Research Article

International Journal of Distributed Sensor Networks

International Journal of Distributed Sensor Networks
2018, Vol. 14(3)

DOI: 10.1177/1550147718763550
journals.sagepub.com/home/dsn
SAGE

A flexible human detection service suitable for Intelligent Spaces based on a multi-camera network

Douglas Almonfrey[1], Alexandre Pereira do Carmo[1], Felippe Mendonça de Queiroz[2], Rodolfo Picoreti[2], Raquel Frizera Vassallo[2] and Evandro Ottoni Teatini Salles[2]

Abstract

The research field of the Intelligent Spaces has experienced increasing attention in the last decade. As an instance of the ubiquitous computing paradigm, the general idea is to extract information from the ambient and use it to interact and provide services to the actors present in the environment. The sensory analysis is mandatory in this area and humans are usually the principal actors involved. In this sense, we propose a human detector to be used in an Intelligent Space based on a multi-camera network. Our human detector is implemented in the same paradigm of our Intelligent Space. As a contribution of the present work, the human detector is designed to be a service that is scalable, reliable and parallelizable. It is also a concern of our service to be flexible, less structured as possible, attending different Intelligent Space applications and services, as well as their requirements. As it can be found in different everyday environments, a multi-camera system is used to overcome some difficulties traditionally faced by existing human detection approaches. To validate our approach, we implement three different applications that are proof of concept of many day-to-day real tasks. Two of these applications involve human–robot interaction. With respect to time and detection performance requirements, our human detection service has proved to be suitable for interacting with the other services of our Intelligent Space, in order to successfully complete the tasks of each application.

Keywords

Human detection, Intelligent Spaces, multi-camera systems, ubiquitous computing, robotics applications

Date received: 10 November 2017; accepted: 12 February 2018

Handling Editor: Janez Perš

Introduction

Back in the 90s, Weiser[1] suggested that the technology of the future would be so immersed in people's lives that it would be unnoticeable. At that time, Intelligent Spaces still could not be implemented, although the personal computers were already being produced and used. Weiser[2] stated that computers were not the ideal tool yet, and that a good tool should be invisible. Computing should be used in a way that the user would not notice it or have technical knowledge about it. By this, Weiser defined what is called ubiquitous computing.

Studies in ubiquitous computing have resulted in the current concepts of Intelligent Spaces, Smart Spaces or Ambient Intelligence.[3–5] Although there are different definitions, all of them consider that computing is immersed in the environment. Thus, since we do not

[1]Instituto Federal do Espírito Santo, Vitória, Brasil
[2]Universidade Federal do Espírito Santo, Vitória, Brasil

Corresponding author:
Douglas Almonfrey, Instituto Federal do Espírito Santo, Av. Vitória, 1729, Vitória 29075-910, Espírito Santo, Brasil.
Email: dalmonfrey@ifes.edu.br

aim to discuss the specific features and differences between the existing definitions, we will use Intelligent Space as the general term in this article.

Nowadays, Intelligent Spaces experience increasing attention, always chasing the idea of transferring more intelligence and computing to the environment, and thus reducing the need for devices with high computational capacity and processing. Usually, the objective is to extract information from the environment, perform all the processing needed, and use it to interact and provide services to human beings.[6] The developed works vary from implementing smart houses and meeting rooms[7,8] to surveillance and industrial applications that also demand knowledge of the ambient to analyze events[9] or to control automatic processes in a warehouse.[10] Many of these applications in Intelligent Spaces need to detect people in order to promote human–machine or human–robot interaction, to offer services or even to infer contexts by observing human motion.[11–15]

Therefore, it is common to see multi-sensor networks being used to extract useful information from the environment and detect human presence.[14,16] Some works avoid using cameras due to the complexity of the algorithms involved and the problems with occlusion, color, and brightness variations, in addition to changes on objects appearance due to the adopted viewpoint. In these cases, sensors such as lasers, ultrasounds, motion sensors, and light cameras are usually used.[17–20]

However, when it comes to direct interaction with humans, not only presence detection is needed, but also a more precise estimation of people's position, and sometimes face or gesture recognition. Multi-camera networks are then preferred, since cameras provide a rich source of information about the environment and the presence of objects in the scene.[21,22]

In the specific case of human detection, most of the methods are model-based and only evaluated in few public datasets, after hours or days of hyperparameter tuning. As mentioned in Li et al.,[23] these kinds of detectors seek for a generic solution, trying to reduce error as much as possible, and, with very few exceptions, only offline tests are conducted.

Despite the progress on object detection leaded by the region over convolutional neural network (R-CNN) family,[24] the effectiveness of object and people detection in real-time applications, as demanded by Intelligent Spaces, is still an unsolved problem. If performing people detection with one camera already demands high computational capacity, performing this with a network of cameras is usually more difficult. Therefore, regarding the implementation of human detection in Intelligent Spaces with cameras as main sensors in the environment, one of the current issues is to develop a light detector that does not overload the network, saving computational resources for other services or applications.

In addition to that, if the Intelligent Space is implemented using an infrastructure based on cloud computing and service-oriented architecture (SOA) platform, it will present features as parallelism, reliability and dynamic allocation of computational resources. Thus, the human detectors developed for this type of platform can benefit from the advantages offered by the multi-camera network and the features available due to the Intelligent Space architecture. Human detection should be provided as a service that may be used by different applications and run in different types of infrastructures, being at the same time light and flexible.

With this in mind, in this work, we propose a light and flexible service that provides human detection in an Intelligent Space based on a multi-camera network, which architecture uses cloud computing and SOA. Visual information from the environment is provided using just the network of cameras, while a simple human detector is offered as a service to different applications. Besides performing human detection, the proposed service aggregates and transfers important properties of the Intelligent Space and the multi-camera network to the applications, such as parallelism and scalability. Many different applications, that are proof of concept (PoC) of some day-to-day real tasks and that rely on human detection, may benefit from this other features to work faster and in a more flexible way.

It is important to mention that the focus of the present work is not to propose a generic state of the art human detection system as Zhang et al.[25] We also do not aim to provide just human position information for autonomous robots as can be seen in Ribeiro et al.[26] or Lee et al.[21] But as mentioned before, our human detection service provides two different features: (1) a service with properties as flexibility, scalability, parallelism, and reliability and also (2) human location, given by a multi-camera network, that allows us to avoid some problems that make single camera human detectors fail in real world applications. The interrelationship of the human detection service and the camera network is handled to meet the time requirements of the applications and real-time execution is achieved.

Regarding real-time, according to ATIS telecom glossary,[27] the distinction between real-time and near real-time is somewhat nebulous and must be defined for the situation at hand. Thus, in the context of this work, we adopt the term real-time as the achievement of data processing rates which do not insert perceived delays in the time requirements of the applications. In this case, the experience of the users in the interaction with the infrastructure of the Intelligent Space will not be affected by our data processing techniques.

One example is the response time of the robot in relation to the movement of the human in the applications proposed in this work. Ideally, the response time cannot be affected by the processing time of the human detector.

Therefore, what we consider as the main contributions of this work are as follows:

- A human detection service which, due to the fact that is designed using concepts as cloud computing and SOA, is suitable for being used by different applications of Intelligent Spaces based on multi-camera network, and which requirements can be fulfilled by the used human detector;
- A human detection service designed to be scalable, reliable, and parallelizable in order to meet time and computational capacity demands of the applications. All these features are supported by the architecture and software infrastructure of the Intelligent Space and transferred to the applications through the proposed service;
- Validation of the proposed service by implementing three different PoC real-time applications, deployed in a distributed manner in our experimental Intelligent Space. Two of these applications involve human–robot interaction.

Besides, we consider as marginal contributions and features of our system:

- The three-dimensional (3D) localization of individuals in the working space with an average error smaller than 40% of the average diameter of the human body area on the ground plane. This feature can also be achieved due to the use of a multi-camera network;
- An extensive analysis of the accuracy of the human detector as a context information extraction service for real world applications;
- Release of more than 6000 annotated images for comparison.

This article is organized as follows. In section "Related work," a brief review of the literature is presented, and in section "The Intelligent Space architecture," some details of the architecture of the Intelligent Space, used in this work, are described. Section "The human detection service" discusses the pipeline of the human detection service. In section "Intelligent Space applications," the applications implemented in the Intelligent Space are described, and in section "Experiments," the experiments, their methodology, and results are presented and analyzed. Finally, section "Conclusion" presents the conclusions of this work.

Related work

Human detection

Human detection is an active research area. Even with the success of general object detection,[24,28–30] human detection has been treated as an exclusive field of interest. This is mainly because humans are key components for many current applications, such as driver assistance and intelligent surveillance systems.[25] Besides, human detectors present specific problems when treating discrimination from the background.[31]

The most studied instance of human detection is the pedestrian detection area. Many solutions were already proposed in the literature.[25,32–36] Despite the fact that deep convolutional neural network (DCNN) has pushed the state of the art of pedestrian detection, even the best trained generic detectors have poor performance when evaluating across datasets testing scenario.[37] This indicates that in some cases, the use of additional scene information can still be unavoidable.[23]

In this work, we use a camera network and aggregate different concepts as homography and image segmentation to a light pedestrian detector model. Our model is called independent components channel features (ICCF) and was first presented in Almonfrey et al.[38] Together with aggregate channel features (ACF) detector,[32] it is the basis of the human detection service offered by the Intelligent Space used as our testbed. ICCF and ACF (10^4 parameters to be learned) are four orders of magnitude smaller than VGG16[39] (10^8 parameters to be learned), the most used DCNN model of the literature and present in the major part of state of the art pedestrian detectors.

As already discussed in the previous section, even when the Intelligent Space infrastructure supports high computational capacity, it is interesting to have light services and applications so they do not consume all the resources. Thus, whenever is possible, implementing a human detector that does not use graphical processing unit (GPU) is also desirable. Considering that not every workstation in many Intelligent Space infrastructures have a GPU available, it is usually difficult to allocate and distribute services that are GPU dependent in different processing nodes.

Therefore, the fact that we do not use GPU in our proposed human detection service, even for real applications, brings flexibility to our system. In Ribeiro et al.,[26] for example, differently from our work, the use of GPU is mandatory to perform human detection and to accomplish the demands of their application. Besides, their human detection method is not delivered as a service that can be flexibly used by different applications. The solution proposed in Ribeiro et al.[26] is not designed to be distributed over the nodes of the infrastructure; thus, it is not concerned with issues such as

synchronism, scalability and reliability. They use only one powerful node (GPU) to have a low false positive rate, while we use simple techniques and the redundant information provided by a network of cameras to achieve this goal. In this case, we have a light detector that can be distributed over the infrastructure.

It is important to mention that our Intelligent Space is not unable to receive nodes with a GPU in its infrastructure. However, we built a human detection service that is not GPU dependent and that is suitable for many applications, using the power of a multi-camera network and also avoiding the use of computationally expensive solutions.

Human detection in Intelligent Spaces

In this section, we will discuss some works on Intelligent Spaces that perform human detection using sensor networks. Some different sensors may be addressed, but we will focus on works that mainly use visual sensors or their combination with a few others. The idea is to highlight the differences between these works and the one presented in this article.

In Surie et al.,[18] a wall mounted Kinect is used to detect humans using skeleton and face recognition, extracted from images and depth fields. In our case, the implemented human detection service is model based and relies only in red-green-blue (RGB) images, while no rangefinder is used to obtain the 3D information. Surie et al.[18] also mention that their workspace is restricted because of the limited field of view of the kinect's camera, requiring a more structured setup. On the other hand, our work benefits from different points of view provided by the network of cameras, identifying humans with various poses in relation to these sensors. The only prior setup required is the calibration of the camera.

Glas et al.[11] developed a "network robot system" to deploy social robots in practical applications as shopping malls and other commercial spaces. In the referred work, applications of ambient intelligence and human–machine interaction are implemented using data obtained from sensors installed in the environment. The sensors are laser rangefinders and a human supervisor is also employed when the system faces difficulties in recognition and planning tasks. In our case, we do not use any human supervisor and our system is built using a philosophy of no human intervention during the offer of services. Also, since our work uses a network of cameras, which are usually already installed in most commercial buildings, it would require less physical modifications to be used, compared to installing lasers in the environment. Besides that, in Glas et al.,[11] the network of robots is the only provided service. In our system, robots (actuators) and cameras (sensors) are treated as entities, while human detection is just one of the many services provided to the applications. Therefore, robots can also be added to the sensors network if needed.

A robotic transportation system for shopping assistance is developed in Matsuhira et al.[40] Differently from our work, the focus is not on a system with distributed services. The application seems to work fine, but the human detection approach is simpler than ours and depends on the fusion of video and laser rangefinder sensors. In spite of using a set of cameras, 3D information is not fully explored as in our work, where the calibration and homography between cameras are used.

The aim in Albawendi et al.[41] is to investigate a low cost and acceptable visual camera monitoring system to the elderly. The idea is to limit the amount of information transmitted from the visual sensor, reducing the level of intrusion of the camera in the routine of the elderly. Comparing to our work, their system is more an application and their object detection is too much dependent on background subtraction (BS), requiring a more structured workspace and thus being more restrictive than ours.

In Adduci et al.,[42] a multi-camera tracking application is built. However, according to the authors, their application, that relies on point cloud estimation, demands hardware and software modifications to be prepared for real-time tasks. Our human and robot detection services are adequate for real-time applications just as a service for controlling the robot. Differently from the previous mentioned works, we are concerned that our cloud services meet time and reliability requirements of the applications.

As a last comparison, Lee et al.[21] develop a vision-based method for human and robot localization in order to be used in an active information display service in an Intelligent Space. For this, they configure a multi-camera network to estimate the 3D positions of a human and a robot, which are detected using the histogram of oriented gradients (HOG) feature method. One of the main differences from our work is that their system is not based on a service architecture, having a fixed configuration. Although they used several distributed intelligent network devices (DINDs), which consists on a camera and a network device, each DIND is connected to a desktop and acts as an individual processing node that may communicate to others through the local network. Thus, no services, as human and robot detection or robot control, are provided by their system architecture. Also to estimate the 3D position of a person, it is necessary to match at least two detections provided by different cameras, so their system can perform triangulation. Their localization function may experience some problems if many people are present in the workspace, since having two detections of the same person is mandatory. In our case, human and robot detection can be performed even if just one camera is

able to observe the robot and the person. Because we assume that humans and robots are always standing on the room's floor, our system can estimate 3D positions with just one image. Additional detections are used to eliminate false positives that may occur and to improve the estimated 3D location. Finally, their environment is a lot more structured them ours, having no objects in the workspace defined for the experiments and a light and homogeneous background, which facilitates detections of people and robot.

Complementary remarks

Besides the differences among our work and the ones highlighted previously in this section, to the best of our knowledge, we did not find any other work that presents a human detection service like ours, considering the aspects of its implementation and usability.

Some related work may present human detectors with remarkable performances on specific benchmarks, but it is important to mention that we do not aim to propose a state of the art detector on a specific dataset. We are interested in presenting a light and flexible service that provides human detection in a distributed way over the infrastructure to different applications deployed in an Intelligent Space. Therefore, choosing a simpler human detector allows us to provide a light service that works in real-time and in a more flexible way, since it can be used by different applications. It also enables the usage of regular computational nodes and not just powerful computers or machines that include GPUs.

Additionally, parallelism and scalability can be found in some works, usually when a multi-core processor or a GPU is used to implement the detector, in only one computational node. In our work, we are concerned with designing a service that can be distributed in different nodes among the Intelligent Space infrastructure, according to the availability and computational capacity of the nodes. Whenever needed, the human detection service can be instantiated in many nodes and even restarted in new ones if some of the current nodes drop off. This approach allows us to use more efficiently the available nodes in the Intelligent Space, no matter what is the computational capacity of each one.

The Intelligent Space architecture

An Intelligent Space[4,5] can be described as an interactive environment equipped with a network of sensors (e.g. cameras, microphones, ultrasound), able to gather information about the surroundings, and a network of actuators (e.g. robots, mobile devices, information screens), which can be directly controlled by different computing services to act or modify the environment. Besides controlling the actuators, computing services can also analyze the gathered information in order to support decision-making and task execution.

Sensors, actuators and computing services are underpinned by a software infrastructure in charge of providing communication facilities and abstractions needed. Services and resources from a specific device (sensor or actuator) can then be accessed and used by different entities such as other services, applications, and even other devices.

As mentioned before, our Intelligent Space is based on computer vision. Therefore, it is instrumented with a network of IP cameras capable of capturing digital images and videos, as shown in Figure 1. The system is also able to control actuators like a robot. The cameras are used as the main sensors of the environment, similarly to the project described in Rampinelli et al.[43] In order to acquire high-level understanding of the environment, we designed a software infrastructure to tackle the task of processing and analyzing data extracted from the distributed cameras in real-time.

The software infrastructure of our Intelligent Space is conceived as a development platform, that is, as a platform as a service (PaaS). Therefore, application developers can use different computing services, even if some of them were initially designed to be used by a specific application. That is why these services should be flexible enough to, at the same time, meet the applications' requirements and provide a high-level programming abstraction for the developers. The SOA model was applied in the design of such software infrastructure in order to provide the necessary programmability and re-usability on service level. These features make building and deploying real-time applications easier for developers, as well as allowing applications to integrate new services to the platform.

Also, our platform is deployed on top of a cloud infrastructure like infrastructure as a service (IaaS) to meet specific requirements of computer vision applications such as low latency, large bandwidth, and high processing capacity. The programmability of this IaaS enables the platform to meet the stringent requirements, mainly for real-time applications.[44]

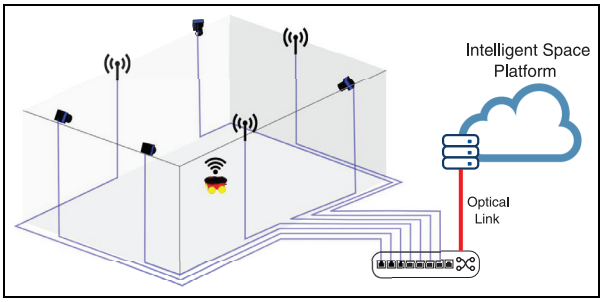

Figure 1. Concept of Intelligent Space (IS).

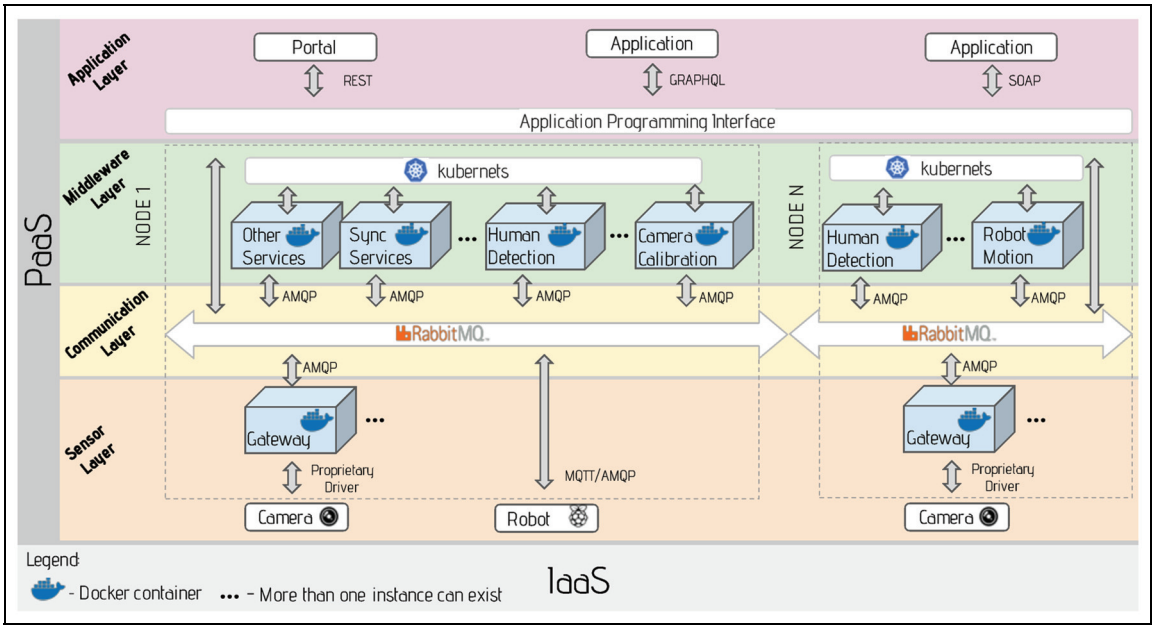

Figure 2. Intelligent Space architecture.

Figure 2 shows the platform architecture used in our Intelligent Space prototype in order to support computer vision applications. This platform has four layers: sensing, communication, middleware, and application, which are going to be described in the following subsections.

Sensing layer

The sensing layer is responsible for exposing the resources of the physical domain to the digital domain. In addition, this layer should simplify communication among heterogeneous devices through standardized interfaces.

For that, the sensing layer acquires information and controls the objects' behavior in the physical space. As can be noted from Figure 2, the physical entities of this layer are sensors and actuators. Each physical entity in the real world is represented in the digital domain as a virtual entity. The virtual entity is associated with resources that allow interaction, through services, with the physical entity that it represents.

Although it is possible to directly integrate equipments into the platform, the standardization function for these equipments is usually performed by gateways. They are responsible for translating the specific protocol of the device to the standard interface provided by the virtual entity. In the same way, it also performs the conversion of the data to a standard format.

Communication layer

The communication layer is responsible for routing and forwarding messages, time decoupling and monitoring services. To perform these functions, all the communications between the system components go through a RabbitMQ message broker, as shown in Figure 2. Since all the messages go through the broker, it is possible to monitor the flow of messages and even modify them before being forwarded.

The main advantage of having a platform that uses a broker to communicate among its elements is that entities only need to know their address to communicate with each other. That makes the development of services and applications much more easier. Moreover, it offers the possibility of message persistence. If the recipient is not available, the message is stored until it becomes available. On the other hand, the problem of this approach is that the broker becomes the single point of failure and the performance bottleneck. To minimize the problem of converging message flow to a single point, the architecture provides the possibility of using federated brokers working together at different points in the network.[45]

Middleware layer

The middleware provides service interfaces for applications allowing users to get rid of the not so trivial details of the other layers. This layer is composed by several services that run functions to support the infrastructure, besides specific services of computer vision, which can be used by many applications (Figure 2).

Two important support services provided by the Intelligent Space platform are the synchronization service and the cameras calibration service:

- Synchronization service: Usually, synchronization problems are faced in computer vision

applications that require synchronized images from two or more cameras in real-time. This requirement is important to avoid inconsistency between the data obtained from the images, since, without synchronism, there is no guarantee that the images correspond to the same scene at the same time. The synchronization service periodically monitors and applies delays for capturing the cameras' images and, if needed, data from other connected devices, in order to maintain a maximum error of synchronism acceptable to the application being developed.

- Cameras calibration service: This is a semi-automatic process already implemented as a service in our Intelligent Space. This service returns all the intrinsic and extrinsic parameters of the installed cameras after capturing and processing images of a pattern, held and moved manually. Once the calibration is done, there is no need to recalibrate the cameras, unless one of them is moved from its position. We consider this service as semi-automatic because there is still a human intervention on the process, which is holding and moving the calibration pattern.

Among the specific services deployed for applications, we are going to focus on the ones used to develop the tasks presented in this article. The main service addressed is the human detection service, which is detailed and described in section "The human detection service." Additional services such as the ones related to robot motion, tracking a pattern, filtering, and other processes are discussed in section "Intelligent Space applications." Also in that section, the interrelationship between services and applications is highlighted.

Every service in the platform is virtualized in a container. Service virtualization through containers enables the development of loosely coupled services that are deployed independently. Many of these services can be connected in a service function chaining (SFC) to form an application. Containers virtualization aims to isolate the services and ensure their independence, as long as the interfaces are maintained. These containers can be easily shared, deployed, updated, and scaled instantly and independently of the other services that constitute the application.

Docker[46] is used in our Intelligent Space platform as the container technology, due to its ease of building services in containers. Moreover, Docker virtualization is lightweight and, because of that, multiple applications may use services at the same time, in the same physical or virtual server. This scenario enables an orchestration to provide the right amount of resources to containers at the right time, allowing better allocation of the containers in the cloud infrastructure. The orchestration of Docker containers in our platform is performed by Kubernetes,[47] which functions include automate deploying, scaling, and operating application containers.

Application layer

The last layer of Figure 2 is the application layer that exposes high-level services in the form of an application programming interface (API) for developers that may interact with the platform. This API allows the development in different programming languages, leaving transparent the access to the services and equipments of the platform.

Just a few services are made available to developers, usually the ones needed to deploy the applications, such as detecting and tracking marks or people. Other services, such as the ones that support the platform or perform resource orchestration, are hidden since the final user should not need to worry about these functionalities. The principal aim is to make the platform as transparent as possible for the users.

Architectural requirements

Our platform was designed to abstract the complexities of the system or hardware, allowing the application developer to focus all his effort on the task to be solved. Since it is designed for computer vision applications, the platform has a number of features in order to meet the specific requirements for this application domain.

- Scalability: The platform needs to be scalable to meet the increasing demand for application resources. For a better management of resources, the services that comprise an application must be simple, have low coupling, and must be implemented in a way that allows its reuse by other applications. In addition, stateless and parallelized services allow the platform to raise new service instances for different applications.
- Real-time: The platform must provide real-time services when the correctness of an operation depends not only on its logical correctness but also on the time in which it is performed. As computer vision applications may deal with many real-time tasks (e.g. detecting and tracking agents for moving, carrying or even health care issues), delivering information or services on-time for those applications is critical. Delayed information or services in such applications can make the system useless and even dangerous. For that, a synchronism service and a strictly monitoring service are primordial.
- Reliability: Applications should remain operational while a task is being executed, even in the presence of failures. Each service that composes

the application and supports other services needs to be reliable itself to achieve overall reliability. The used Intelligent Space platform has mechanisms that allow the distribution, throughout the infrastructure, of both management services and specific services used by the applications.

To achieve these features, we designed a system architecture that allows us to scale services both vertically and horizontally. The vertical scaling allows the increase of the computational power for the same instance of a service, while horizontal scaling allows the increase of the number of instances, of the same service, to handle an increase in demand.

To illustrate that we describe what happens if the number of cameras in the Intelligent Space increases. According to our implementation, the human detection process is divided into smaller services, which are going to be better explained and detailed in the next section. Just to resume, as a first step, a service consisting of a simple human detector runs on each image plane. Thus, for each camera in the environment, an instance of such service is started. That makes as many instances as the number of cameras, and all of them run in parallel.

Each instance of this simple human detector provides a set of detections to a following service that filters and merges the information received from all of them. If the number of cameras is increased, the system will perform a horizontal escalation of the service, raising the number of instances to the maximum that a physical node can deal with. Once this limit is reached, the system starts new instances of the service on a different computational node. This way, the large volume of traffic generated by the increase of cameras is distributed across the infrastructure and not concentrated in a single point, generating a traffic bottleneck and making the system operation unfeasible.

Regarding the reliability of our system architecture, it is provided by the replication controller implemented by Kubernetes.[47] This controller keeps the desired number of instances of a service running. If a node in the infrastructure becomes unavailable, the services that were running in the referred node are started in other node of the infrastructure. Thus, the fact that our human detector is implemented as a service, enables it to inherit all these important characteristics provided by the paradigm in which our Intelligent Space is inserted. The reliability is specially important, for example, to the bounding box (BB) Filtering and Robot Control services (to be explained in later sections), which availabilities are critical to the pipelines of our applications. Therefore, all applications to be developed using our architecture may benefit from these features, since they are deployed taking these peculiarities into account.

The human detection service

The human detection service developed in this work employs the ICCF and ACF detectors, members of the well-known filtered channel features family. This category of detectors was the state of the art of generic pedestrian detection for many years.[33,48] Channel features family is known for its fast response, low computational complexity, mainly due to the employment of decision trees. Decision trees allow fast rejection of negatives samples (not pedestrian), as can be seen in Dollár et al.[32]

Figure 3 shows the testing stage of the ICCF detector used in our human detection service. In step A, a pyramid of images is computed for multiscale detection. Then, during step B, for each scale, six HOG, one gradient magnitude, and LUV color channels are computed to compose the 10 channels called HOG + LUV in the literature. In step C, the HOG + LUV channels are diversified through convolution using "n" filters obtained from independent component analysis (ICA). Finally, the lexicographic version of the channels is used as features to classify image patches obtained from a sliding window approach. The NMS abbreviation stands for nonmaximum suppression that is a stage responsible for removing multiples detections of a single human. For additional information about this detector the reader can refer to Almonfrey et al.[38] The ACF and ICCF differ only in step C of Figure 3 which is not present in the ACF.

As mentioned before, even the best trained generic detectors have low performance when evaluating across datasets testing scenario.[37] This means that when the detector is not trained with data extracted from the ambient being analyzed, the accuracy tends to decrease. In this case, the fusion of different methodologies must be considered to build a functional approach for a real world task.

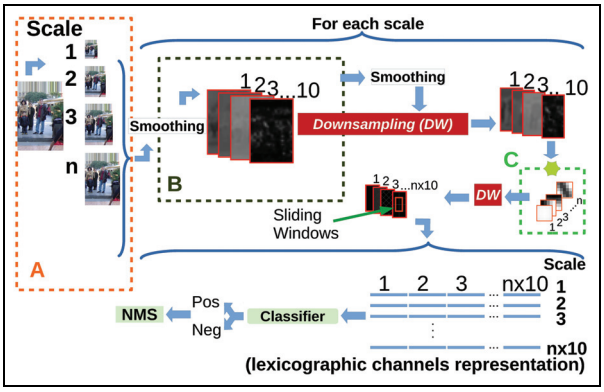

Figure 3. ICCF testing stage.

To deal with non ideal performance of detectors, it is also important to consider some prior information. Structural or geometric restrictions of the environment are good candidates. However, care must be taken when dealing with prior information. The system should not become too much configurable, because this implies in an overhead in the deployment stage, reducing the portability of the system to different environments.

Thus, with this in mind, the first prior information required by our human detection service is the calibration of the camera. As mentioned before, this process is implemented as a service in our Intelligent Space. Once calibration is done and because we consider all humans with their feet on the ground, we would be able to recover the 3D position of the humans detected, even if we had just only one camera. But since we are using a network of cameras, the calibration parameters also allow us to estimate the homography between the different images and, thus, transform human BBs found in one image to the other. With this information we can match simultaneous detections and reduce false positive indications. Even with this calibration restriction, our system is easily portable to different environments. Camera calibration is only needed to recover 3D and homography estimation. If services and applications do not require this kind of information, camera calibration can be avoided.

Figure 4 shows the human detection pipeline. Our human detection service (Figure 4(b)) is composed by two independent services which are the human localization and the BB filtering services. These two services are shown in Figure 4(c) and in stage E of Figure 4(b), respectively. In short, the human localization service uses the ACF and ICCF detectors besides a BS procedure in order to detect humans in the images. Finally, these detections are further refined by the BB filtering service using homography. In the following, the components of the human detection service are described in detail.

In stage A (Figure 4(a)), camera devices provide images (S_x, x = 1, ..., N) by means of camera gateways. In the present work, as four camera devices are used, N = 4. However, an arbitrary number N of gateways can be instantiated if more camera devices are provided. These gateways are completely independent and can be distributed over the workstations present in our infrastructure. In Figure 4(c), from stage B to D, the human localization process occurs. Each stream of processing is instantiated as an independent service. The human localization service takes an image (S_x) of a camera and delivery a set of BBs BB_x. In stage B, the ACF detector is employed because it is very fast (30 frames per second (FPS)) and it can recover a respectable amount of human BBs of the image.[32] However, some nonhuman BBs are also returned. To

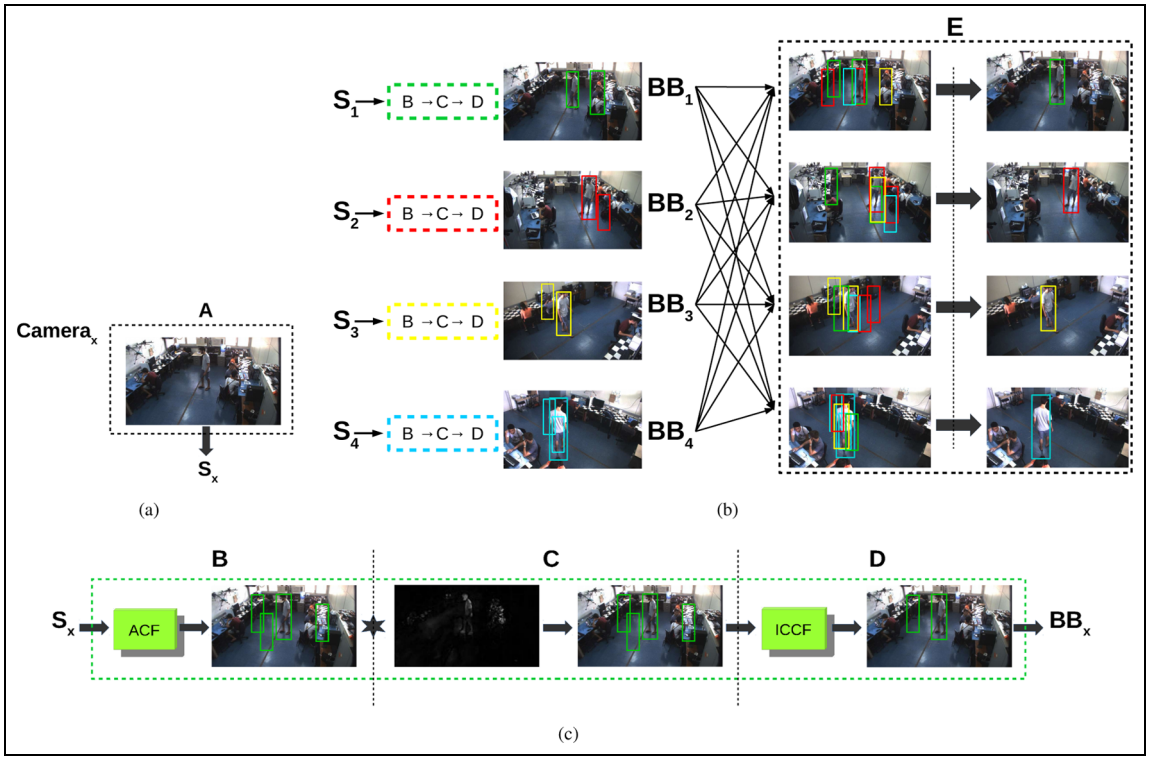

Figure 4. Human detection pipeline: (a) camera gateway, (b) human detection service, and (c) human localization service.

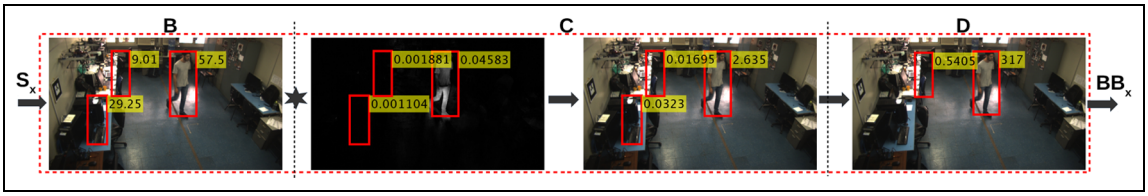

Figure 5. Human localization service.

remove this nonhuman samples, we employ a cascade of methods just in the reduced set of image patches returned by ACF. This choice makes the cascade detection not computationally prohibitive.

BS is a common method employed in computer vision area. However, it is very sensitive to illumination changes. Because of this, we use it with caution. In our case, BS is used in stage C just to weight the detection scores of each BB returned by ACF in stage B of Figure 4(c). The scores can be viewed as the reliability degree of a BB being a human. Equation (1) demonstrates the weighting process

$$Score_{new} = Score_{old} \times BB_{ProbBack}, \tag{1}$$

where $BB_{ProbBack}$ is the sum of pixels values inside the BB divided by the BB area, when analyzing the BS image. This procedure can be better understood by analyzing Figure 5. The score's value shown in the first image of stage C is the $BB_{ProbBack}$. Note that the BBs in the image of stage B that are related to black areas of BS image have the score decreased. Only the scores are changed and the shapes of the BBs are kept unchanged.

Thus, differently from Albawendi et al.,[41] where BS is crucial to detect spatial position of objects, we use it just to reduce the reliability of high score nonhuman BBs. Eventually, variations in the BS image, caused by abnormal illumination changes, could result in an increase in the scores of nonhuman BBs. However, this can be handled by other stages of negative filtering of our pipeline. This makes our system more robust to the influence of illumination changes in the BS process. After this processing, low score detections are discarded by a rejection threshold. The BS followed by a rejection threshold can be seen as another classifier. A semantic segmentation process could be used in the place of it when the camera system is not fixed, which is not the case of our work.

In stage D (Figure 4(c)), another round of nonhuman removal is performed by the ICCF detector. ICCF discriminates pedestrians from the background better than ACF, but in a lower FPS rate.[38] Because of this, it is only used in the reduced set of detections returned by ACF. The scores of the detections returned by ICCF are multiplied by the scores of the stage C, to compose strong confidence scores. Note that in stage D of Figure 5, one BB of the previous stage was eliminated by ICCF. Additionally, still in stage D, low score detections returned by ICCF are also discarded.

The last step of the human detection pipeline is the BB filtering service and it is shown in stage E of Figure 4(b). It employs homography to transform BBs between the images of the Intelligent Space. To model the acquisition geometry of the camera, we used the pinhole model, represented by the following equation

$$\lambda_i \tilde{\mathbf{m}}_\mathbf{i} = \mathbf{K_i} \Pi [\mathbf{R_i}, \mathbf{T_i}] \tilde{\mathbf{M}}, \tag{2}$$

where λ_i is a scale factor, $\tilde{\mathbf{m}}_\mathbf{i} = [u_i \; v_i \; 1]^T$ is a point in the ith camera image, $\mathbf{K_i}$ is the intrinsic matrix, Π is the projection matrix, and $[\mathbf{R_i}, \mathbf{T_i}]$ is the extrinsic matrix composed, respectively, by one rotation and translation. Finally, $\tilde{\mathbf{M}} = [x \; y \; z \; 1]^T$ is a 3D point in the world reference frame in which the cameras were calibrated, which generates the projections $\tilde{\mathbf{m}}_\mathbf{i}$ in the image plane. The variables with the $\sim$ sign are represented in homogeneous coordinates

It is possible to rewrite equation (2) as

$$\lambda_i \tilde{\mathbf{m}}_\mathbf{i} = \mathbf{A_i} [x \; y \; z \; 1]^T \tag{3}$$

representing the term $\mathbf{K_i} \Pi [\mathbf{R_i}, \mathbf{T_i}]$ by the matrix $\mathbf{A_i}$ of dimension 3×4, whose columns are indicated by $\mathbf{a_i^c}$. As we consider that the humans are always on the ground plane, that is, with $z = 0$, the incognitos of equation (3) are just x, y, and λ_i. Note that the matrix $\mathbf{A_i}$ is already known from the calibration and the point $\tilde{\mathbf{m}}_\mathbf{i}$ can be obtained from the image as the feet of the humans (approximately the midpoint of the BB's baseline). Consequently, the coordinates x, y in the world reference frame can be calculated using the following equation

$$\begin{bmatrix} \mathbf{a_i^1} & \mathbf{a_i^2} & -\tilde{\mathbf{m}}_\mathbf{i} \end{bmatrix} \begin{bmatrix} x \\ y \\ \lambda_i \end{bmatrix} = \begin{bmatrix} -z\mathbf{a_i^3} - \mathbf{a_i^4} \end{bmatrix}. \tag{4}$$

This is the way we recover the position of the detected humans in the 3D space.

Once the 3D information $[x \; y \; z]$ is known, we can use the homography $(\mathbf{H_j^i})$ between the ith and jth camera to match points $(\tilde{\mathbf{m}}_\mathbf{i})$ and $(\tilde{\mathbf{m}}_\mathbf{j})$ on its respective images, as shown in the following equation

$$\tilde{\mathbf{m}}_\mathbf{i} = \mathbf{H}^\mathbf{i}_\mathbf{j}\tilde{\mathbf{m}}_\mathbf{j}. \quad (5)$$

Using the homography, we can reproject a BB from one image to any other image of our camera system. If a BB present in one image does not have at least one corresponding BB in any other image, it is discarded. To compare BBs, we use the intersection over union (IoU) metric and a match is considered for an IoU greater than 0.5, as commonly employed in the pedestrian detection literature.[49]

It is worth to mention that all the images used to detect humans, during each sampling instant, are synchronized in order to guarantee that the same person, at the same time, is being detected in the different images. The synchronism between the cameras is provided by one of the support services, as mentioned before in section "Middleware layer."

At the end of the human detection pipeline, we expect that only standing humans are finally detected in the images. In the scope of this work, standing humans are considered those that need some service from the robots in the environment. It is important to mention that the human detection service can work even if just one camera is providing images. However, in this case, the human detection pipeline loses the strength and benefits obtained from the multi-camera network.

Intelligent Space applications

In this work, three PoC applications were developed to show the effectiveness of our human detection service in the context of the Intelligent Space architecture. Figure 6 illustrates the tasks executed in each application developed.

The first application is a human-following task performed by a robot (Figure 6(a)). The robot must follow a human that has just entered the room. This is a very common strategy, because humans that have just arrived in the room might need some service from the attendance robot.

Now, Figure 6(b) shows another task, where the robot has to navigate in the Intelligent Space deviating from humans present in the environment. That is important, for example, to help in the first task, where the robot must get to someone but should not hit other humans while it moves. These applications are PoC of many day-to-day real tasks that may be performed in environments where a multi-camera network is provided, such as banks, museums, shopping malls, and squares.

Figure 6(c) shows the cumulative occupancy map of the Intelligent Space. This map is referred as cumulative, because it has the objective of consolidating the most visited places in the room through time. Therefore, it is a temporal analysis. This is very useful, for example, to dynamically determine the best places to put advertisements or to determine the rental price of a room in a shopping mall. The most visited places are usually those in which the products or advertisements are more exposed and seen by consumers. Consequently, commercial rooms are comprehensively more expensive in these places.

Figure 7 shows the specific computer vision services that compose the applications and their interrelationship within the architecture of the Intelligent Space. Figures 2 and 7 are complementary. The first illustrates the messages flow between services and applications, while the latter presents services and applications distributed in the different layers of the architecture. Support services such as calibration service and synchronization service are not shown in the diagram of Figure 7 for better viewing.

The Pattern Tracker service is responsible to recover and publish the robot's 3D position, using a geometric pattern recognized by the multi-camera network. With this visual odometry, the robot can be controlled by the Robot Control service. The Human Localization and the Pattern Tracker services consume the frames published by the Camera gateways. The Human Localization service provides BBs to the BB Filtering service. The BB Frame Conversion service is responsible to project the BBs published by the BB Filtering service to the 3D coordinate of the world reference frame. Once the 3D information of the humans and the robot is available, the applications can be performed.

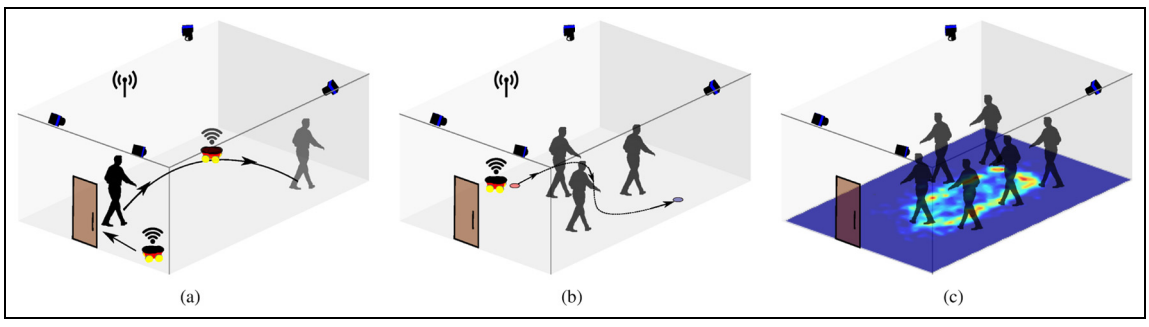

Figure 6. Applications of (a) human-following, (b) human-deviation, and (c) cumulative occupancy map.

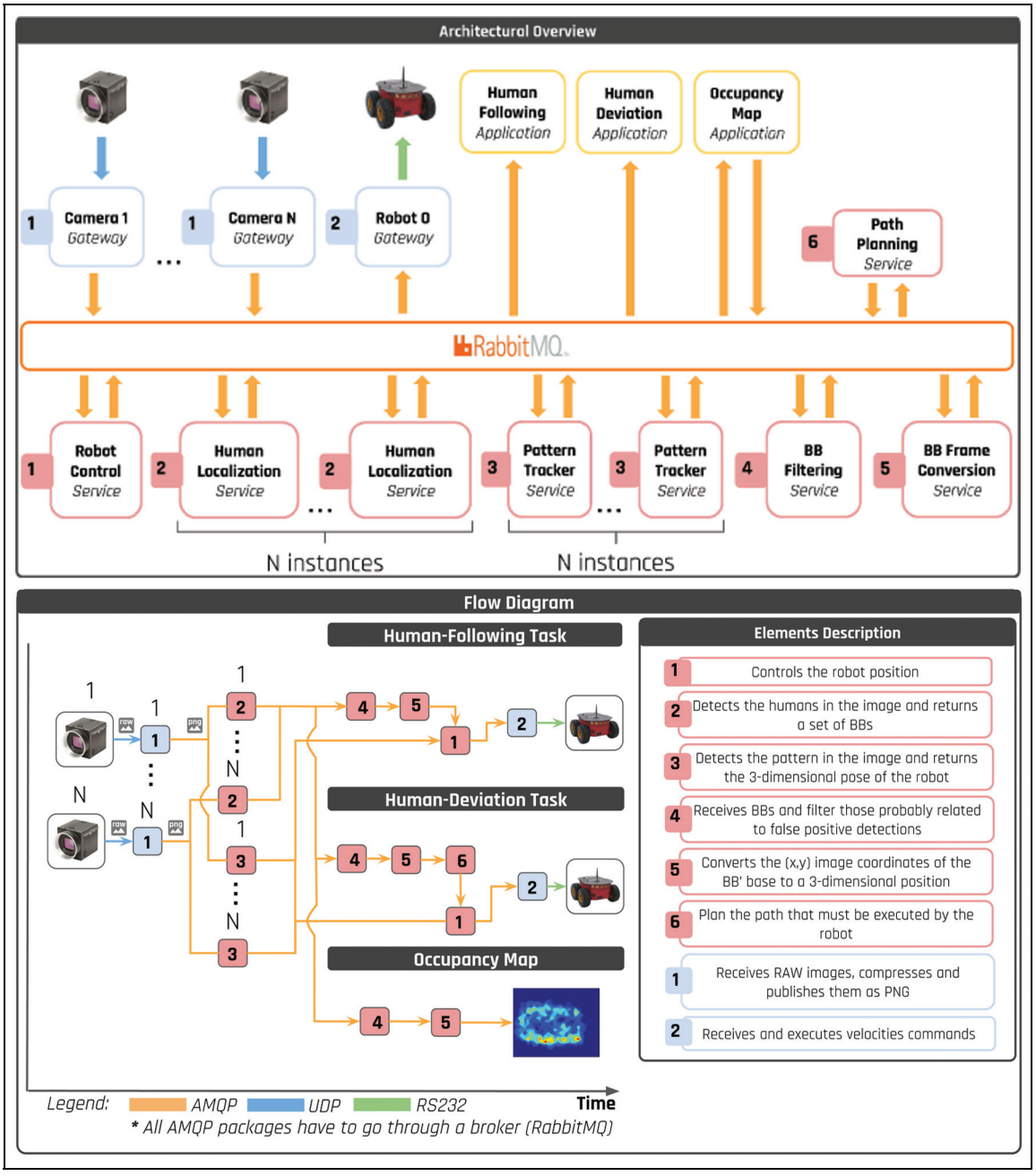

Figure 7. Interrelationship between services and applications.

The only application that does not use the Pattern Tracker service is the Occupancy Map. It is important to note that all the applications shown in Figure 7 are independent and may be executed at the same time in the Intelligent Space. Each flow of Figure 7 is just one occurrence of the main loop of each application. The flows are systematically repeated during the execution of the applications.

The flows originated from the Human Localization and Pattern Tracker services are asynchronous and executed in parallel. In the Human-Following and Human-Deviation applications, these flows converge to the Robot Control service, since they provide the information needed to execute the proposed applications. Each of the N instances of the Human Localization and the Pattern Tracker services are also executed in parallel. This shows the intrinsic parallelism of our system. This feature is of particular importance, due to the fact that human and object detectors are normally time-consuming tasks. In this case, these processes can be distributed over the infrastructure of our Intelligent Space during the test stage, using those nodes with more available resources.

Table 1. Intelligent Space infrastructure.

Device	Description processor/RAM memory
Node 1	i7-6850K/128 GB
Node 2	i5-3570/16 GB
Node 3	i5-4460S/8 GB
Node 4	E5504/4 GB
Cameras 1–4	Blackfly BFLY-PGE-09S2C
Robot	Pioneer P3-AT

Experiments

In this section, we demonstrate and validate the effectiveness of our human detection service in the context of the Intelligent Space architecture based on a multi-camera network. We would like to stress that we are considering that effectiveness is not only the property of detecting humans, but also the fulfillment of time requirements and the ability to provide properties as reliability and parallelism, inherited from the platform, to the applications.

Among others, the main contributions validated during the experiments are as follows:

- The effectiveness of the human detection service in meeting the demands of the applications through the use of an Intelligent Space based on a multi-camera network;
- The design of a human detection service able to cooperate with other services of the Intelligent Space architecture;
- The fulfillment of the requirements of real-time applications due to the suitable cooperation between the human detection and other services.

Materials and methods

Infrastructure. Table 1 briefly describes our infrastructure. The four cameras are the only sensors employed. There is no specific limit in the number of cameras that can be used by the human detection service and any other service of the Intelligent Space.

As mentioned before, the robot is tracked using a visual odometry service that detects a pattern attached to it, as shown in Figure 8. The built-in odometer and laser sensor of the robot are not used.

As can be noticed from Table 1, although Node 1 has a relative high amount of memory, the four instances of the Human Localization service (the most costly computational service) together uses approximately just 800 MB of RAM memory. Except for the robot, all devices uses Gigabit Ethernet interfaces.

Pedestrian detector models. The pedestrian detectors (ICCF and ACF) used in our human detection service were trained using the INRIA dataset[50] as described in Almonfrey et al.[38] No image sample of our Intelligent Space was used for training. These detectors can deal with different scales using image resampling, at the detection stage. They are not specifically trained to treat heavy occlusion and, because of this, the use of the redundant information of the camera network is important. For completeness, we present a quantitative and qualitative analysis of our human detection service to show its strengths and weaknesses.

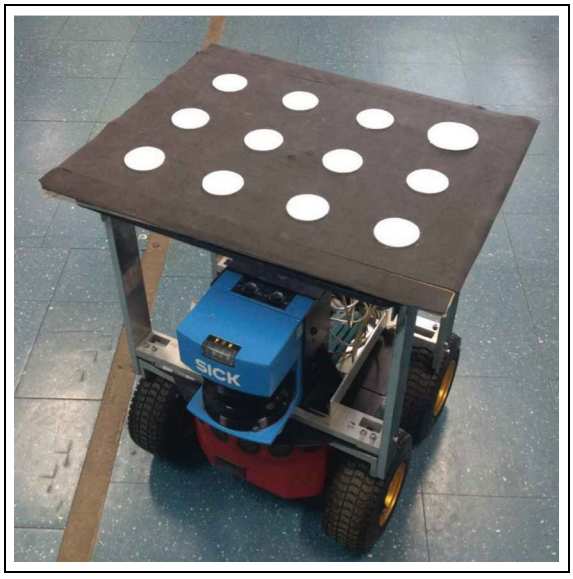

Figure 8. Robot used in the experiments.

With respect to the evaluation on the image plane, the Miss Rate (MR), False Positives Per Image (FPPI), and Precision (PR) are the metrics we used. These metrics were chosen because they are commonly employed in the literature[49] for image plane analysis. MR is the fraction of total humans not identified by the method during the experiment. FPPI is the number of nonhuman samples detected as humans divided by the number of images of the experiment. Precision represents the fraction of correct human detections in relation to all detections returned by our human detection service. It is important to mention that a match between a detection and a ground truth on the image plane is considered if there is an IoU greater than 0.5. Each ground truth can be matched at most once.

Concerning the evaluation on the ground plane, the number of True Positives (TP), False Positives (FP), and False Negatives (FN) are presented. Besides that, the localization error between the TP detection and its matched ground truth position is analyzed. To consider a match between a detection and a ground truth on the ground plane, an Euclidean distance smaller than 0.5 m (usually the average diameter of the area occupied by the human body on the ground plane) is demanded. Again, each ground truth can be matched at most once.

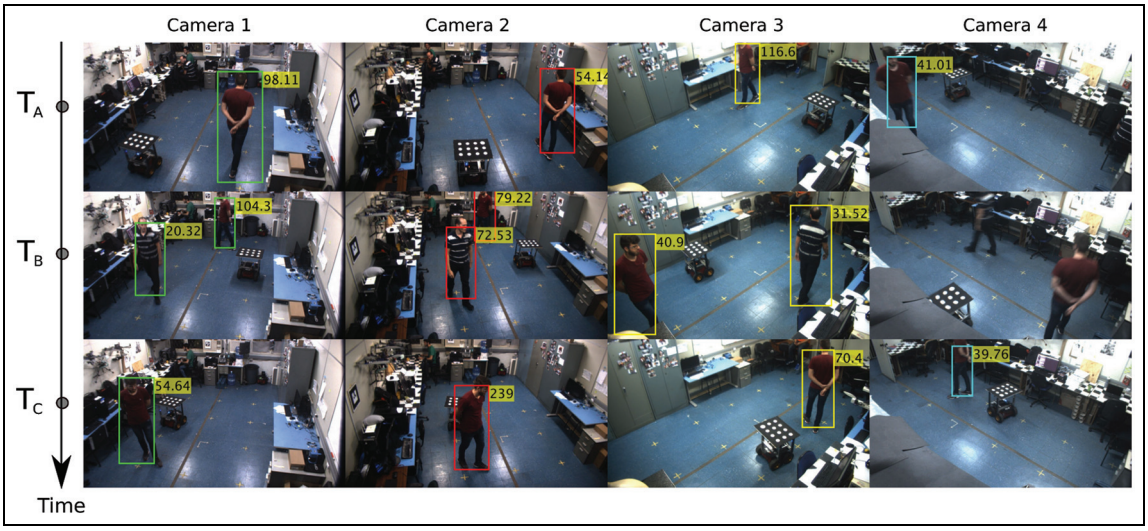

Figure 9. Human detection during the human-following task.

To compute the values of the metrics mentioned above, we manually annotated on the image plane all humans present in the experiments using BBs. To compute the localization error during the experiments, the 3D ground truth positions are obtained from the projection of the manually annotated BBs (perfect detector) on the ground plane. In a complementary analysis, we also compute the localization error through specific and uniformly distributed points on the ground plane of the Intelligent Space. In this case, we obtained the 3D ground truth annotations measuring the positions directly from the ground plane.

We also present a comparison of our human detection service with two well-known generic detectors of the pedestrian detection literature. ACF, which is included in part of our human detection pipeline, and locally decorrelated channel features (LDCF)[48] are used. These two detectors were chosen for comparison purposes because they present respectable results in public benchmarks, as shown in Ohn-Bar and Trivedi.[51] Although ACF has a lower accuracy when compared to LDCF, it is faster, presenting a better compromise between speed and accuracy. The used model of the LDCF detector was also trained in the INRIA dataset. Finally, without loss of self-completeness of the present work, a sample source code of the human detection service and all annotated data are made publicly available.[52]

Human-following task

In this experiment, the human detection service is used to generate set points to the robot's controller during the human-following task. Figure 9 shows the human detection process at three different times (T_A, T_B, and T_C) during robot navigation. Besides, the numbers shown in such figure are the values of the human detection confidence score.

Note that in Figure 9, at T_B, another human is present in the workspace just to show that the system can detect more than one pedestrian at a time during the experiment. However, the robot is configured to continue following the first human. Despite the fact that not all humans are detected by all cameras, they are detected by the majority of them most of the time. Therefore, the use of a camera network increases the human detection rate. It is worth to mention that our detector was not trained using images of our Intelligent Space and, because of that, some mistakes in the single view detection are expected.

For completeness, Table 2 shows an analysis on the image plane of the human detection service accuracy. The idea is not to claim state of the art detection, since this is not the focus of the present work. We just want to evaluate the detection in the context of the application's requirements. In the referred table, the detection performances of ACF and LDCF are also presented. By comparing our human detection service with two other detectors of the literature, we intend to show the importance of our complete detection pipeline to the accomplishment of the application.

From Table 2, it is possible to see that the overall PR (in bold) of our human detection service is much higher than the ones presented by the two other detectors. However, our method presents also a higher MR which is compensated by the fact that we use an array of cameras. The column related to the MR* metric shows that the overall MR (in bold) of our human detection service is decreased when considering information from all cameras. This lower MR indicates that humans lost in

Table 2. Analysis of the human detection on the image plane for the first application, considering our service, ACF, and LDCF.

Device	Ours				ACF			LDCF		
	MR (%)	MR* (%)	FPPI	PR (%)	MR (%)	FPPI	PR (%)	MR (%)	FPPI	PR (%)
Camera 1	14.0	11.2	5.4×10^{-3}	99.1	4.2	4.50	14.3	7.0	0.81	47.3
Camera 2	26.4	11.2	2.7×10^{-2}	94.9	4.8	6.40	9.2	5.6	2.70	19.1
Camera 3	38.7	6.6	4.9×10^{-2}	86.8	7.5	0.92	36.7	7.5	0.14	79.0
Camera 4	62.3	16.4	3.8×10^{-2}	71.4	24.6	0.27	48.4	21.0	3.8×10^{-2}	86.5
All cameras	30.3	**10.8**	3.0×10^{-2}	**93.1**	8.0	3.02	15.3	8.7	0.92	36.9

ACF: aggregate channel features; LDCF: locally decorrelated channel features; MR: Miss Rate; FPPI: False Positives Per Image; PR: Precision.
The symbol "*" denotes the BBs from images of other cameras are projected into the image of the current camera. This shows the advantage of using a camera network.

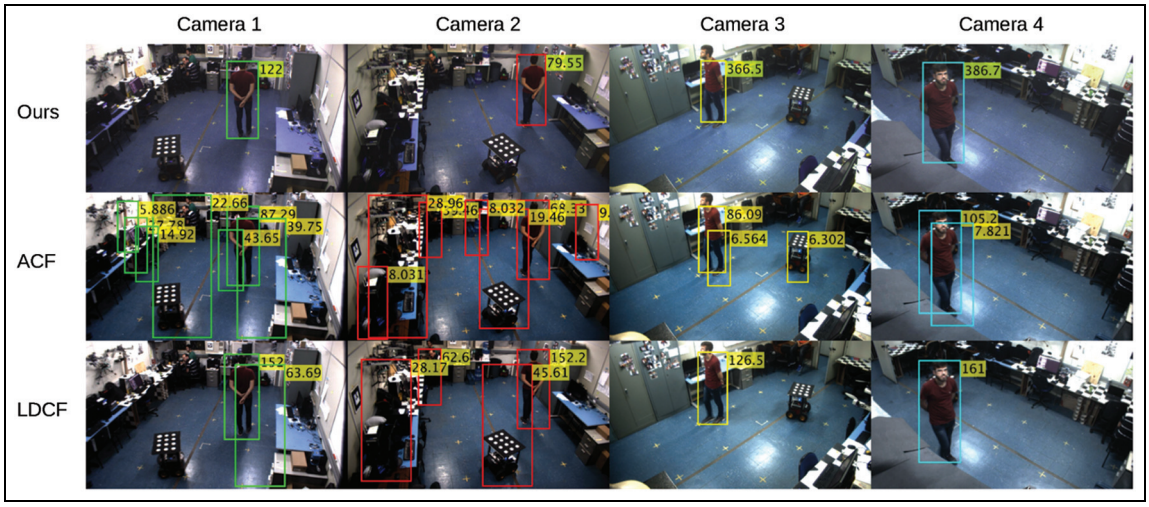

Figure 10. False positives of our service (first line), ACF (second line), and LDCF (third line).

one camera are detected in some other camera. This way we achieve a compromise between precision and loss of detections.

Without our complete detection pipeline to remove FP and the employment of a camera network to decrease the MR, the generic detectors ACF and LDCF cannot accomplish the task. It is important to mention that, even if ACF and LDCF were using a camera network, their PR would be inappropriate for the application, since they do not have a detection pipeline such ours to reduce FP. Figure 10 illustrates the higher number of FP of ACF and LDCF when compared to our method.

We relaxed the IoU threshold to consider a ground truth match to 0.3 just for the accuracy analysis of the reprojected BB. This is a fairer way to illustrate the accuracy, since the detections are approximately touching the ground plane. Therefore, some minor variations regarding the localization of the reprojected BB can be observed after homography transform. However, as mentioned before, during the experiments, we used an IoU threshold equal to 0.5. The humans located at the border of the image are not taken into account in the accuracy analysis. Although occlusions are not considered in the reasonable evaluation setup of the pedestrian detection area,[53] humans with occlusion are considered in our analysis.

Figure 11 shows the trajectory described by the human and the robot during the experiment. It is possible to see from the same figure that the human detection service provides suitable information for the task to be accomplished. The instants T_A, T_B, and T_C shown in Figure 9 are highlighted in the trajectory. Just the trajectory of the human being followed by the robot is shown for a better understanding. At the beginning of the experiment (time T_A), the robot was oriented 330° counterclockwise. In this case, it had to rotate approximately 120° on its own axis to start the following task.

As mentioned before, 3D human positions are obtained from the BBs on the images captured during the same sampling window. For any human, multiple positions can be obtained, since each camera can

Table 3. Analysis on the ground plane of the human detection during the first application.

TP	FP	FN	NDTS	NGTS	LOCERR (m)
129	5	20	134	149	0.11

TP: True Positives; FP: False Positives; FN: False Negatives.
NDTS is the total number of detections; NGTS is the number of ground truth positions estimated using the manually annotated BBs on the image plane; LOCERR is the localization error between the TP detection and its matched ground truth position.

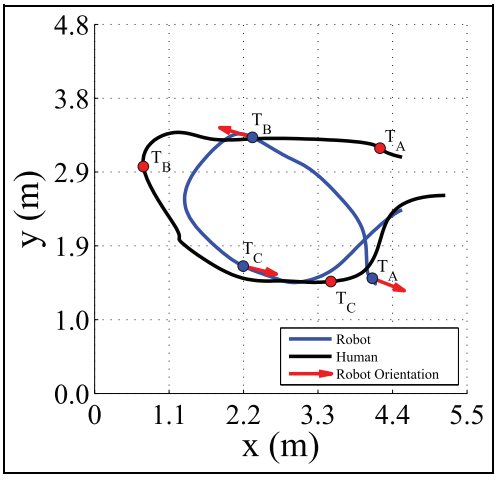

Figure 11. Trajectories of the human and the robot during the human-following task.

provide a detected BB, from which a 3D position is estimated. In all experiments of this work, human positions are clustered if they are inside a region of 0.5 m in diameter. This cluster is used to represent the localization of the human in the 3D space. During robot navigation, to improve robustness against punctual miss detections, a buffer is used to accumulate the 10 last positions of the human being followed by the robot. The average value of this buffer is adopted as the actual position.

Table 3 presents the metrics evaluated also on the ground plane for our human detection pipeline. Note that the MR on the ground plane (FN/NGTS) is approximately 13%, which is near the overall MR* on the image plane when considering information from all cameras (Table 2). This lower MR is further attenuated by the buffer used during robot navigation. This helps the control task in transient situations, when the human is missed. Note also that the low number of FP is in accordance with the low number of FPPI computed on the image plane. It is important to mention that the evaluation on the ground plane is correlated with the evaluation on the image plane, but not directly numerically related. Still from Table 3, we can note a localization error smaller than a quarter of the average diameter of the human body area on the ground plane (0.5 m).

Thus, the human detection service has a low number of FP that meets the demands of the proposed application, while keeping a proper MR. Again, this low number of FP is achieved due to the negative rejection cascade employed in the human detection service. Finally, our service was able to cooperate with the other elements of our infrastructure to accomplish the task.

Human-deviation task

In this experiment, the robot's controller uses the human detection service to perform the human-deviation task. We treat humans as obstacles and plan the path that must be executed by the robot using a path planning strategy presented in Şucan et al.[54] Figure 12 shows the human detection process during robot navigation in three different times (T_A, T_B, and T_C).

Figure 13 shows the trajectory described by the robot and the positions of the humans during the robot navigation. Using the information provided by the human detection service, the robot could navigate without hitting the individuals present in the environment. The instants T_A, T_B, and T_C shown in Figure 12 are highlighted in the trajectory. The path planning strategy looks for the path with less possibility of collision.

As presented in section "Human-following task," to validate the effectiveness of the service in meeting the requirements of the application, an analysis of the detector accuracy on the image and ground planes are presented in Tables 4 and 5, respectively. It is possible to see the same trend of the experiment of section "Human-following task." In general, our method has a higher overall precision, at the cost of a higher MR, which is attenuated by the use of a camera network and the buffering of 3D positions. Note that the low number of FN, presented in Table 5, corroborates with the low MR* (in bold) presented in Table 4. It is possible to see that ACF and LDCF still have a higher overall FPPI. Although the referred generic detectors have a low FP in some cameras, they would not be able to take advantage of the array of cameras, which is crucial in a more dynamic setup, as presented in sections "Human-following task" and "Occupancy map of the environment." On the other hand, our solution is more flexible

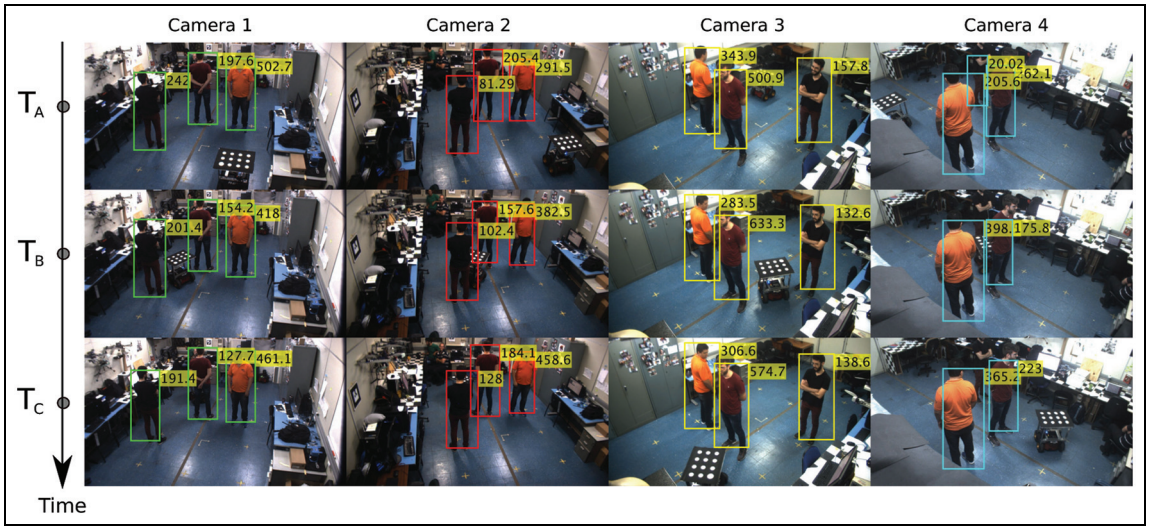

Figure 12. Human detection during the human-deviation task.

Table 4. Analysis of the human detection on the image plane for the second application, considering our service, ACF and LDCF.

Device	Ours				ACF			LDCF		
	MR (%)	MR* (%)	FPPI	PR (%)	MR (%)	FPPI	PR (%)	MR (%)	FPPI	PR (%)
Camera 1	0	0	0	100	0	5.19	36.7	0	2.0	60.0
Camera 2	2.5	0	4.0×10^{-2}	98.5	0	7.36	29.0	0	3.3	47.3
Camera 3	3.5	0	4.0×10^{-2}	98.5	0	0.86	77.7	0	3.2×10^{-1}	90.2
Camera 4	37.4	0	2.6×10^{-1}	82.8	0	1.93	51.0	0	8.0×10^{-2}	96.1
All cameras	8.4	**0**	8.7×10^{-2}	**96.7**	0	3.83	41.8	0	1.4	65.7

ACF: aggregate channel features; LDCF: locally decorrelated channel features; MR: Miss Rate; FPPI: False Positives Per Image; PR: Precision.
The symbol "*" denotes the BBs from images of other cameras are projected into the image of the current camera. This shows the advantage of using a camera network.

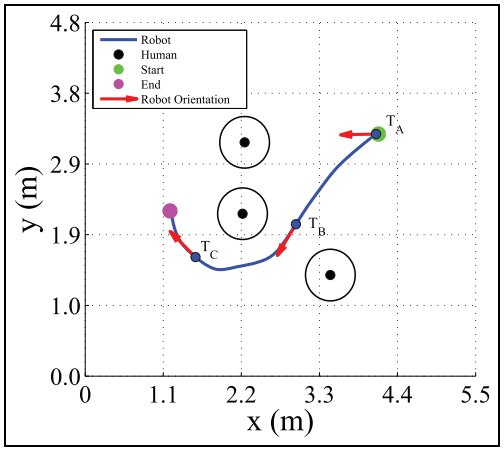

Figure 13. Trajectory described by the robot and the positions of the humans during the robot navigation.

since it is not dependent on only one specific camera, serving a higher range of applications. Note also that the localization error, for this experiment, is just 12% of the average diameter of the human body area on the ground plane (0.5 m). This is a reasonable value, considering that the detection process is applied to a low resolution image (515 × 291 pixels).

Finally, Table 6 shows a time analysis of the human detection service. A time T_{HDS} higher than T_{RC} means that the human detection service is not being able to process the images in the time demanded by the robot service. T_{HDS} less than or equal to T_{RC} implies that the human detection service processing time is in accordance with the time requirements. In practice, the value of T_{HDS} is expected to oscillate around a value near to T_{RC}. As indicated by the trajectory graphs, and confirmed by the time analysis, our method was able to serve the application while respecting the time requirements.

Occupancy map of the environment

In this experiment, we built a cumulative occupancy map of our Intelligent Space. The objective is to show

Table 5. Analysis on the ground plane of the human detection during the second application.

TP	FP	FN	NDTS	NGTS	LOCERR (m)
405	0	0	405	405	0.06

TP: True Positives; FP: False Positives; FN: False Negatives.
NDTS is the total number of detections; NGTS is the number of ground truth positions estimated using the manually annotated BBs on the image plane; LOCERR is the localization error between the TP detection and its matched ground truth position.

Table 6. Time analysis.

T_{RC} (s)	T_{HDS} (s)
0.250	0.195

T_{RC} is the maximal time interval required by the robot service to receive human position information in order to properly operate the robot; T_{HDS} is the approximate measured time between capturing the image and delivering the control commands to the robot. Therefore, this is the human detection service processing time. This time was obtained from an average of 180 samples.

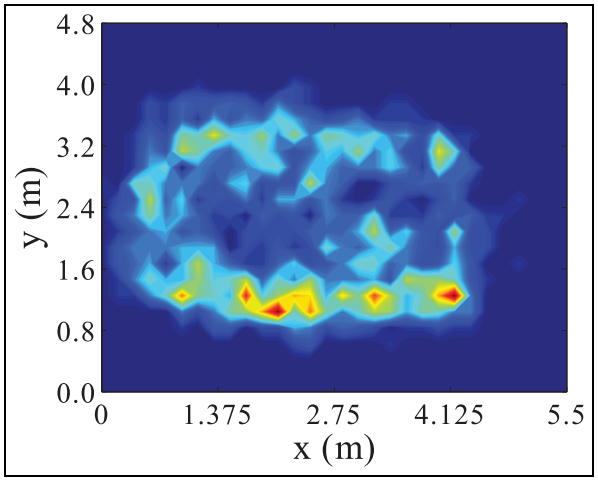

Figure 14. Cumulative occupancy map.

the most visited places of the environment through time. Figure 14 shows the cumulative occupancy map. Our Intelligent Space measures 4.8 m × 7.3 m. However, due to the positioning of the camera system, part of the human bodies is outside of the image area or distorted for most cameras in the last 1.8 m of the horizontal axis. As no human is detected in the referred region, we just show a 4.8 m × 5.5 m area from our Intelligent Space. This is the operating area of the detection service. The humans were asked to mainly execute an elliptical trajectory in this experiment and this can be confirmed from Figure 14. In Surie et al.,[18] where the solution is highly structured, the authors mention that a larger field of view is expected to impair the detection. As our work can deal with workspaces less structured, a larger field of view will benefit the detection due to a larger operating area.

As already done in the other experiments, Tables 7 and 8 present the accuracy analysis on the image and ground planes, respectively. Again, the trend is the same of the other experiments when comparing our method with ACF and LDCF. The difference now is that more people are present in the Intelligent Space, increasing the MR of all detectors, because of the higher rate of occlusion. In fact, occlusion is still an open problem on the literature. It is important to mention that even the best trained generic detectors present a drop in performance when considering occlusion, as can be confirmed in the Caltech dataset benchmark.[55] From Table 8, it is also possible to see that the MR on the ground plane (FN/NGTS) is approximately 29%, which is near the overall MR* on the image plane when considering information from all cameras (Table 7). The precision of our detector is still much higher than the precision of the other detectors used for comparison, as can be confirmed in Table 7.

As a complementary analysis, we show that the higher number of FN, on the ground plane, is somehow related to minor variations in detections due to the challenging setup of this experiment. Just at this evaluation, we relaxed the threshold distance on the ground plane from 0.5 to 0.8 m, for the clustering approach and accuracy analysis. Also, we calculated the metrics on the ground plane after the buffering approach, to show its effectiveness. As can be seen from Table 8, when considering this relaxed setup, the number of FN* reduces considerably at the cost of a minor increase in the number of FP*. As can be seen, with a reasonable relaxation in the threshold distance on the ground plane, a lot of FN are avoided. This shows that many mistakes are indeed made by a minor margin, which represent a minor impact on the applications. Therefore, the higher MR on the image plane is indeed reduced on the ground plane when using an array of cameras.

Regarding the localization error, from Table 8, we can note a slight increase of the error in relation to that presented in the other experiments. This is mainly due to the more dynamic setup of this experiment, where there are more movement and occlusion, which increases the instability on the human detection on

Table 7. Analysis on the image plane of the human detection during the third application.

Device	Ours				ACF			LDCF		
	MR (%)	MR* (%)	FPPI	PR (%)	MR (%)	FPPI	PR (%)	MR (%)	FPPI	PR (%)
Camera 1	38.3	26.2	5.5×10^{-3}	93.5	4.5	4.95	20.0	5.4	0.53	69.7
Camera 2	48.5	26.3	8.4×10^{-2}	87.1	7.9	5.10	16.8	9.9	1.58	39.0
Camera 3	60.2	22.3	5.1×10^{-2}	86.5	13.5	0.67	51.5	17.4	0.20	77.6
Camera 4	84.4	29.8	1.8×10^{-2}	85.6	28.1	0.22	69.7	36.5	7.0×10^{-2}	86.5
All cameras	54.0	**26.0**	5.2×10^{-2}	**89.6**	11.6	2.73	24.26	14.7	0.59	58.7

ACF: aggregate channel features; LDCF: locally decorrelated channel features; MR: Miss Rate; FPPI: False Positives Per Image; PR: Precision.
The symbol "*" denotes the BBs from images of other cameras are projected into the image of the current camera. This shows the advantage of using a camera network.

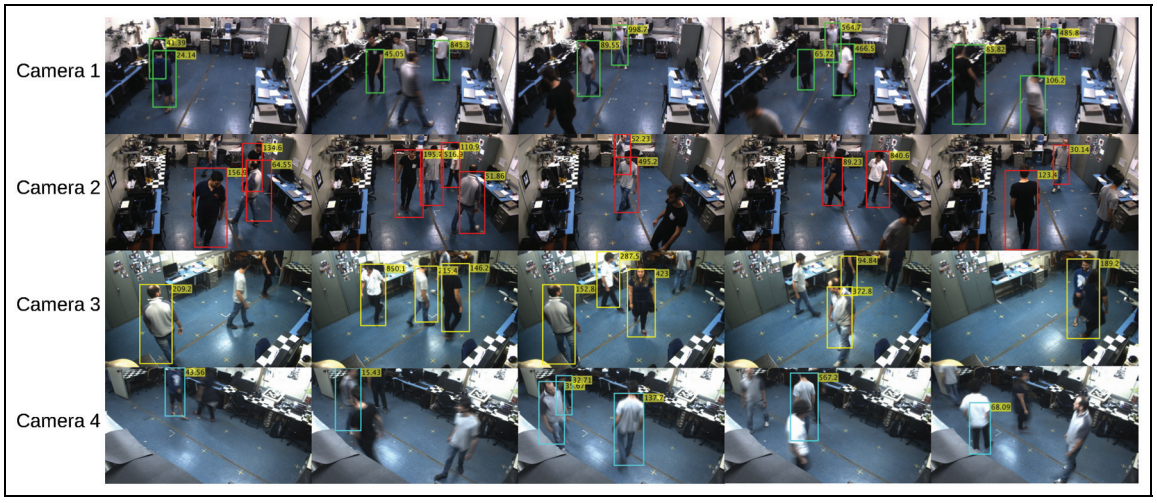

Figure 15. Qualitative results of the detection for occupancy map application. Both successful and faulty detections can be seen in the presence of occlusions.

Table 8. Analysis on the ground plane of the human detection during the third application.

TP	FP	FN	NDTS
1411	115	595	1526
NGTS	LOCERR (m)	FP*	FN*
2006	0.18	198	159

TP: True Positives; FP: False Positives; FN: False Negatives.
NDTS is the total number of detections; NGTS is the number of ground truth positions estimated using the manually annotated BBs on the image plane; LOCERR is the localization error between the TP detection and its matched ground truth position.
The symbol "*" denotes metrics computed after threshold distance relaxation and the buffering approach.

each image and thereafter on human localization on the ground plane. However, once that detection is applied to a low resolution image (515 × 291 pixels), an error smaller than 40% of the average diameter of the human body area on the ground plane can be considered reasonable.

Finally, Figure 15 presents some qualitative results, including some successful detections and some failures. For a better diversity, the four images of the cameras are not necessarily correspondent in this case.

Localization error of the Intelligent Space

As a final evaluation, we measured the localization error of the human detection service for positions distributed as an uniform grid on the operating area of our Intelligent Space (Figure 16). In order to do that, a person stood at the grid locations, which were also manually annotated on the ground, while images were captured. The idea was to use a more structured setup to have a better analysis of the influence in the localization error due to the detection and calibration data.

As can be seen from Figure 16, this error is not uniformly distributed over the environment, and this is

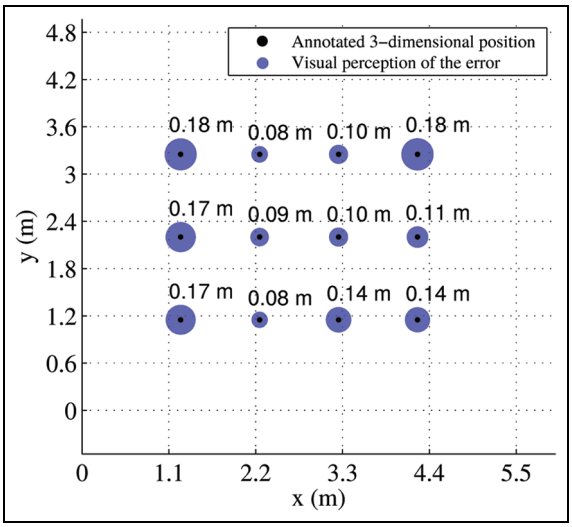

Figure 16. Localization error along the operating area of the Intelligent Space.

probably due to the nonuniform redundancy inserted by the overlapping area of the cameras. The redundancy in the detection is also important for a precise localization on the ground plane. At the border of the operating area, where there is less redundancy in the detection, the error is greater. Nevertheless, the obtained error is again smaller than 40% of the average diameter of the human body area on the ground plane.

Conclusion

In this work, we presented a human detection service suitable for Intelligent Spaces based on a multi-camera network. We presented a review of the literature to highlight the main distinction between our method and the others existing in the area. While in the literature most of the solutions are just independent applications, our human detection service is designed to interact harmoniously with the architecture of our Intelligent Space.

The proposed human detector, as well as our Intelligent Space, is implemented using concepts of cloud computing and SOA. Our service is designed to be flexible, less structured as possible, meeting the requirements of different Intelligent Spaces applications and services present in our architecture. Our human detection service, due to the paradigm in which is implemented, is scalable, reliable and parallelizable.

As it can be commonly found in different environments, the multi-camera system is used to overcome some limitations of existing human detection approaches. Finally, concerning time and detection performance requirements, our solution proved to be suitable for interacting with other services and applications of our Intelligent Space and successfully accomplish the proposed tasks.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Douglas Almonfrey https://orcid.org/0000-0002-0547-3494

References

1. Weiser M. The computer for the 21st century. *Sci Am* 1991; 265(3): 66–75.
2. Weiser M. The world is not a desktop. *Interactions* 1994; 1(1): 7–8.
3. Coen MH. Design principles for intelligent environments. In: *Proceedings of the fifteenth national/tenth conference on artificial intelligence/innovative applications of artificial intelligence*, Madison, WI, 1 July 1998, pp.547–554. Menlo Park, CA: American Association for Artificial Intelligence.
4. Wright S and Steventon A. Intelligent spaces—the vision, the opportunities and the barriers. *BT Technol J* 2004; 22(3): 15–26.
5. Lee JH and Hashimoto H. Intelligent space: concept and contents. *Adv Robotics* 2002; 16(3): 265–280.
6. Lee C, Nordstedt D and Helal S. Enabling smart spaces with OSGi. *IEEE Pervas Comput* 2003; 2(3): 89–94.
7. Helal S, Mann W, El-Zabadani H, et al. The Gator Tech Smart House: a programmable pervasive space. *Computer* 2005; 38: 50–60.
8. Chen H, Finin T, Joshi A, et al. Intelligent agents meet the semantic web in smart spaces. *IEEE Internet Comput* 2004; 8: 69–79.
9. Frejlichowski D, Gosciewska K, Forczmanski P, et al. "SmartMonitor"—an intelligent security system for the protection of individuals and small properties with the possibility of home automation. *Sensors* 2014; 14: 9922–9948.
10. Wang T, Ramik DM, Sabourin C, et al. Intelligent systems for industrial robotics: application in logistic field. *Ind Robot* 2012; 39: 251–259.
11. Glas DF, Kamei K, Kanda T, et al. Human-robot interaction in public and smart spaces. In: Mohammed S, Moreno J, Kong K, et al. (eds) *Intelligent assistive robots: recent advances in assistive robotics for everydayc activities*, vol. 106. Cham: Springer, 2015, pp.235–273.
12. Glas DF, Satake S, Ferreri F, et al. The network robot system: enabling social human-robot interaction in public spaces. *J Hum Robot Interact* 2013; 1(2): 5–32.
13. Zhou Z, Yang Z, Wu C, et al. Towards omnidirectional passive human detection. In: *2013 Proceedings IEEE*

INFOCOM, Turin, 14–19 April 2013, pp.3057–3065. New York: IEEE.
14. Mrazovac B, Bjelica MZ, Kukolj D, et al. System design for passive human detection using principal components of the signal strength space. In: *2012 IEEE 19th international conference and workshops on engineering of computer-based systems*, Novi Sad, 11–13 April 2012, pp.164–172. New York: IEEE.
15. Brščić D. Social robots in smart public environments. In: *2014 IEEE 3rd global conference on consumer electronics (GCCE)*, Tokyo, Japan, 7–10 October 2014, pp.651–653. New York: IEEE.
16. Cook DJ, Crandall AS, Thomas BL, et al. CASAS: a smart home in a box. *Computer* 2013; 46(7): 62–69.
17. Morioka K, Hashikawa F and Takigawa T. Human identification based on walking detection with acceleration sensor and networked laser range sensors in intelligent space. *Int J Smart Sens Intell Syst* 2013; 6(5): 2040–2054.
18. Surie D, Partonia S and Lindgren H. Human sensing using computer vision for personalized smart spaces. In *2013 IEEE 10th international conference on ubiquitous intelligence and computing and 2013 IEEE 10th international conference on autonomic and trusted computing*, Vietri sul Mere, 18–21 December 2013, pp.487–494. New York: IEEE.
19. Cook DJ, Crandall A, Singla G, et al. Detection of social interaction in smart spaces. *Cybernet Syst* 2010; 41(2): 90–104.
20. Chen SL, Chang SK and Chen YY. Development of a multisensor embedded intelligent home environment monitoring system based on digital signal processor and Wi-Fi. *Int J Distrib Sens N* 2015; 11: 171365.
21. Lee JE, Kim JH, Kim SJ, et al. Human and robot localization using histogram of oriented gradients (HOG) feature for an active information display in intelligent space. *Adv Sci Lett* 2012; 9: 99–106.
22. Zabulis X, Grammenos D, Sarmis T, et al. Multicamera human detection and tracking supporting natural interaction with large-scale displays. *Mach Vision Appl* 2013; 24(2): 319–336.
23. Li B, Yao Q and Wang K. A review on vision-based pedestrian detection in intelligent transportation systems. In: *Proceedings of 2012 9th IEEE international conference on networking, sensing and control*, Beijing, China, 11–14 April 2012, pp.393–398. New York: IEEE.
24. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the 2014 IEEE conference on computer vision and pattern recognition*, Columbus, OH, 23–28 June 2014, pp.580–587. Washington, DC: IEEE Computer Society.
25. Zhang L, Lin L, Liang X, et al. Is faster R-CNN doing well for pedestrian detection? In: B Leibe, J Matas, N Sebe, et al. (eds) *Computer vision –ECCV 2016* (Lecture Notes in Computer Science, vol. 9906). Cham: Springer, pp.443–457.
26. Ribeiro D, Mateus A, Miraldo P, et al. A real-time deep learning pedestrian detector for robot navigation. In: *2017 IEEE international conference on autonomous robot systems and competitions (ICARSC)*, Coimbra, 26–28 April 2017, pp.165–171. New York: IEEE.
27. ATIS telecom glossary, http://www.atis.org/glossary/
28. Cai Z, Fan Q, Feris R, et al. A unified multi-scale deep convolutional neural network for fast object detection. In: Leibe B, Matas J, Sebe N, et al. (eds) *ECCV* (Part of the Lecture Notes in Computer Science book series (LNCS), vol. 9908). Cham: Springer, pp.354–370.
29. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE T Pattern Anal* 2017; 39: 1137–1149.
30. Redmon J and Farhadi A. Yolo9000: better, faster, stronger. arXiv:1612.08242.
31. Mao J, Xiao T, Jiang Y, et al. What can help pedestrian detection? In: *30th IEEE conference on computer vision and pattern recognition*, July 2017, pp.3127–3136. New York: IEEE.
32. Dollár P, Appel R, Belongie S, et al. Fast feature pyramids for object detection. *IEEE T Pattern Anal* 2014; 36: 1532–1545.
33. Benenson R, Omran M, Hosang J, et al. Ten years of pedestrian detection, what have we learned? In: Agapito L, Bronstein M and Rother C (eds) *Computer vision—ECCV 2014 workshops. ECCV 2014* (Lecture Notes in Computer Science, vol. 8926). Cham: Springer, pp.613–627.
34. Zhang S, Benenson R and Schiele B. Filtered channel features for pedestrian detection. In: *2015 IEEE conference on computer vision and pattern recognition*, Boston, MA, 7–12 June 2015, pp.1751–1760. New York: IEEE.
35. Hosang J, Omran M, Benenson R, et al. Taking a deeper look at pedestrians. In: *2015 IEEE conference on computer vision and pattern recognition*, Boston, MA, 7–12 June 2015, pp.4073–4082. New York: IEEE.
36. Du X, El-Khamy M, Lee J, et al. Fused DNN: a deep neural network fusion approach to fast and robust pedestrian detection. arXiv:1610.03466.
37. Zhang S, Benenson R and Schiele B. CityPersons: a diverse dataset for pedestrian detection. In: *30th IEEE conference on computer vision and pattern recognition*, July 2017, pp.3213–3221. New York: IEEE.
38. Almonfrey D, Vassallo RF, Salles EOT, et al. Neural cells insights on pedestrian detection. In: *CBA 2016—XXI Brazilian conference of automation*, http://www.swge.inf.br/proceedings/paper/?P=CBA2016-0590
39. Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv: 1409.1556.
40. Matsuhira N, Ozaki F, Tokura S, et al. Development of robotic transportation system—shopping support system collaborating with environmental cameras and mobile robots. In: *ISR 2010 (41st international symposium on robotics) and ROBOTIK 2010 (6th German conference on robotics)*, Munich, 7–9 June 2010, pp.1–6. New York: IEEE.
41. Albawendi S, Appiah K, Powell H, et al. Overview of behavioural understanding system with filtered vision sensor. In: *2015 international conference on interactive technologies and games*, Nottingham, 22–23 October 2015, pp.90–95. New York: IEEE.
42. Adduci M, Amplianitis K and Reulke R. A quality evaluation of single and multiple camera calibration approaches for an indoor multi camera tracking system.

In: *ISPRS—International archives of the photogrammetry, remote Sensing and spatial information sciences*, Riva del Garda, 23–25 June 2014, pp.9–15. International Society of Photogrammetry and Remote Sensing (ISPRS).
43. Rampinelli M, Covre VB, de Queiroz FM, et al. An intelligent space for mobile robot localization using a multi-camera system. *Sensors* 2014; 14(8): 15039–15064.
44. Gomes RL, Martinello M, Dominicini CK, et al. How can emerging applications benefit from EaaS in open programmable infrastructures? In: *IEEE summer school on smart cities 2017: IEEE S3C2017*, Natal, 6–11 August 2017, http://www.ict-futebol.org.br/wp-content/uploads/2017/08/How-can-emerging-applications-benefit-from-EaaS-in-open-programmable-infrastructures.pdf
45. Rostanski M, Grochla K and Seman A. Evaluation of highly available and fault-tolerant middleware clustered architectures using RabbitMQ. In: *2014 federated conference on computer science and information systems*, Warsaw, 7–10 September 2014, pp.879–884. New York: IEEE.
46. Merkel D. Docker: lightweight Linux containers for consistent development and deployment. *Linux J* 2014; 2014: Article 2.
47. Burns B, Grant B, Oppenheimer D, et al. Borg, omega, and kubernetes. *Queue* 2016; 14: 70–93.
48. Nam W, Dollár P and Han JH. Local decorrelation for improved pedestrian detection. In: *Proceedings of the 27th international conference on neural information processing systems*, Montreal, QC, Canada, 8–13 December 2014, vol. 1, pp.424–432. Cambridge, MA: MIT Press.
49. Dollár P, Wojek C, Schiele B, et al. Pedestrian detection: an evaluation of the state of the art. *IEEE T Pattern Anal* 2012; 34: 743–761.
50. Dalal N and Triggs B. Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society conference on computer vision and pattern recognition*, San Diego, CA, 20–25 June 2005, vol. 1, pp.886–893. New York: IEEE.
51. Ohn-Bar E and Trivedi MM. To boost or not to boost? On the limits of boosted trees for object detection. In: *2016 23rd international conference on pattern recognition (ICPR)*, 4 December 2016, pp.3350–3355. New York: IEEE.
52. Human detection service, https://bitbucket.org/Monfa/humandetectionservice
53. Dollár P, Tu Z, Perona P, et al. Integral channel features. In: Cavallaro A, Prince S and Alexander D (eds) *Proceedings of the British machine vision conference*. Durham: BMVA Press, pp.91.1–91.11.
54. Şucan IA, Moll M and Kavraki LE. The open motion planning library. *IEEE Robot Autom Mag* 2012; 19(4): 72–82, http://ompl.kavrakilab.org
55. Caltech dataset benchmark, http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/