Semantic Perception, Mapping and Exploration

April 17, 2018

Introduction

Related Work

Acquisition and modeling of semantic information is a key requisite for mobile robots to be deployed in human environments. In this field, fundamental aspects faced by research are: the recognition of places and objects, the construction of semantic models and the exploration strategies to enrich contextual knowledge. In the remainder of this section, we will present relevant work that focused on the mentioned problems, namely: semantic perception, semantic mapping and semantic exploration.

Semantic Perception

Extracting semantic information from visual data is one of the fundamental problems of computer vision. Scene understanding can be decomposed into sub-tasks, depending on the information one is interested to extract from the input data. These sub-tasks can be organized on a progression that goes from coarse to fine grained inference.

Image classification is the task of assigning a semantic label to an input image from a fixed set of categories. Ulrich and Nourbakhsh [?] propose an appearance-based place recognition system for topological localization. They use colour histogram features [?] and a simple voting scheme for nearest-neighbor matching. In a similar fashion, Torralba *et al.* [?] derive an hidden Markov model (HMM) for place recognition and new place categorisation based on the global statistic feature retrieved from texture [?]. In contrast, Lisin *et al.* [?] propose to model classes of images as a probability distribution over local features, in order to be combined with global features. This method has proven to perform well in applications where a rough segmentation of objects is available.

Object detection consists of making a prediction not only of object categories but also of their spatial locations. A seminal work can be considered that of Viola and Jones [?], who proposed a fast and robust face detection. Their method makes use of Haar-like features [?] to search for likely face candidates, which can then be refined using a cascade of more expensive but selective detection algorithms [?]. Likewise, a well-known example of pedestrian detection has been proposed by Dalal and Triggs [?], who use a set of overlapping Histogram of Oriented Gradients (HOG) descriptors fed into a Support Vector Machine (SVM) [?].

Image segmentation is the task of finding groups of pixels that possess some ”similarity” and is one of the oldest and most widely studied problems in computer vision. Early techniques focus on local region merging and splitting [?, ?], while, more recent algorithms often optimize some global criterion, such as intra-region consistency and inter-region boundary lengths or dissimilarity [?, ?, ?, ?, ?].

Despite the popularity of the presented methods, a recent breakthrough in scene understanding has been the adoption of Convolutional Neural Networks (CNNs) [?]. Krizhevsky *et al.*in [?] present the pioneering deep CNN that, despite its simplicity, won the Imagenet 2012 classification challenge with wide margin on the closest competitor. Similarly, different object detection methods based on deep neural netowrks have shown to outperform the state-of-the-art [?, ?, ?]. Consequently, the capabilities of such networks have been also investigated in pixel-level labeling problems like semantic segmentation. In this context, a milestone is the work of Long *et al.* [?] who transformed existing classification models ([?, ?]) into fully convolutional ones to output spatial maps instead of classification scores. One of the main reason behind its popularity is that, with this approach, CNNs can be trained end-to-end and efficiently learn to make dense predictions with inputs of arbitrary size.

Semantic Mapping

Building a digital representation of the environment where the robot has to operate, is a well studied problem in both robotics and AI communities (citations needed). In literature, different types of representations have been proposed, as well as different methods to obtain each of them. It is possible to classify these representations by the type of information they convey to the robot, i.e.: metric, that can be used to measure physical quantities in the environment such as distance to obstacles; topological, that allows the robot to assess the connectivity between places and/or objects in the environment and semantic, that tells the robot to which category the objects in the map belong to. Traditional approaches dealt with the reconstruction of metric and topological information, whereas, recent years have seen growing interest in enriching the existing representations with semantics [?, ?]. This involves segmenting a scene by assigning a label to the extracted features and mantaining the correspondence with already seen features to integrate the current measurement in a globally consistent model.

Hermans et al., in [?], address the problem of dense semantic segmentation of 3d point clouds. Here, the term ”dense” indicates that the proposed method works with the whole measurement returned by the sensor, thus, each feature is a single point of the 3d point cloud obtained from the RGB-D camera. In this work, the authors propose a mapping scheme which is made of three components: a Randomized Decision Forest (RDF) to obtain a 2D segmentation of the RGB image, a 3D scene reconstruction to obtain a geometric model of the environment and the correspondence between features of two consecutive frames and, finally, a Conditional Random Field [**?**] to properly assign the semantic labels of the 2d segmentation to the 3d points of the geometric model. A significant contribution of this work is that the proposed pipeline can meet real-time requirements, however, segmentation results are not comparable with offline methods.

McCormac et al. [**?**] follow a similar approach to the one described so far, with the difference that they use a Convolutional Neural Netowork [**?**] to obtain a semantic segmentation of the RGB image. This clearly improves classification results, as CNNs have shown impressive performance on such tasks, however still it's not clear how a semantically annotated 3d point cloud can be used by the robot for its purposes. This is because a point cloud is an unstructured representation, i.e. no connectivity information is provided, making it not suitable for tasks like path planning and execution.

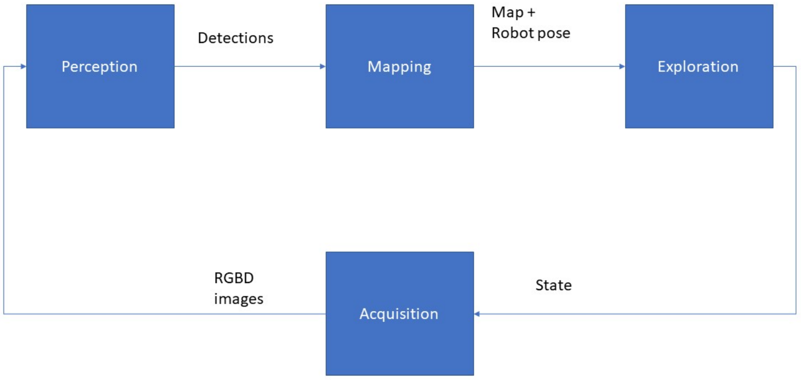
In (cite), Gunther et al. consider a different model to represent the environment, that is, instead of using a point cloud, which can be seen as a low-level representation, their model is directly made of objects. To build such a model, the authors use a 3d object recognition system, which, given a 3d point cloud, returns the set of objects detected in the scene in terms of their 3d pose, bounding box and semantic label. Moreover they propose an effective segmentation scheme that takes advantage of the contextual relations inherent to human-made spaces in order to improve classification results. This is achieved by building a graph, from the objects in the scene, which is exploited by a CRF to smooth the output of the object recognition. The main drawback of this method, is that the object recognition method guarantees reasonable results only under strong assumptions, that is, the scene observed by the robot must be a table, the objects in the scene must be already known and a CAD model for each of them must be available.

A possible solution to this shortcoming is to keep the map representation proposed in Gunther et al., while using a more sophisticated method to segment the scene. Nowadays, CNNs are proving to be an effective solution for visual problems like that, thus, it's a reasonable idea to consider their use in such a context.

Semantic Exploration

Technical Section

Our pipeline is made of three modules that exchange data structures among each other to build and maintain a semantic map. The perception module is in charge of classifying objects seen by the robot, it receives a pair of RGB-D



images and returns the set of detections. Consequently, the mapping module builds a local map from the objects detected by the perception module and updates the global map with the incoming information. Finally, the exploration module takes as input the updated global map and current robot pose and finds the best view to improve robot knowledge of the environment.

Perception

Semantic segmentation can be regarded as the task of localizing and classifying objects in the scene to label each image pixel with its semantic class. More formally, given an image $I \subset \mathbf{N}^{3\times2}$ and a set of semantic labels $\mathcal{L} = \{l_i\}_{i=1\ldots L}$, semantic segmentation computes an assignment of labels $l_i \in \mathcal{L}$ to each image pixel $x = (u, v) \in I$.

In a simulated environment it's possible to execute this task by performing only geometrical computations, since one knows the pose of the objects and the robot in the world. A virtual camera can be defined through its field-of-view, thus, inferring which objects it's framing can be done through view frustum culling (cite), as shown in (fig). After this step, it's necessary to decide which elements in the view frustum are visible by the robot, and which are hidden. This is a well-known problem in Computer Graphics and can be solved with Z-buffer algorithm (cite). Since we know that the robot is equipped with a visual sensor, we can assume that the depth image $D \subset \mathbf{R}^2$ is indeed a Z-buffer, thus, we use this information for occlusion check. First, each model 3D bounding box $BB =< min, max >\in \mathbf{R}^{3\times2}$ is transformed in camera frame and min,max are recomputed. Then, occlusion check is performed by counting for each model how many times it's voted, i.e. x in BB, by a point of the depth image x (binning). In this way, models that get no votes are hidden to the robot. A model that passes this step is upgraded to a detection <Type,Position2D,Size2D>.

```
Data: $\mathcal{T}$
Result: $\mathcal{D}$
for $m_i \in \mathcal{M}$ do
    $m_i \leftarrow T * mi_i$
end
for $x \in D$ do
    $m_i \leftarrow$ voteModel($x$)
    if !($m_i \in \mathcal{D}$) then
        $d \leftarrow$ detect($m_i$)
        $\mathcal{D} \leftarrow$ insert($d$)
    end
    $d \leftarrow$ update($x$)
end
```

Algorithm 1: Semantic Segmentation

Mapping

A key requirement for a "mobile and autonomous" robot is the capability of autonomously exploring and manipulating in human environments. To perform these tasks the robot needs a representation of the environment. In human environments, robots have to build autonomously such representation. As discussed in sec. (related), a suitable representation for this type of robotic applications should contain object-level features to encode also semantic information.

The process of building a semantic map consists in integrating into the robot digital representation the sequence of measurements acquired with on-board sensors. As a first step, objects are extracted from the depth image and assembled in a local map. Then, the update of the map can be done by registering the current frame with a global model. To this end, it's necessary to know the pose of the robot relative to the global reference frame. This information may be provided by SLAM systems, while, in simulation environments is given by default. Once objects are transformed in global frame, data association is performed to find correspondences between local and global map (object tracking) and update the global map accordingly.

Our system of object tracking has the task of identifying the occurrence of scene features (in our case objects) across different views of the scene. This is typically performed in two steps. 1) feature extraction, build object from detection, 2) data-association, finds corresponding objects between global and local map.

Feature extraction is performed by taking the output of the perception module and building an object (atomic entity of the map) from each detection. After that, by knowing the pose of the robot in the global frame, it's possible to transform the extracted objects in the global frame to find correspondences.

Data association finds which objects in the global map correspond to the one observed in the local map. This correspondence is evaluated through a similarity measure. So that, finding a corresponding object can be casted into a nearest-neighbor search. Usually, similarity is computed through euclidean distance.

In our problem, it's possible to leverage also semantic information. In case of a limited number of objects, brute-force is a reasonable technique to choose. In the simulated environment, the knowledge of the semantic class is perfect. In real world scenarios, object detectors return a probability distribution over semantic labels. In the first case, the corresponding object is the closest one with the same type. In the second case, euclidean distance could be weighted with class probabilities to represent a form of similarity measure.

Map update is performed by consistently integrating incoming information from local map into global map. Input: maps, correspondences. Output: updated global map. Steps: for each object in local map check if it's already been seen, yes: merge, no: add to map. This strategy may fail in case of false positives returned by data association. Possible solution: aging scheme.

References