth camera) Smart Resource as a 3D sensor [33], and a monocular RGB camera Smart Resource as a 2D sensor. Therefore, these experiments include two different feature classifications that involve heterogeneous magnitude observations in the 2D and the 3D plane. Second Turtlebot has endowed with thermal camera Smart Resource, which adds a new magnitude classification in the 2D plane. The smart resources were implemented on a beaglebone board [34]. This board has an ARM Cortex 8 microprocessor with 1 GHz processing frequency. In order to make different experiments with different smart resource links, we use the CKMultipeer middleware to Fog and Cloud level. The average times of the message latency times obtained in these channels are obtained using the protocol presented in the previous section.

The figure 13 shows the total time to obtain a final result of information integration, that is the addition of all times involved in the data path. Times have been taken from processes running independently. Table 1 shows the estimated process times based on the decoupled tasks and results. In the case that a smart resource has two sensors (turtlebot 1) common times, as t_a or t_cl, considered are the worst time between all sensors times.

Table 1. Resuls of the average estimated times in each stage of the process. The acronym "n.a". (not appropriate) means that time is not provided because this phase is not done in the experiment.

Object	SR	t_a	t_cl	t_f	t_c	t_i	Total
BBox	turtlebot 1	16 ms.	12 ms.	25 ms.	n.a.	n.a.	53 ms.
	turtlebot 2	32 ms.	18 ms.	n.a.	n.a.	n.a.	50 ms.
	Integration (Fog)	n.a.	n.a.	n.a.	244 ms.	23 ms	320 ms.
XtionBox	turtlebot 1	16 ms.	15 ms.	24 ms.	n.a.	n.a.	55 ms.
	turtlebot 2	32 ms.	17 ms.	n.a.	n.a.	n.a.	49 ms.
	Integration (Fog)	n.a.	n.a.	n.a.	244 ms.	25 ms.	318 ms.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

It should be noted that the process times are similar, due to both smart resources use the same libraries, microcontroller board, and the boxes to detect are quite similar. It can be see, CKMultipeer introduces high latency. To check if information integration is profitable, it is necessary to study if the latency time and spent to increase object recognition can justify a low percentage of certainly improved. The ratio obtained using local integration and colaborative integration is studied in the next subsection.

3.2. Object recognition

In the proposed scenario, the two robots (Turtlebot 1 and Turtlebot 2) navigate until they detect the same set of objects. Both robots have a different perspective, and they are located correctly on the map. To show the process better, the results of each detected object have been shown separately.

The table 2 shows the results obtained when the data from the sensors of the first robot (Turtlebot 1) is compared with the geometries of boxes. It can be seen that both are very similar, with a certain difference favourable to the box of the BeagleBone. In the case of texture, the RGB sensor clearly detects a tendency to the correct object.

Table 2. Results in the percentage of certainty in the correct object detection applying the integration method with two similar objects patterns (BeagleBone box)

Object: BeagleBone box	**Turtlebot 1**			**Turtlebot 2**	
Object Pattern	Geometry	Texture	Fusion	Texture	Integration
BeagleBone box	0.726	0.671	0.792	0.789	0.824
Asus Xtion box	0.647	0.127	0.651	0.192	0.658

As can be seen in the table, when the Smart Resource 2 requests the system (through the CKMultiPeer topics) the certainty of the object, the correct object is always reinforced. Data in the opposite direction, when the integration is done by Smart Resource 1, and the information of certainty is provided by Smart Resource 2, are similar. Consequently, it is possible that two uncoupled and heterogeneous systems collaborate to improve their perceptions.

Table 3. Results in the percentage of certainty in the correct object detection applying the integration method with two similar objects patterns (XTion box)

Object: Xtion box	**Turtlebot 1**			**Turtlebot 2**	
Object Pattern	Geometry	Texture	Fusion	Texture	Integration
BeagleBone box	0.243	0.231	0.253	0.210	0.259
Asus Xtion box	0.851	0.712	0.886	0.812	0.902

The Turtlebot 2, has also detected the two objects, but only by means of texture. Consequently, the smart resource of the Turtlebot 2, requests the texture service to the , and upon receiving the data, the correct object is reinforced. When merging the information, in the Turtlebot 2 the object recognized as the BeagleBone box will be reinforced much more than the Xtion box object (table 3). In the case of the second object, you can check the same trend, but it is the box of the Xtion that is reinforced.

4. Discussion

In the experiments and results presented previously, it can be seen how the local processing of information is less expensive in time than the information integration. On average, information integration spends, in terms of time, six times than the classification and subsequent local integration. However, the information integration provides an improvement in object recognition. Obviously, with the results obtained, the waiting time is high only to improve a low percentage in recognition. The presented system allows a smart resource to decide if the percentage of local certainty recognizing an

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 November 2019 doi:10.20944/preprints201911.0035.v1

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

object is enough to make a decision, or if it is better to use information integration with other smart resources to reinforce certainty. For example, if a user is looking for a specific object such as a waste bin, it is probably enough to recognize it with low certainty and wait for the vehicle to recognize another with a higher percentage. After all, there are many waste bin in the city. However, if the object recognized with little certainty is a traffic cone, which indicates a problem and a potential accident, it is better for the vehicle to decrease their speed and ask the nearby smart resources for more information to integrate it, and increase the certainty rate. From the result of the information integration, a smart resource can take a decision, for example, if it is possible to avoid the obstacle or better stops the vehicle.

The system used and experiments performed are placed on the Fog level. Fog level implies the use of a common communication channel between all smart resources. Additionally, the communication channel, must manage the connection between smart resources and third parties. The use of a common communication channel provides smart resources of transparent and decoupled access to information. If two, or more, smart resources need to be coupled, for example, to share high-speed or high-frequency rate information, it is necessary to use a communication channel close to the Edge level. This communications channel can be oriented to a physical connection, such as I2C or wireless such as Bluetooth.

To know what improvements the Fog brings, local accuracy has been compared with the accuracy obtained by the collaboration of the two smart resources 4. In this table, optimization is obtained by dividing between the best local accuracy and fog accuracy. The cost of optimization is obtained by dividing into optimization and maximum latency. This last parameter indicates how many milliseconds we need to spent to increase the accuracy in one percentual point.

Table 4. Comparison results between local and fog accuracy and latency, and optimization

		Local		Fog			
Object	SR	Accuracy	Latency	Accuracy	Latency	Optimization	Optimization Cost
BBBox	1	0,792	53 ms.				
	2	0,789	50 ms.	0.824	320 ms.	4%	79.2 ms
xTionbox	1	0.886	55 ms.				
	2	0.812	49 ms.	0.902	318 ms.	2%	176.1 ms.

In the case of the study presented the Cloud has not been considered. This is because this level manage large amount of complex data and consequently, computation is expeensive in time. Cloud is used, mostly, to store new patterns or update existing. The possibility of use Cloud as a repository, allow smart resources to have more detection power and adapt the patterns used locally to different environments. Cloud allows smart resources to push the power limits of microprocessors computation and storage.

5. Conclusions

The combination of sensors, micro-controllers and communications allows cities to implement intelligent distributed systems. Because of the large number and variety of sensors existing in smart cities, it is convenient to organize them into devices that can interact with each other. In this paper has been presented how considering services as a communication method in a smart device, it allows to integrate information from different sensors. The experiments carried out to verify the integration of the information, increase notably the success in the detection of an object.

Based on the results, it is possible to apply information integration in smart cities as a method to improve the services offered by the different elements. Based on the experiments carried out, it is convenient to test how the smart resources employees detect other objects. The paper shows the experiments performed with a system with two smart resources detecting two different objects, adding more objects and smart resources it is possible to study the cost in workload to recognise an

Peer-reviewed version available at Sensors 2019, 20, 112; doi:10.3390/s20010112

environment. Distributing the objects characteristics to be recognised, it is possible to balance the workload to use an optimal amount of system resources, such as a processing time or communications load.

Funding: This research was funded by the Spanish Science and Innovation Ministry grant number MICINN: CICYT project PRECON-I4: "Predictable and dependable computer systems for Industry 4.0" TIN2017-86520-C3-1-R.

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

Abbreviations

The following abbreviations are used in this manuscript:

DDS	Data Distribution Service
IoT	Internet of Things
RAMI	Reference Architectural Model Industrie 4.0
SR	Smart Resource

References

1. Munera, E.; Poza-Lujan, J.L.; Posadas-Yagüe, J.L.; Simó-Ten, J.E.; Noguera, J. Dynamic reconfiguration of a rgbd sensor based on qos and qoc requirements in distributed systems. *Sensors* **2015**, *15*, 18080–18101.
2. Roscia, M.; Longo, M.; Lazaroiu, G.C. Smart City by multi-agent systems. 2013 International Conference on Renewable Energy Research and Applications (ICRERA). IEEE, 2013, pp. 371–376.
3. Ström, D.P.; Nenci, F.; Stachniss, C. Predictive exploration considering previously mapped environments. Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015, pp. 2761–2766.
4. Zdraveski, V.; Mishev, K.; Trajanov, D.; Kocarev, L. ISO-standardized smart city platform architecture and dashboard. *IEEE Pervasive Computing* **2017**, *16*, 35–43.
5. Hankel, M.; Rexroth, B. The reference architectural model industrie 4.0 (rami 4.0). *ZVEI, April* **2015**.
6. Dastjerdi, A.V.; Buyya, R. Fog computing: Helping the Internet of Things realize its potential. *Computer* **2016**, *49*, 112–116.
7. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of things for smart cities. *IEEE Internet of Things journal* **2014**, *1*, 22–32.
8. Cao, J.; Song, C.; Peng, S.; Xiao, F.; Song, S. Improved traffic sign detection and recognition algorithm for intelligent vehicles. *Sensors* **2019**, *19*, 4021.
9. García, C.G.; Meana-Llorián, D.; G-Bustelo, B.C.P.; Lovelle, J.M.C.; Garcia-Fernandez, N. Midgar: Detection of people through computer vision in the Internet of Things scenarios to improve the security in Smart Cities, Smart Towns, and Smart Homes. *Future Generation Computer Systems* **2017**, *76*, 301–313.
10. Guerrero-Gómez-Olmedo, R.; López-Sastre, R.J.; Maldonado-Bascón, S.; Fernández-Caballero, A. Vehicle tracking by simultaneous detection and viewpoint estimation. International Work-Conference on the Interplay Between Natural and Artificial Computation. Springer, 2013, pp. 306–316.
11. Lim, G.H.; Suh, I.H.; Suh, H. Ontology-based unified robot knowledge for service robots in indoor environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **2011**, *41*, 492–509.
12. Zhang, J. Multi-source remote sensing data fusion: status and trends. *International Journal of Image and Data Fusion* **2010**, *1*, 5–24.
13. Deng, X.; Jiang, Y.; Yang, L.T.; Lin, M.; Yi, L.; Wang, M. Data fusion based coverage optimization in heterogeneous sensor networks: A survey. *Information Fusion* **2019**, *52*, 90 – 105. doi:https://doi.org/10.1016/j.inffus.2018.11.020.
14. Azim, A.; Aycard, O. Detection, classification and tracking of moving objects in a 3D environment. Intelligent Vehicles Symposium (IV), 2012 IEEE. IEEE, 2012, pp. 802–807.
15. Jain, A.K.; Duin, R.P.; Mao, J. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence* **2000**, *22*, 4–37.

16. Yurish, S.Y. Sensors: smart vs. intelligent. *Sensors & transducers* **2010**, *114*, I.
17. Poslad, S. *Ubiquitous computing: smart devices, environments and interactions*; John Wiley & Sons, 2011.
18. Hancke, G.P.; Hancke Jr, G.P.; others. The role of advanced sensing in smart cities. *Sensors* **2012**, *13*, 393–425.
19. Lazar, A.; Koehler, C.; Tanenbaum, J.; Nguyen, D.H. Why we use and abandon smart devices. Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2015, pp. 635–646.
20. Chen, Y. Industrial information integration—A literature review 2006–2015. *Journal of Industrial Information Integration* **2016**, *2*, 30–64.
21. Rincon, J.; Poza-Lujan, J.L.; Julian, V.; Posadas-Yagüe, J.L.; Carrascosa, C. Extending MAM5 meta-model and JaCalIV E framework to integrate smart devices from real environments. *PloS one* **2016**, *11*, e0149665.
22. Eugster, P.T.; Felber, P.A.; Guerraoui, R.; Kermarrec, A.M. The many faces of publish/subscribe. *ACM computing surveys (CSUR)* **2003**, *35*, 114–131.
23. Simó-Ten, J.E.; Munera, E.; Poza-Lujan, J.L.; Posadas-Yagüe, J.L.; Blanes, F. CKMultipeer: Connecting Devices Without Caring about the Network. International Symposium on Distributed Computing and Artificial Intelligence. Springer, 2017, pp. 189–196.
24. Tijero, H.P.; Gutiérrez, J.J. Criticality Distributed Systems through the DDS Standard. *Revista Iberoamericana de Automática e Informática industrial* **2018**, *15*, 439–447. doi:doi.org/10.4995/riai.2017.9000.
25. Poza, J.L.; Posadas, J.L.; Simó, J.E. From the queue to the quality of service policy: A middleware implementation. International Work-Conference on Artificial Neural Networks. Springer, 2009, pp. 432–437.
26. Amurrio, A.; Azketa, E.; Javier Gutierrez, J.; Aldea, M.; Parra, J. A review on optimization techniques for the deployment and scheduling of distributed real-time systems. *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial* **2019**, *16*, 249–263.
27. Munera, E.; Poza-Lujan, J.L.; Posadas-Yagüe, J.L.; Simó-Ten, J.E.; Noguera, J.F.B. Dynamic Reconfiguration of a RGBD Sensor Based on QoS and QoC Requirements in Distributed Systems. *Sensors* **2015**, *15*, 18080–18101. doi:10.3390/s150818080.
28. Garage, W. Turtlebot. *Website: http://turtlebot. com/last visited* **2011**, pp. 11–25.
29. Coley, G. Beaglebone black system reference manual. *Texas Instruments, Dallas* **2013**.
30. Munera Sánchez, E.; Muñoz Alcobendas, M.; Blanes Noguera, J.; Benet Gilabert, G.; Simó Ten, J. A reliability-based particle filter for humanoid robot self-localization in RoboCup Standard Platform League. *Sensors* **2013**, *13*, 14954–14983.
31. Chow, J.; Lichti, D.; Hol, J.; Bellusci, G.; Luinge, H. Imu and multiple RGB-D camera fusion for assisting indoor stop-and-go 3D terrestrial laser scanning. *Robotics* **2014**, *3*, 247–280.
32. Rogers III, J.G.; Nieto-Granda, C.; Christensen, H.I. Coordination strategies for multi-robot exploration and mapping. Experimental Robotics. Springer, 2013, pp. 231–243.
33. Chen, L.; Wei, H.; Ferryman, J. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters* **2013**, *34*, 1995–2006.
34. Chianese, A.; Piccialli, F.; Riccio, G. Designing a smart multisensor framework based on beaglebone black board. In *Computer Science and its Applications*; Springer, 2015; pp. 391–397.