SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2018, Vol. 61 050102:1–050102:16
https://doi.org/10.1007/s11432-017-9380-5

Special Focus on Internetware: Research Progress and Current Trend

RoboCloud: augmenting robotic visions for open environment modeling using Internet knowledge

Yiying LI, Huaimin WANG*, Bo DING & Wei ZHOU

College of Computer, National University of Defense Technology, Changsha 410073, China

Received 16 November 2017/Revised 5 February 2018/Accepted 13 March 2018/Published online 18 April 2018

Abstract Modeling an open environment that contains unpredictable objects is a challenging problem in the field of robotics. In traditional approaches, when a robot encounters an unknown object, a mistake will inevitably be added to the robot's environmental model, severely constraining the robot's autonomy, and possibly leading to disastrous consequences in certain settings. The abundant knowledge accumulated on the Internet has the potential to remedy the uncertainties that result from encountering with unknown objects. However, robotic applications generally pay considerable attention to quality of service (QoS). For this reason, directly accessing the Internet, which can be unpredictable, is generally not acceptable. RoboCloud is proposed as a novel approach to environment modeling that takes advantage of the Internet without sacrificing the critical properties of QoS. RoboCloud is a "mission cloud–public cloud" layered cloud organization model in which the mission cloud provides QoS-available environment modeling capability with built-in prior knowledge while the public cloud is the existing services provided by the Internet. The "cloud phase transition" mechanism seeks help from the public cloud only when a request is outside the knowledge of the mission cloud and the QoS cost is acceptable. We have adopted semantic mapping, a typical robotic environment modeling task, to illustrate and substantiate our approach and key mechanism. Experiments using open 2D and 3D datasets with real robots have demonstrated that RoboCloud is able to augment robotic visions for open environment modeling.

Keywords Internet-augmented, environment modeling, uncertainty, robotic visions, semantic mapping, robotic software, cloud robotics

Citation Li Y Y, Wang H M, Ding B, et al. RoboCloud: augmenting robotic visions for open environment modeling using Internet knowledge. Sci China Inf Sci, 2018, 61(5): 050102, https://doi.org/10.1007/s11432-017-9380-5

1 Introduction

By integrating various sensors and actuators, more and more computer-based systems can interact directly with physical space. As a result, current software must play its role not only in the traditional information space but also in physical space. A typical example is robotic environment modeling [1], which is the foundation of human-robot communication and of a robot's capabilities for planning and operating in the real world. As an essential component in the robotic software stack, environment modeling enables a mobile robot to obtain useful information, such as geometric and semantic, about the environment. For example, semantic mapping is a typical robotic environment modeling task for building a geometric map with a set of semantic labels (e.g., object names). To tag objects, traditional approaches in this field heavily rely on prior knowledge that has been built into the robot by machine learning or other means [2]. However, because the physical environment is inherently open and dynamic, all the objects that a robot

* Corresponding author (email: hmwang@nudt.edu.cn)

may encounter for a complex task can be difficult to predict with accuracy. When a robot encounters an unknown object, a mistake will inevitably be added to the robot's environment model. The lack of necessary knowledge could even lead to disastrous consequences in extreme cases, such as the navigation accident of autopiloted cars[1].

Aside from the emergence of cyber-physical systems in the past three decades, we have also witnessed the abundant knowledge accumulated on the Internet [3]. The knowledge can be accessed by software in a variety of forms, such as open datasets, Wiki pages, big data-based Internet services (e.g., Google's Cloud Vision API[2] and CloudSight[3] for object recognition). The Internet has the potential to help robots handle uncertainty when encountering unknown objects by providing the necessary information about such objects. However, there are a number of difficulties to be overcome to enable robots to seek such information. First, the unpredictable objects must be identified dynamically, which means that the existing robotic environmental modeling approaches must be significantly extended. More importantly, quality of service (QoS) assurance, such as latency or real-time assurance, is essential to robotic software and applications. Since the Internet and its services are mostly based on the "best-effort" model, direct access by robots to the Internet is unacceptable in most cases. Hence, a mechanism is required to enable a robot to decide when and how to seek help appropriately from the Internet. There has been some research into a cloud robotic paradigm of robotic environmental modeling [4]. In the existing study, the cloud acts merely as a computational offloading infrastructure or a data repository with a fixed boundary. These solutions result in an isolated information island, but the power of cloud computing is mainly derived from openness rather than isolation. Thus, the challenges mentioned above remain unresolved.

In this paper, we propose a novel environment modeling approach named RoboCloud, whose goal is to model the environment and cope with uncertainties. With RoboCloud, we introduce a "mission cloud–public cloud" cloud organization model. The mission cloud is proprietary, and all its controllable resources are deployed near the robots, thereby providing QoS-available environmental modeling capabilities with prior knowledge that is highly related to the robotic modeling mission. An example of such prior knowledge is knowledge of objects predicted to appear in the environment. When a robot's request is beyond the capability of the mission cloud and the QoS cost is acceptable, the robot will seek help from the public cloud which refers to the mature existing services on the Internet. We have designed a delicate mechanism called the "cloud phase transition" for the cooperation of these two clouds to decide when and how the mission cloud should seek help from the public cloud. RoboCloud takes advantage of the great potential of the Internet without sacrificing critical QoS properties. RoboCloud also draws on the idea of edge computing and can be regarded as an innovative application and the extension of the edge computing into robotics. Edge computing, which calls for the processing of data at the edge of a network, has recently emerged especially in the Internet of things (IoT). We have adopted semantic mapping based on the RoboCloud approach to illustrate the performance of the cooperation mechanism and the improvement in the capability of robotic environment modeling. Experiments conducted on both open 2D and 3D datasets with real robotic scenarios have demonstrated that RoboCloud is able to augment robotic visions for semantic mapping in the open, uncertain environment.

2 Background and related work

This section first introduces the computing paradigms that have inspired cloud robotic architecture. We then introduce a typical robotic environment modeling task, robotic semantic mapping with its existing cloud-based approaches. Finally, we discuss those Internet services, such as big data-based Internet image recognition services, that can be used in robotics.

[1] Autopilot-tesla. https://www.tesla.com/autopilot/.

[2] Google Cloud Vision API. https://cloud.google.com/vision/.

[3] CloudSight. http://cloudsightapi.com/.

2.1 Cloud robotic architecture

Having been initially proposed in 2010, cloud robotics is a relatively new field of research [4], in which most of the existing work has been based on the "single" cloud architecture. The cloud computing paradigm has inspired robotics by offloading specific robotic algorithms onto cloud infrustractures, such as Hadoop clusters, while gathering, sharing, and obtaining information on the cloud. One famous cloud robotics project is a European FP7 project named RoboEarth [5], which is based on a three-layer architecture: a back-end database, a cloud engine, and robots. With the development of cloud computing applications in the real world, such as applications on mobile devices, a new computing paradigm named edge computing has emerged. Compared to the fast-developing data processing speed of the cloud, data transportation is becoming a bottleneck for the cloud-based computing paradigm. The idea of introducing an edge side with its small-scale computing infrastructure near intelligent terminals is to avoid overly long response time (i.e., to improve certain QoS properties), such as the work in Cloudlet [6]. Therefore, since many robotic tasks request real-time assurance, edge computing can be applied to cloud robotics, so that robots can perform actions in the real physical world. Edge computing plays an intermediary role in allowing computation to be performed at the edge of the network on downstream data on behalf of cloud services and on upstream data on behalf of robotic services. There have been some related work on such layered architecture in cloud robotics. For example, UNR-PF [7] is a cloud robotic platform that divides the cloud into two layers: a local platform and a global platform for the coordination of robots over a wide area. Overall, the main problem of cloud robotic architecture we are addressing is the utilization of the large amount of knowledge in the cloud while satisfying QoS assurances.

2.2 Robotic semantic mapping and existing cloud-based solutions

Accurate and reliable environment modeling is essential for mobile robots. This topic has been discussed for several decades. A common task is semantic mapping [2], which builds a map with semantic labels. The mapping process can be divided into two iterative stages: obtaining a geometric map and adding high-level semantic labels [8]. The former can be realized by off-the-shelf simultaneous localization and mapping (SLAM) algorithms [9], while the latter is usually realized by machine-learning approaches that heavily rely on the knowledge possessed by the robot (e.g., object shapes and structures). A basic form of semantic labeling is obtaining the names of objects in the environment.

Given that robotic semantic mapping is a computation- and knowledge-intensive task, there have been several cloud robotics-based approaches. For example, a complete cloud-based robotic semantic mapping engine is present in the RoboEarth project [5], which combines a knowledge base with a visual SLAM map at the object level to realize the RoboEarth web and cloud mapping services. A robot that operates in an environment for the first time can benefit from information that has been previously stored in the cloud by other robots. Rapyuta is a part of RoboEarth and helps robots to offload heavy computation, such as SLAM, to the cloud by providing each robot with a customized virtual computing environment [10]. RoboBrain provides a set of cloud services to store and share knowledge, such as object labels, among robots [11]. In [12], a low-cost backup Hadoop cluster is leveraged to support UAVs for the acquisition of scene semantic information. Heavy computations, such as image processing, are offloaded to this cluster.

The above-mentioned studies validate the cloud's boosting of a robot's environment modeling capability. The focus of these studies is the offloading of computations to or obtaining knowledge from the cloud on demand. However, the question, "How to process unfamiliar objects that are not in the cloud's knowledge repository?" remains. Our study implemented robotic semantic mapping task in an open environment. By utilizing Internet intelligence services, a robot's modeling capability of uncertainty can be improved.

2.3 Internet-based object recognition services

In recent years, the development of computational intelligence [3] has significantly promoted the cognitive capability of computers. Since the efficiency of computational intelligence relies heavily on the scale of

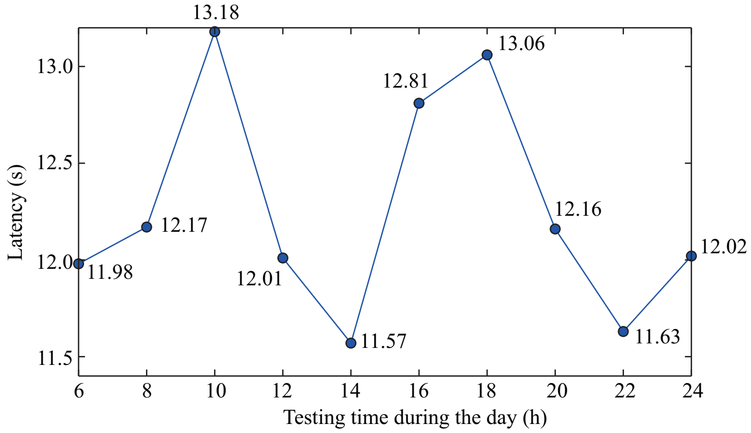

Figure 1 (Color online) Recognition latency of CloudSight.

training data and the capability of computing infrastructures, it is an ideal solution to put the implementation into cloud facilities. Various Internet-based image recognition services, such as CloudSight, Google Vision API, and Image++[4], have recently emerged. For example, CloudSight, a service originally designed for mobile phone apps, can recognize more than 40 million objects and has already processed more than 400 million images. By exploiting the potential of big data and distributed computing in the back end, these services can provide highly accurate recognition results through service-oriented, HTTP-based interfaces. However, these services are not able to provide the QoS assurance that is critical to many robotic tasks. For example, the recognition latencies of CloudSight were tested at different moments of the day. The results are shown in Figure 1. The average unstable latency is about 12.26 s under a 4G network.

3 RoboCloud system architecture

The RoboCloud environment modeling approach is designed to fulfill two goals: (1) Augmenting a robot's environment modeling capability and efficiency with the help of cloud resources, especially mission cloud resources, and (2) handling uncertainty with the help of the knowledge on the Internet via public cloud services. The first goal is realized by offloading the computations of the robot onto the edge of the network and introducing a state-of-the-art deep neural network (DNN) or support vector machine (SVM)-based classifier that can make use of rich computing resources for real-time and highly accurate recognition of 2D or 3D objects images expected to appear in the modeling environment. The second goal, which is also the major feature of our approach, is realized by allowing the mission cloud to seek help from the public cloud services when necessary and in a carefully designed way. Figure 2 gives an overview of the main components of the RoboCloud environment modeling architecture as well as the runtime interactions among these components. For this study, we adopted a typical environment modeling task, semantic mapping, as a case study. The RoboCloud system plays three roles.

Robots. These are autonomous devices that move around in the environment, collecting images and uploading the surroundings images, as well as the robots' pose and odometric data, to the cloud for semantic mapping. By utilizing the cloud, a robot can break through the limitations of local resources, which include its computing resources and knowledge.

Mission cloud. The mission cloud is a small-scale computing infrastructure that is deployed near the robots at the edge of the network. For example, we deployed a mission cloud on a computing cluster in a robotic operations center. The cloud was composed of three modules: (1) SLAM module. The geometric map can be drawn according to the laser and pose data dynamically collected by a mobile robot connected to the mission cloud. We adopted a mature SLAM algorithm based on ROS (robot operating

[4] Image++. http://www.imageplusplus.com/.

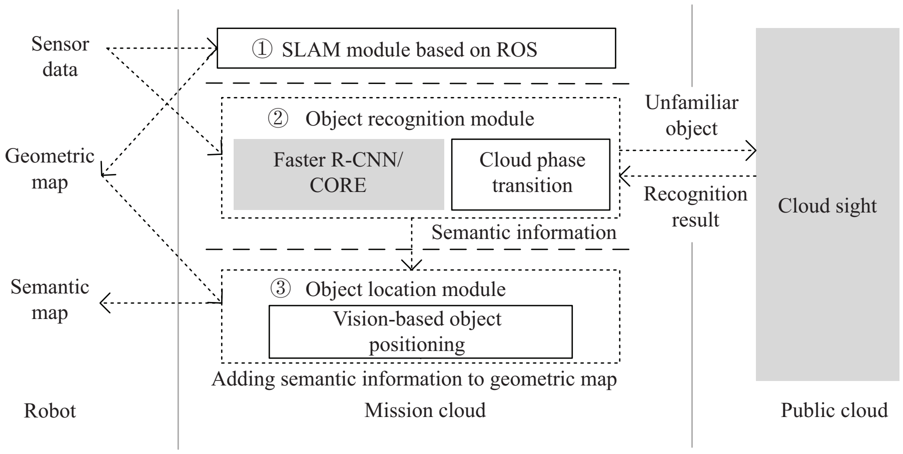

Figure 2 RoboCloud semantic mapping system.

system), slam_gmapping, to create the map. (2) Object recognition module. The mission cloud aims to collect the semantic information of the environment at the object level. For 2D images captured by robots, we adopted a state-of-the-art DNN-based recognition engine named Faster R-CNN [13] for image region proposal and object recognition. In addition, robots equipped with sensors for their surroundings were able to capture 3D point cloud images. We adopted an SVM-based classifier named CORE (cloud object recognition engine) [14] to process the RGB-D images for recognition. The softmax/SVM classifier outputs the object category label as the recognition result. Knowledge that is highly related to the environment modeling mission (e.g., objects expected to appear in the current environment) and can be learned in advance is built into this cloud. Fulfilling certain QoS requirements while a robot's request is within the boundary of the mission cloud is possible because the resources in the mission cloud and the network that connects the robots are controllable. As for unfamiliar objects beyond the knowledge of the Faster R-CNN/CORE module (i.e., not trained), the mission cloud may seek help from the public cloud, depending on whether the mission cloud can judge that an outputted result is wrong, and the real-time constraint is not likely to be violated. The cloud phase transition mechanism is responsible for these judgments and for forwarding requests to the public cloud. This mechanism is discussed further in Subsections 4.1 and 4.2. (3) Object localization module. Based on the depth information (i.e., the distance) outputted by the robot sensors and the region proposal information provided by the recognition module, the positions of objects can be computed and object labels can be accurately marked on the map.

Public cloud. We adopted an existing Internet service, CloudSight, as the cloud service for re-recognizing the objects picked out by the cloud phase transition mechanism on the mission cloud.

At runtime, the geometric map of the environment is built continuously and collaboratively by the robot and the "mission cloud–public cloud" layer. The robot is responsible for collecting raw sensor data and the mission cloud is responsible for building the semantic map with the assistance of the public cloud. After the geometric map has been built, the semantic information of the objects (i.e., object labels) can be obtained through the collaboration of the Faster R-CNN/CORE module and CloudSight service. Simultaneously, the positions of the objects are computed by the object localization module and the labels are dynamically added to the geometric map.

4 Key mechanisms

Given that the geometric map can be accessed by the off-the-shelf SLAM algorithms, the greatest challenge in robotic semantic mapping is to obtain accurate semantic information. In our approach, we introduce object recognition based on Faster R-CNN/CORE to obtain the semantic labels of pre-trained 2D/3D objects in the mission cloud, as well as introduce the cloud phase transition mechanism to obtain the semantic labels of unpredictable and untrained objects in the public cloud. In addition, we briefly introduce the vision-based object positioning method to determine the locations of the objects on the geometric map.

4.1 Cloud phase transition in 2D object recognition

Object recognition is the process of obtaining the class name of an object, which is the basic semantic information of the object. Recently, with the development of machine learning, DNN has exhibited outstanding performance in object recognition tasks [15]. Multi-layer convolutional neural network (CNN) is the most prominent example. There has been a series of influential studies on 2D object recognition, including R-CNN [16], Fast R-CNN [17], and the latest version of Faster R-CNN [13]. The mean average precision (mAP) of Faster R-CNN can reach 69.9% on the PASCAL VOC 2007 dataset [18] and has a high frame rate of 5 fps on a GPU. Faster R-CNN has been the basis of several first-place entries in ILSVRC and COCO image competitions [13]. In addition, the mission cloud, with its powerful GPUs, huge memories and numerous processor cycles, are able to accelerate the operation of Faster R-CNN. Therefore, we adopted Faster R-CNN as the object recognition component on the mission cloud. To the best of our knowledge, our study is the first one to successfully apply Faster R-CNN to cloud robotic-based object recognition and environment modeling.

In the RoboCloud semantic mapping approach, the cloud phase transition mechanism plays an important role. The objective is to identify the images of objects that have been misrecognized by the mission cloud and determine if the specific real-time constraints would be violated if help were sought from CloudSight. If not, the relative image would be sent to CloudSight for re-recognition.

4.1.1 *Unfamiliar objects filtering*

To filter the unpredictable objects in an open environment, we propose two algorithms: probability score-based selection and “other-objects class” filtering.

Probability score-based selection. Unfamiliar objects are most likely to be misrecognized by Faster R-CNN on the mission cloud. During runtime, Faster R-CNN outputs the probability scores using its softmax classifier, which can also be used to select the unfamiliar objects.

For the softmax classifier, given an object x after the region proposal process and an object class set C containing K classes, for each $c \in C$, the probability score that the classifier calculates x belonging to class c is $P_c(x)$,

$$\sum_{c=1}^{K} P_c(x) = 1.$$

The final classification result of x is

$$f(x) = \arg\max_{c \in C} P_c(x).$$

The probability score of the final result reflects the recognition confidence. Thus, we define the maximum probability score as the recognition confidence:

$$\Psi(x) = P_{f(x)}(x).$$

On the basis of the recognition process, we introduce a constant $\Psi_{\rm thr}$ to illustrate a minimum threshold for the recognition confidence. When $\Psi(x) < \Psi_{\rm thr}$, we consider object x likely to be misrecognized to a great extent by Faster R-CNN. Then, this image region proposal is ready to transferred to CloudSight. The value of $\Psi_{\rm thr}$ can be estimated from the training set or can be set manually.

“Other-objects class” filtering. Although the above method can filter many misrecognized objects, it has no utility in the context of unpredictable objects whose categories have not been pre-trained, because the knowledge in the mission cloud has been previously built in, so its softmax classifier will classify an unfamiliar object into a “most like” object class within its knowledge boundary. As a result, a high probability score may be obtained, but the recognition result would definitely be wrong. Another case has objects with features that are very different from the pre-trained objects of the same class. For example, the "scissors" class may be trained by paper cutting scissors, pruning shears, clipper gauges, and tin snips. However, when an object in the image is a U-style thread clipper, the classifier component experiences difficulty in classifying the clipper into the "scissors" class. Instead, the clipper may be placed into another class, such as "chopsticks", with a high probability score.

To address such situations, we introduce another unfamiliar object filtering algorithm named "other-objects class" filtering, which was inspired by the idea of distinguishing targets from the background in visual classification problems [19]. Unlike a specific class, such as "scissors" or "apple", "other-objects class" contains many samples of heterogeneous objects, because the class represents "all other classes". Our objective is to distinguish the objects that do not belong to any of the classes determined in advance or are difficult to classify by the mission cloud. Hence, we add this special class to the object training set. The samples in this class can be randomly captured by the RGB camera from the real world or randomly chosen from 2D images of objects downloaded from the Internet. After the Faster R-CNN region proposal process, once an object is classified into the "other-objects class", this image region is considered to contain an object that is difficult for Faster R-CNN classifier to recognize, and so, the mission cloud will seek help from the public cloud Internet service.

4.1.2 *Real-time constraint violation judgment*

Specific real-time constraints should be taken into account during online robotic semantic mapping. However, Internet services may introduce unpredictable or unacceptable latencies. A real-time constraint violation judgment is also presented to maximize the benefits of seeking help and minimize the negative impacts, thereby staying with the real-time constraints.

An estimation of the latency test for recognition by CloudSight can be obtained from the historical sampling data. We collected 20 latencies of CloudSight per hour in one day and calculated the average latency as the corresponding estimation latency test of each hour. In the robotic semantic mapping process, we set a maximum tolerance latency $t_{\rm tol}$ of an object when requesting help from CloudSight. An unfamiliar object selected by the above mechanism did not require transfer to CloudSight if $t_{\rm tol} < t_{\rm est}$. However, if the CloudSight recognition result of the uploaded object (when $t_{\rm tol} \geqslant t_{\rm est}$) could not be returned in $t_{\rm tol}$ or CloudSight returned a status of "timeout" in $t_{\rm tol}$, the system no longer waited but assigned an "unknown" label to the object as the final result.

Combining the unfamiliar objects filtering and real-time constraint violation judgment algorithms, a robot's environment modeling capability can be boosted in an open and uncertain environment. We present a pseudocode description of the whole 2D object-level semantic information cognition process in Algorithm 1. The details of this process can be found in our previous study [20].

4.2 Cloud phase transition in 3D object recognition

Besides the RGB information, robots in the physical world can also obtain depth information with their sensors. Therefore, for 3D images of objects, we adopted CORE [14], an open-source object recognition engine designed specifically for cloud robotics, as the foundation of the recognition module on the mission cloud. CORE comprises an SVM classifier, as well as a set of data filters, feature descriptors, and other image processing libraries, for recognition. CORE is also trained by the images that have high probabilities of appearing in the environment.

The two algorithms discussed in Subsection 4.1 are also applied here. The real-time constraint violation judgment algorithm is the same as that introduced in Subsection 4.1.2. Since the classification method of CORE is SVM-based instead of DNN-based in Faster R-CNN, there are some differences in the respective mechanisms for filtering unfamiliar objects.

Vote-based selection. Here, we also want to measure the confidence of the object recognition results. Since CORE adopts an SVM classifier and utilizes a "one versus one" model [21] for multi-class classification, we propose a vote-based selection algorithm to select objects with high probabilities of being misrecognized. More specifically, the problem involving K classes is reduced to $K(K-1)/2$ (i.e.,

Algorithm 1 2D object-level semantic information cognition

```
INPUT: Scene image $x$ captured by the robot;
OUTPUT: Class label set $C$ describing the objects in an image and label $c$ for each object.
Use CNN to build feature map for the image;
Use RPN to obtain bounding-boxes for objects in the image;
Obtain features for each bounding-box $x$;
for each $x$ do
    $(c_{\text{mission}}, \Psi) = \text{Faster } R - \text{CNN}(x)$;
    if $c_{\text{mission}} \neq$ "other-objects class" and $\Psi \geqslant \Psi_{\text{thr}}$ then
        $c = c_{\text{mission}}$;
    else
        if $t_{\text{tol}} \geqslant t_{\text{est}}$ then
            if $\text{hasvalue}(x, t_{\text{tol}})$ then
                $c_{\text{public}} = \text{CloudSight}(x)$;
            else
                $c_{\text{public}} =$ "unknown";
            end if
            $c = c_{\text{public}}$;
        else
            $c = c_{\text{mission}}$;
        end if
    end if
end for
```

all possible pairs of classes) binary classification problems. The classification is based on a max-wins voting strategy. Given an image x and an object class set C, for each $c \in C$, the number of votes of x is

$$\text{vote}_c(x) = \sum h_{c'c}(x), \quad c' \in C - \{c\},$$

where $h_{c'c}(x) \in \{0, 1\}$, and 1 indicates that the binary classifier had assumed that image x belonged to class c instead of class c'. The final classification result of x is

$$f(x) = \arg\max_{c \in C} \text{vote}_c(x).$$

The number of votes of the final result reflects the confidence of the classifier. A class can obtain up to $K - 1$ votes, so we can define the confidence of the classification result as

$$\Psi(x) = \frac{\text{vote}_{f(x)(x)}}{K - 1}.$$

Then, a constant Ψ_{thr} is introduced to define a minimum threshold for the confidence. When $\Psi(x) < \Psi_{\text{thr}}$, we assume that image x is most likely to be misrecognized by CORE. In the experiment described in Section 6, we demonstrated that $\Psi_{\text{thr}} = 1$ was the best choice for our training set.

Background class filtering. The main idea of background class filtering is consistent with the "other-objects class" filtering algorithm described in Subsection 4.1.1. However, Faster R-CNN contains the object segmentation process and the generic objects in each image are selected initially from the background by the region proposal network (RPN). CORE directly takes the whole image of the object in front of the vision sensor into the recognition process. Besides the objects of other classes, "background class" also contains images of the real-world background, such as pictures of walls, lawns, and floors. Once an image is classified into this class, we assume that the image contains an object difficult for the mission cloud to recognize. The details can be found in our previous study [22].

4.3 Vision-based object positioning

The vision-based object positioning method aggregates the information about the depth, coordinates, and position of an object in an image, then computes the object's location on the map.

Acquiring an object's position relative to the robot. The RPN in Faster R-CNN estimates the positions of objects as pixel coordinates. The depth and angle information of an object to a robot can be obtained by the robot's visual sensors, such as Kinect[5]. We consider the coordinates of the center point as the object's position in a scene image. On the basis of the viewable range of Kinect and the image resolution, the angle of an object to a robot can be calculated in proportion. In addition, the depth from the object to the robot can be obtained from the Kinect information. CORE assumes that the object appears in front of Kinect, thereby simplifying the calculation.

Calculating an object's position on the map. The real-time position of the robot's pose is computed by the ROS transformer package using the robot's initial position, the map coordinate system, and the odometric data. The object's rotation angle is then computed from the quaternion of the robot. As a result, the object's position on a map is calculated by geometric and trigonometric functions.

5 Performance benefits of RoboCloud

The benefits of introducing the approach of RoboCloud into the object-level robotic semantic mapping process can be analyzed theoretically from two aspects.

5.1 Comparison with "robots + only mission cloud"

By adding the public cloud to the back end of the mission cloud, the robot's cognitive capability in an open environment can be improved. The final recognition accuracy of all misrecognized objects can be increased from 0 to $(1-\rho)P$, where P is the recognition accuracy of CloudSight and ρ is the false positive rate of the cloud phase transition.

The promotion of recognition accuracy is not evident for all types of images. The "collateral damage" of the cloud phase transition mechanism should be considered. For example, an object that has already been correctly recognized by the mission cloud may be mistakenly sent to the public cloud. Therefore, we will provide detailed analyses of this situation based on the DNN and SVM classifications for 2D and 3D images, respectively.

The results of Faster R-CNN are evaluated by mAP. C is the object class set and AP reflects the "order-matters recall" according to the probability scores, which range from high to low. In AP, m_j is the number of all relevant objects in class j. Precision(R_{jk}) at k is a percentage of the correct items among the first k recommendations when the recognition results belong to class j while object k is the real relevant object of class j. Assuming that the test set has s object classes, of which t classes are pre-trained by Faster R-CNN, the mAP for the test set is

$$\mathrm{mAP}(C)=\frac{1}{|C|}\left(\sum_{j=1}^{|C_t|}\frac{1}{m_j}\sum_{k=1}^{m_j}\mathrm{Precision}(R_{jk})+\sum_{i=1}^{|C_{s-t}|}\frac{1}{m_i}\sum_{k=1}^{m_i}\mathrm{Precision}(R_{ik})\right),$$

where $\sum_{i=1}^{|C_{s-t}|}\frac{1}{m_i}\sum_{k=1}^{m_i}\mathrm{Precision}(R_{ik})=0$ for Faster R-CNN, which has no knowledge of unpredictable objects. However, with the help of CloudSight and the cloud phase transition, many unfamiliar or misrecognized objects may be corrected. The collaborative mAP of our system is

$$\begin{aligned}\mathrm{mAP}(C)=&\frac{1}{|C|}\left(\sum_{j=1}^{|C_t|}\frac{1}{m_j}\sum_{k=1}^{m_j}\mathrm{Precision}(R_{jk})+\sum_{i=1}^{|C_{s-t}|}\frac{1}{m_i}\sum_{k=1}^{m_i}\mathrm{Precision}(R_{ik})\right)\\=&\frac{1}{|C|}\left(\sum_{j=1}^{|C_t|}\frac{1}{m_j}\sum_{k=1}^{m_j}(M_{jk}(1-\varrho_{jk})+M_{jk}\varrho_{jk}P+(1-M_{jk})(1-\rho_{jk})P)\right)\\&+\sum_{i=1}^{|C_{s-t}|}\frac{1}{m_i}\sum_{k=1}^{m_i}((1-\rho_{jk})P),\end{aligned}$$

[5] https://en.wikipedia.org/wiki/Kinect.

where M_{jk} is the abbreviated representation of Precision(R_{jk}), ρ_{jk} and ρ_{ik} are the false positive rates of the transition mechanism for the pre-trained object class j and unfamiliar class i, respectively, while ϱ_{jk} represents the false negative rate.

Theorem 1. The cognitive capability (i.e., average precision) for unfamiliar object class i can be improved if $\rho_{jk} < 1$.

The collaborative "mission cloud–public cloud" model may perform better as long as there are objects picked out by the mechanism and CloudSight is able to identify them to a certain extent.

Theorem 2. The cognitive capability (i.e., average precision) for objects whose categories have been pre-trained can be improved if

$$\sum_{k=1}^{m_j}(M_{jk}(1-\varrho_{jk}) + M_{jk}\varrho_{jk}P + (1-M_{jk})(1-\rho_{jk})P) > \sum_{k=1}^{m_j} M_{jk}.$$

In contrast to that of Faster R-CNN, CloudSight may provide lower accuracy for the expectable objects, so the cognitive capability of the collaborative "mission cloud–public cloud" model may also be reduced mainly when Faster R-CNN's performance is sufficiently outstanding and the false negative rate ϱ_{jk} is relatively high.

Theorem 3. The total cognitive capability (i.e., mAP) for all objects in an open uncertain environment can be improved if

$$\sum_{j=1}^{|C_t|}\frac{1}{m_j}\sum_{k=1}^{m_j}(M_{jk}(1-\varrho_{jk}) + M_{jk}\varrho_{jk}P + (1-M_{jk})(1-\rho_{jk})P) + \sum_{i=1}^{|C_{s-t}|}\frac{1}{m_i}\sum_{k=1}^{m_i}((1-\rho_{jk})P)$$
$$> \sum_{j=1}^{|C_t|}\frac{1}{m_j}\sum_{k=1}^{m_j} M_{jk}.$$

We denote the accuracy of CORE on the mission cloud as M and the false negative rate of the transition mechanism as ϱ. Then, the total accuracy H can be calculated as

$$H = M(1-\varrho) + M\varrho P + (1-M)(1-\rho)P.$$

Through a simple deformation, Theorem 4 can be obtained.

Theorem 4. Recognition accuracy $H \geqslant M$ if and only if $P \geqslant \frac{M_\varrho}{M_\varrho + (1-M)(1-\rho)}$.

With the help of cloud services and the cloud phase transition mechanism, a large proportion of unfamiliar objects can be corrected from a statistical point of view, and the performance promotion would be greater with more openness and uncertainty in the environment.

5.2 Comparison with "robots + only public cloud"

The objective of introducing the mission cloud is to optimize the QoS requirement, particularly, the latency for the objects familiar to the mission cloud. The total latency of the objects correctly recognized by the mission cloud can be calculated by

$$l_{m\_\mathrm{correct}} = l_m + l_p\varrho,$$

where l_m is the average latency of the mission cloud and l_p is that of the public cloud service. We can obtain Theorem 5.

Theorem 5. If $\varrho < 1 - \frac{l_m}{l_p}$, then $l_{m\_\mathrm{correct}} < l_p$.

l_m is typically significantly less than l_p in practice, because the mission cloud is deployed near the robots and the public cloud is on the Internet. As a reference, l_m/l_p in the experiments described in the next section is frequently below 0.1. Consequently, $1 - l_m/l_p$ is usually close to 1, and the final average latency for the object classes that have been trained on the mission cloud is certainly nearly less than that of the public cloud.

6 Experiments and evaluation

We deployed a DELL PowerEdge R730 server that directly connects to the robot via Wi-Fi on the mission cloud. A TurtleBot, a wheeled mobile robot with a Kinect, was adopted as the robotic platform to move within the test environment. We focused on the performance benefits of RoboCloud in the experiments based on Faster R-CNN and CORE on the mission cloud, respectively. The results are analyzed for both the open 2D/3D datasets and a real robotic scenario.

6.1 Experiments on open datasets

We test the object-level environment cognitive capability on the PASCAL VOC 2007 dataset for Faster R-CNN-CloudSight and on the Washington dataset for CORE-CloudSight.

6.1.1 *Experiments on PASCAL VOC* 2007 *dataset for Faster R-CNN-CloudSight*

The actual benefits of introducing an Internet-based object recognition service while setting a mission cloud were evaluated by experiments on the PASCAL VOC 2007 dataset [18], which consisted of 9963 images with 24640 objects labeled in 20 categories.

Accuracy promotion for the trained categories. To evaluate the cognitive capability (i.e., mAP) promotion for unseen object instances where there are no unfamiliar object classes in the test environment, we trained Faster R-CNN using the train-validation part of the dataset, which included 5011 images with 12608 objects labeled in 20 categories and left the remainder of the dataset as the test set. The "other-objects class" was constructed by randomly downloading 500 RGB images with the objects bounding-boxes from the Internet. The recognition confidence threshold $\Psi_{\rm thr}$ was set to 0.7. The mAP result was obtained by the RoboCloud collaborative system (86.3%), which displayed a promotion of 16.4% over the "robots + only mission cloud" model (69.9%). In particular, the probability score-based selection mechanism contributed 12.7%, while the "other-objects class" filtering mechanism contributed 3.7%.

Accuracy promotion for unfamiliar objects. To evaluate the cognitive capability promotion when unfamiliar objects exist and the real-time requirement is not strict, we kept the test set unchanged and calculated the mAP of the 20 object categories while the training set was reduced by $1, 2, \ldots, 10$ object classes in turn. As shown in Figure 3(a), the mAP of the "robots + only mission cloud" declined dramatically as the proportion of unfamiliar objects in the test set increased because the recognition component had no knowledge of the untrained object classes. After introducing CloudSight to Faster R-CNN on the mission cloud, the mAP performance was considerably high and stable. The mAP promotion was expected to become larger when more unfamiliar objects were encountered.

Recognition latency optimization. Compared with the "robots + only public cloud" model, the advantage of our approach is latency optimization, particularly for objects trained in advance on the public cloud. We repeated the last experiment and recorded the average latency instead of accuracy. As shown in Figure 3(b), the latency of the cloud service was relatively high (12.68 s on the average) and highly unstable. The behavior of Faster R-CNN on the mission cloud was significantly more predictable and its latency was always low (approximately 0.07–0.1 s). Finally, the average latency of RoboCloud varied from 2.27 to 4.38 s. With the help of the Internet service and mechanism, RoboCloud obtained relatively low recognition latency while maintaining relatively high accuracy.

Variables related to cloud phase transition mechanism. Experiments were conducted to evaluate the effects of the variables related to the cloud phase transition mechanism. The two main variables are the confidence threshold $\Psi_{\rm thr}$ and the number of unfamiliar objects in the test set. Figure 3(c) presents the mAPs with different proportions of unfamiliar objects when $\Psi_{\rm thr}$ was varied from 0.5 to 0.9. It is obvious that regardless of the number of unfamiliar objects, the mAP continued to grow as $\Psi_{\rm thr}$ increased from 0.5 to 0.7, but slightly declined as it increased from 0.7 to 0.9. Hence, the optimal value of $\Psi_{\rm thr}$ is 0.7 for this dataset.

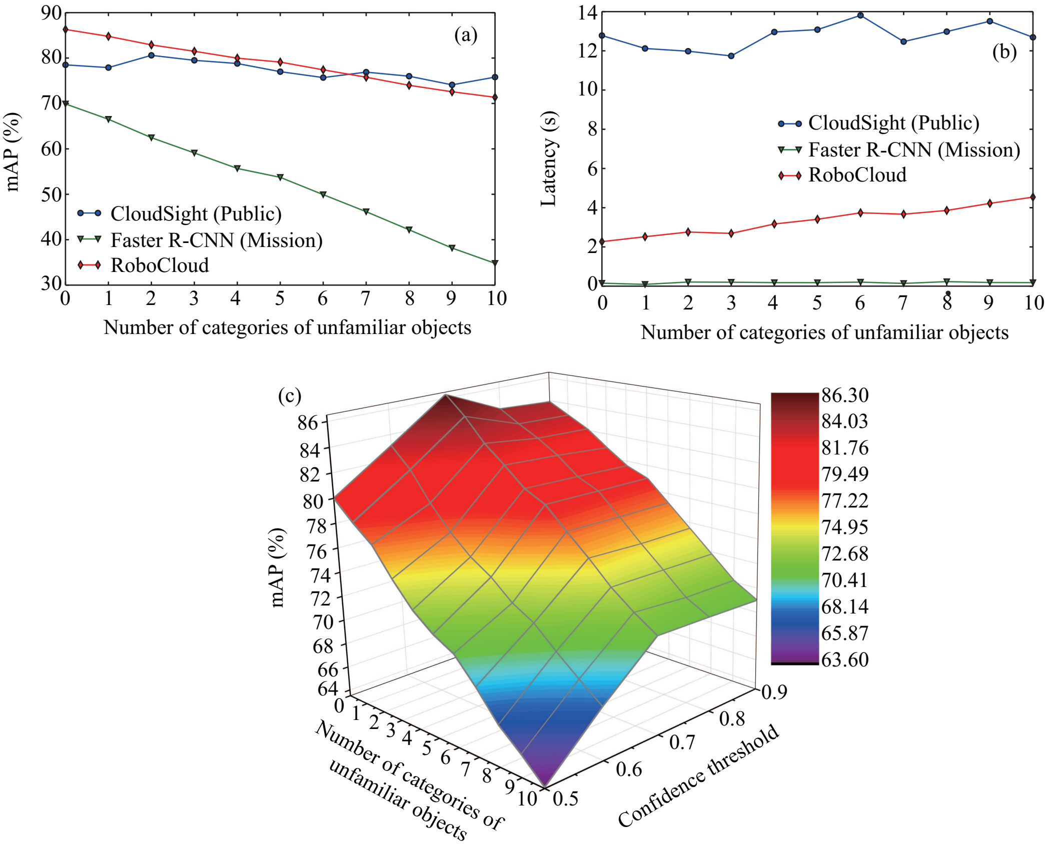

Figure 3 (Color online) Experiments for Faster R-CNN-CloudSight. (a) mAP with unfamiliar objects; (b) latency with unfamiliar objects; (c) mAPs with combinations of variables.

6.1.2 *Experiments on Washington dataset for CORE-CloudSight*

The Washington RGB-D object dataset for robotic object recognition consists of 300 common household objects (51 categories) with 41877 RGB-D images captured from different angles. The main purpose of the experiments described in this subsection is consistent with that of the experiments described in Subsection 6.1.1. In this part, we briefly discuss the results. Further details can be found in our previous study [22].

For the promotion of the accuracy of the trained categories, we trained 40 categories and left one object in each category for testing. The background class was constructed with images from the other 8 categories in the dataset and 800 RGB-D images randomly downloaded from the Internet. The confidence threshold $\Psi_{\rm thr}$ in the vote-based selection mechanism was set to 1. During the test, RoboCloud exhibited a significant accuracy advantage of 15.3% over CORE (only the mission cloud). In particular, the vote-based selection contributed 5.2%, whereas the background class filtering contributed 10.1%. As for the unfamiliar objects, we randomly picked 1 to 19 objects from those three categories and added them to the testing set. As shown in Figure 4(a), the performance of RoboCloud was also considerably more stable than that of the "robots + only mission cloud" model. Figure 4(b) shows the recognition latency optimization. In Figures 4(a) and (b), RoboCloud exhibits relatively low latency and high accuracy. Figure 4(c) shows the results of testing $\Psi_{\rm thr}$. The accuracy increased sharply when the value of $\Psi_{\rm thr}$ was set to 1.

6.2 Experiment in a real environment

We demonstrated how a robot efficiently builds a semantic map in an open environment containing unfamiliar objects. Here, we mainly discuss the results and analyses of the Faster R-CNN-CloudSight

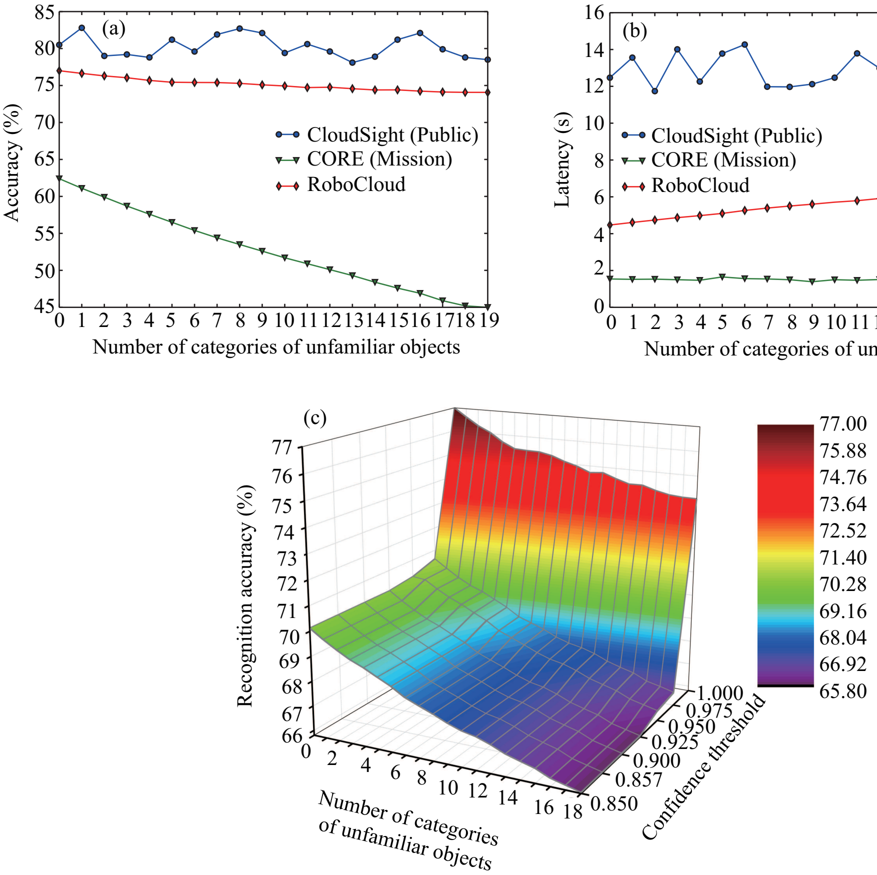

Figure 4 (Color online) Experiments for CORE-CloudSight. (a) Accuracy with unfamiliar objects; (b) latency with unfamiliar objects; (c) accuracies with combinations of variables.

RoboCloud system on semantic mapping with Faster R-CNN's having processed the object segmentation in each image captured by the TurtleBot. We also briefly discuss the results of the CORE-CloudSight model.

We set the speed of Turtlebot to 0.3 m/s and teleoperated it to move around while taking one photograph every 5 s. Given that some images may contain the same objects, we measure the distances between every two instances of objects appeared in the two adjacent pictures. If the distance was less than 0.5 m, we assumed that these two objects referred to the same object and marked the label which has the highest number of occurrences on the object's position on the map. Figure 5 depicts the laboratory environment from the TurtleBot's perspective and the mapping process. For the test, we chose 54 objects in 20 classes, of which 9 object classes had been pre-trained. Figure 6(a) shows the semantic map based on only Faster R-CNN without the real-time constraint. Misrecognized objects are indicated by the red text in the solid line boxes. Figure 6(b) shows the results from RoboCloud. CloudSight re-recognized 20 objects (indicated by the blue text in the dotted line boxes). With the help of the Internet service and RoboCloud mechanism, the accuracy of the test environment model was promoted from 44.4% (with only the mission cloud) to 78.0%.

The real-time constraint violation judgment mechanism was evaluated by setting different values of maximum tolerance latency $t_{\rm tol}$ in the real robotic scene. The estimation of the latency $t_{\rm est}$ for recognition by CloudSight was 12.81 s. Figure 7 shows that when $t_{\rm tol} < t_{\rm est}$, no objects will be uploaded to CloudSight. Consequently, the mAP is the same as that of only the mission cloud. When $t_{\rm tol} \geqslant t_{\rm est}$, the objects picked out will be uploaded. However, if the result from CloudSight is not returned in $t_{\rm tol}$, an "unknown" label will be assigned to the result, so some objects may not receive the proper semantic information. As the value of $t_{\rm tol}$ gradually increases, the mAP is relatively high and stable when more unfamiliar objects are

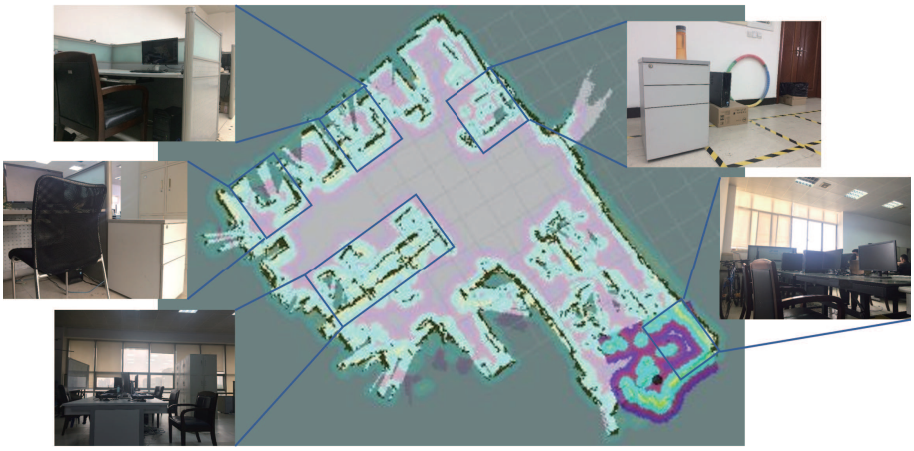

Figure 5   (Color online) Test environment from the TurtleBot's perspective.

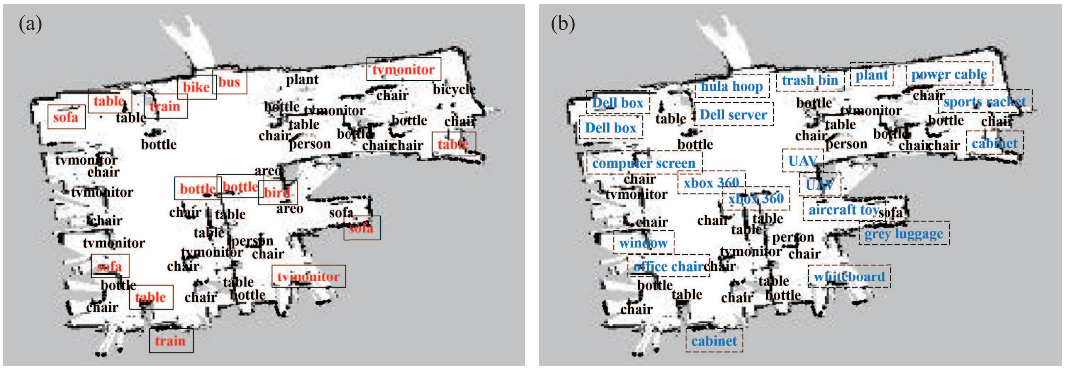

Figure 6   (Color online) Semantic mapping for Faster R-CNN-CloudSight. (a) Semantic map (only Faster R-CNN); (b) semantic map (Faster R-CNN and CloudSight).

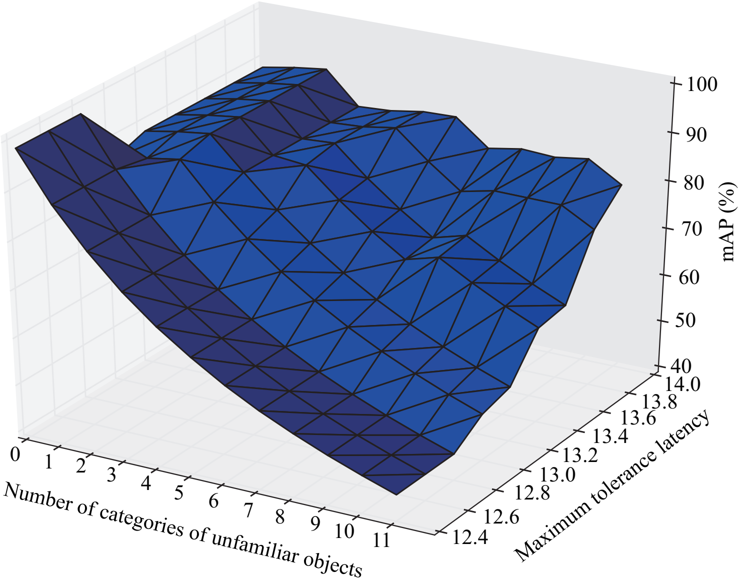

Figure 7   (Color online) mAP considering real-time constraints.

involved. Therefore, for the expectable object classes, Faster R-CNN can provide compelling real-time assurance. For unfamiliar objects, a trade-off between accuracy and real-time properties should be made in advance.

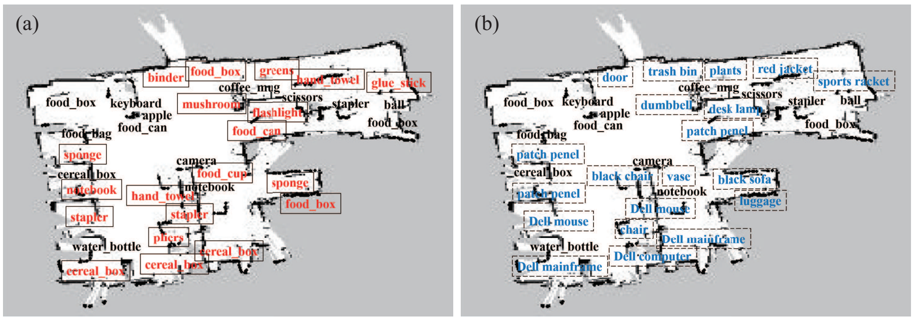

Figure 8 (Color online) Semantic mapping for CORE-CloudSight. (a) Semantic map (only CORE); (b) semantic map (CORE and CloudSight).

Figure 8 shows the effects of the CORE-CloudSight RoboCloud system. Figure 8(a) depicts the results of environment modeling at the object level based on only mission cloud (CORE trained by the Washington dataset) while Figure 8(b) shows the semantic map combining CORE and CloudSight.

7 Conclusion

We have discussed the problem of robotic modeling in an open environment and showed that the knowledge accumulated on the Internet could assist a robot, which is a typical cyber-physical system, to cope with uncertainty, such as the unpredictable objects in the real world. We also proposed a novel environment modeling approach named RoboCloud for acquiring knowledge dynamically from the Internet when a robot encounters an unexpected object. A mission cloud was also introduced for its adoption of Faster R-CNN or CORE to recognize the expected 2D or 3D objects highly related to the modeling mission and for the obtaining of semantic labels, which can make use of rich computing resources on the mission cloud for real-time and highly accurate recognition. If an object class is unpredictable in an open environment and beyond the knowledge of the mission cloud, we resort to the public cloud services on the Internet (e.g., the mature Internet-based image recognition services for re-recognition). With a carefully designed cloud phase transition mechanism and related algorithms, our approach can maximize the benefits of introducing Internet services and minimize the negative impacts on QoS properties, which are essential to many robotic applications. In future research, we will apply the architecture of this approach to other cloud robotic software applications.

Acknowledgements This work was partially supported by National Natural Science Foundation of China (Grant Nos. 91118008, 61202117, 61772030), Special Program for the Applied Basic Research of National University of Defense Technology (Grant No. ZDYYJCYJ20140601), and Jiangsu Future Networks Innovation Institute Prospective Research Project on Future Networks (Grant No. BY2013095-2-08).

References

1 Nüchter A, Hertzberg J. Towards semantic maps for mobile robots. Robot Auton Syst, 2008, 56: 915–926
2 Wolf D F, Sukhatme G S. Semantic mapping using mobile robots. IEEE Trans Robot, 2008, 24: 245–258
3 Fulcher J. Computational Intelligence: An Introduction. Berlin: Springer, 2008
4 Kehoe B, Patil S, Abbeel P, et al. A survey of research on cloud robotics and automation. IEEE Trans Autom Sci Eng, 2015, 12: 398–409
5 Riazuelo L, Tenorth M, Marco D D, et al. RoboEarth semantic mapping: a cloud enabled knowledge-based approach. IEEE Trans Autom Sci Eng, 2015, 12: 432–443
6 Satyanarayanan M, Bahl P, Caceres R, et al. The case for VM-based cloudlets in mobile computing. IEEE Pervas Comput, 2009, 8: 14–23
7 Furrer J, Kamei K, Sharma C, et al. Unr-pf: an open-source platform for cloud networked robotic services. In: Proceedings of IEEE/SICE International Symposium on System Integration, Fukuoka, 2012
8 Kostavelis I, Gasteratos A. Semantic mapping for mobile robotics tasks: a survey. Robot Auton Syst, 2015, 66: 86–103
9 Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I. IEEE Robot Autom Mag, 2006, 13: 99–110

10 Mohanarajah G, Hunziker D, D'Andrea R, et al. Rapyuta: a cloud robotics platform. IEEE Trans Autom Sci Eng, 2015, 12: 481–493

11 Ashutosh S, Ashesh J, Ozan S, et al. Robobrain: large-scale knowledge engine for robots. 2014. ArXiv:1412.0691

12 Qureshi B, Javed Y, Koubâa A, et al. Performance of a low cost Hadoop cluster for image analysis in cloud robotics environment. Procedia Comput Sci, 2016, 82: 90–98

13 Ren S Q, He K M, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intel, 2017, 39: 1137–1149

14 Beksi W J, Spruth J, Papanikolopoulos N. Core: a cloud-based object recognition engine for robotics. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, 2015

15 Szegedy C, Toshev A, Erhan D. Deep neural networks for object detection. Adv Neural Inf Process Syst, 2013, 26: 2553–2561

16 Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014. ArXiv:1311.2524

17 Girshick R. Fast R-CNN. 2015. ArXiv:1504.08083

18 Everingham M, Van Gool L, Williams C K I, et al. The pascal visual object classes (VOC) challenge. Int J Comput Vis, 2010, 88: 303–338

19 Torralba A, Murphy K P, Freeman W T. Shared features for multiclass object detection. In: Toward Category-Level Object Recognition. Berlin: Springer, 2006. 345–361

20 Li Y Y, Wang H M, Ding B, et al. Learning from internet: handling uncertainty in robotic environment modeling. In: Proceedings of the 9th Asia-Pacific Symposium on Internetware, Shanghai, 2017

21 Duan K B, Keerthi S. Which is the best multiclass svm method? an empirical study. In: Proceedings of International Workshop on Multiple Classifier Systems, Seaside, 2005. 278–285

22 Li Y Y, Wang H M, Ding B, et al. Toward qos-aware cloud robotic applications: a hybrid architecture and its implementation. In: Proceedings of IEEE Conferences on Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, Toulouse, 2017