Simultaneous Exploration and Segmentation with Incomplete Data

Tomáš Petříček[12]
http://cmp.felk.cvut.cz/~petrito1
Vojtěch Šalanský[1]
http://cmp.felk.cvut.cz/~salanvoj
Karel Zimmermann[1]
http://cmp.felk.cvut.cz/~zimmerk
Tomáš Svoboda[12]
http://cmp.felk.cvut.cz/~svoboda

[1] Center for Machine Perception,
Department of Cybernetics,
Faculty of Electrical Engineering,
Czech Technical University in Prague

[2] Czech Institute of Informatics, Robotics and Cybernetics,
Czech Technical University in Prague

Abstract

We consider the problem of pan-tilt sensor control for active segmentation of incomplete multi-modal data. Since demanding optimal control does not allow for online replanning, we rather employ the optimal planner offline to provide guiding samples for learning a CNN-based control policy in a guided Q-learning framework. The proposed policy initialization and guided Q-learning avoids poor local optima and yields reasonable results already from hundreds of roll-outs. The results suggest that the proposed policy outperforms a baseline and is suitable for real-time control.

1 Introduction

Object detection in an unknown environment from sensory data captured by a mobile robot is crucial for many applications including Search & Rescue (SAR) missions. In a typical SAR scenario, a human operator or a global planner provides a coarse exploration path along which the measurements are to be collected, registered and processed. Since most of the sensors have a limited field of view and the exploration time is typically constrained, resulting coverage of the environment by the sensors may be incomplete, which may decrease the performance of object detection. When sensors are located on pan-tilt units, the dimensionality of the exploration planning is huge and does not allow for real-time replanning when new data arrive. We propose a novel reactive control of body-mounted mobile sensors for accurate classification of data gathered along a given exploration path. We call this problem *simultaneous exploration and segmentation with incomplete data* (SES).

Convolutional Neural Networks (CNN) has recently been shown to be powerful representation for both classification [1] and control [2], which however require either (i) a large number of labeled training examples [1, 4], or (ii) a careful initialization [2, 3] to learn properly. We show that in contrast to a general reinforcement learning task, the structure of SES allows for efficient policy initialization and guided learning to avoid poor local optima and to yield reasonable results already from hundreds of roll-outs. The method is verified on the problem of victim segmentation on a mobile SAR robot.

2 Approach

The sensory suite of our mobile robot consists of (i) the Point Grey Ladybug 3 panoramic camera providing `RGB` images, (ii) the SICK LMS-151 laser scanner on a rotating mount providing depth images `D`, and (iii) the thermal camera IMAGER TIM 160 with a small field of view mounted on a pan-tilt unit and providing thermal measurements `T`. Since temperature is an important cue for detecting victims in SAR, an intelligent control of the pan-tilt unit is needed for compensating the limited sensor coverage and maintain accurate detection.

We design a (re)active victim segmentation algorithm in which CNNs simultaneously segment victims in incomplete `RGBDT` 3-D data and control pan of the thermal camera to minimize segmentation error.

Let us assume that the robot follows a known short-horizon path into an unknown environment. Along the path, the temperature can be measured at some parts of the environment, namely at some *voxels*. Our instance of the SES problem is defined as the classification of all voxels visible from the panoramic sensor and simultaneous control of the thermal camera which yields low classification error.

In the following, the problem is defined more formally. Let us denote $\mathcal{V}(i_1,\dots,i_K)$ the set of the voxels visible by the thermal camera from viewpoints $i_1,\dots,i_K$ captured at K positions along the path (i.e., finite horizon is assumed). We also define $\varepsilon(v)$ to be the expected classification error when voxel v is classified based on `RGBD` data only and $\varepsilon_{\mathrm{T}}(v)$ to be the expected classification error when voxel v is classified based on `RGBDT` data. We assume that the motion dynamics of the thermal camera is constrained and that viewpoint i_k at time k is given as $i_k = f(i_{k-1}, u_k)$, where f is the motion model and u_k is a control action at time k. We approximate the active segmentation task as the solution to the following problem:

$$\begin{aligned}\underset{u_1,\dots,u_K}{\arg\min}\Big[\sum_{v\in\mathcal{V}(i_1,\dots,i_K)}\varepsilon_{\mathrm{T}}(v)+\sum_{v\notin\mathcal{V}(i_1,\dots,i_K)}\varepsilon(v)\Big]\\ \text{s.t. } i_k = f(i_{k-1},u_k)\quad \forall_{k\in\{1,\dots,K\}},\quad i_0=\text{const.},\end{aligned}$$

where i_0 is an initial viewpoint. This problem is rewritten in the more suitable way as follows:

$$\begin{aligned}&\underset{u_1,\dots,u_K}{\arg\min}\Big[-\sum_{v\in\mathcal{V}}\varepsilon(v)+\sum_{v\in\mathcal{V}(i_1,\dots,i_K)}\varepsilon_{\mathrm{T}}(v)+\sum_{v\notin\mathcal{V}(i_1,\dots,i_K)}\varepsilon(v)\Big]=\\ &\qquad\text{s.t. } i_k = f(i_{k-1},u_k)\quad \forall_{k\in\{1,\dots,K\}},\quad i_0=\text{const.},\\ &=\underset{u_1,\dots,u_K}{\arg\min}\sum_{v\in\mathcal{V}(i_1,\dots,i_K)}\Big[\varepsilon_{\mathrm{T}}(v)-\varepsilon(v)\Big]=\\ &\qquad\text{s.t. } i_k = f(i_{k-1},u_k)\quad \forall_{k\in\{1,\dots,K\}},\quad i_0=\text{const.},\\ &=\underset{u_1,\dots,u_K}{\arg\max}\sum_{v\in\mathcal{V}(i_1,\dots,i_K)}\Delta\varepsilon(v),\\ &\qquad\text{s.t. } i_k = f(i_{k-1},u_k)\quad \forall_{k\in\{1,\dots,K\}},\quad i_0=\text{const.},\end{aligned}$$

where $\Delta\varepsilon(v)=\varepsilon(v)-\varepsilon_{\mathrm{T}}(v)$ denotes reduction of the classification error of voxel v when the temperature is known at this particular voxel.

If (i) the visibility of all voxels in all viewpoints along the robot path is available in advance (ii) control signals are discrete and (iii) $\Delta\varepsilon(v)$ is known for all voxels; then the optimal control corresponds to the weighted maximum coverage problem with limited budget and motion constraints. Such formulation is an instance of the Mixed Integer Linear Program (MILP). However since an unknown environment is typically explored, neither the map nor $\Delta\varepsilon(v)$ is known.

It is possible to approximate the map from the depth measurements available so far and assume a uniform reduction in classification error. However, computing the visibility for tens of thousands voxels and solving the MILP with tens of thousands constraints is extremely demanding, which makes real-time replanning, whenever a new measurement is available, technically intractable. Consequently, we focus rather on the reactive control, where a policy directly maps available `RGBDT` measurements on the motion control u_{k+1}. In particular, we follow the recent success of Q-learning with deep CNNs and train a CNN which assigns Q-values representing expected classification error, relative to the one obtained under full visibility and optimal voxel coverage. The policy then chooses the control u^*_{k+1} which maximizes the Q-value.

2.1 Policy initialization

Since the raw sensory measurements are high-dimensional, learning of a deep Q-value network `CNN:RGBD`$\rightarrow$`Q` from randomly initialized weights would require a large number of training examples. To avoid such a demanding training procedure, we suggest to divide `CNN:RGBD`$\rightarrow$`Q` into

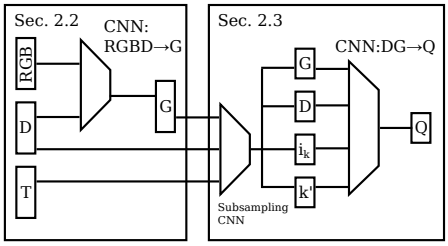

Figure 1: Deep CNN control policy overview. The policy is composed of two subnetworks, `CNN:RGBD→G` and `CNN:DG→Q`, with an interconnecting subsampling layer in the middle.

two sub-networks: (i) `CNN:RGBD→G` which predicts a gain `G` (i.e., the proposed approximation of $\Delta\varepsilon$) from `RGBD`-data and (ii) `CNN:DG→Q` which predicts the Q-values `Q` from the depth `D` and the gain `G`. These two networks are first trained independently and then concatenated into `CNN:RGBD→Q`. The overview of the Q-value network is shown in Fig. 1.

There are many ways how to approximate the decrease in classification error $\Delta\varepsilon$: for example when Gaussian Processes or decision trees are used instead of CNNs, the expected classification errors $\varepsilon, \varepsilon_{\mathtt{T}}$ can be obtained directly from the covariance of the posterior class probability. For CNN, we can approximate $\Delta\varepsilon$ as a signed difference of responses of the previously learned networks, $y \cdot (\mathtt{CNN:RGBDT{\to}V} - \mathtt{CNN:RGBD{\to}V})$, where $y \in \{-1, 1\}$ is the ground-truth label, namely, -1 for background, 1 for victim. Such an approximation, however, suffers from two drawbacks: (i) it requires manually annotated segmentation dataset of $\mathtt{RGBDT}_i$ with ground-truth labels y_i, (ii) `CNN:RGBD→V` already provides presumably the best estimate of y given the model class constraints, and we cannot thus hope to improve it further.

Consequently, we decided to simplify the approximation further and use the absolute difference of responses of the segmentation networks, $\mathtt{G} = |\mathtt{CNN:RGBDT{\to}V} - \mathtt{CNN:RGBD{\to}V}|$, which measures the influence of `T`-measurements on the segmentation output. This approximation is reasonable only when `T`-measurement is assumed to improve the classification in most of the cases. On the other hand, it allows creating gain-annotated data set of $(\mathtt{RGBD}_i, \mathtt{G}_i)$ pairs from arbitrary `RGBDT` data without manual annotation for learning `CNN:RGBD→G`, as described in the next section.

2.2 Multimodal image-based segmentation

Three different image-based models were learned, depending on the sensory modalities used and given task: `CNN:RGBD→V`, `CNN:RGBDT→V`, and `CNN:RGBD→G`. For `RGB`, all the models reuse the 16-layer VGG net as adapted and fine-tuned by [3], namely the *FCN-32s* variant, and combine it with additional models for depth and thermal modalities in a late-fusion scheme. Since annotated depth and thermal data are much scarcer, and no suitable trained models are available for these modalities, we employ smaller models with four convolution layers.

We compose the multimodal models by summing up the outputs of the (last) deconvolution layers, directly before the final softmax layer. All models were learned using a momentum-accelerated stochastic gradient descent on a semi-synthetic image data set of humans in various poses with known ground-truth segmentation. The segmentation models used the multinomial logistic loss for training, while the regression model used the Euclidean loss. The `CNN:RGBD→G` model was fine-tuned from the `CNN:RGBD→V` model after removing the softmax layer.

The pixel-wise predictions are accumulated into the corresponding voxels using the depth measurements `D`, taking occlusions into account.

2.3 Guided Q-learning

The second sub-network `CNN:DG→Q` is trained by the proposed guided Q-learning method. The algorithm successively collects training transitions from available maps and trains Q-value regression network $Q_{\mathbf{w}}(\mathbf{x}, u)$ with weights $\mathbf{w}$, which assigns expected loss in the long-term sum of gains due to control actions u taken in states $\mathbf{x}$. The state vector $\mathbf{x}_k$ consists of (i) down-sampled half-panoramic depth and gain data, `D` and `G`, respectively, observed at the current position, (ii) current viewpoint i_k, and (iii) the remaining length of the robot's path $k' = K - k$. The gain of the pixels for which the thermal measurement is available is set to zero; otherwise, it is provided by the `CNN:RGBD→G` network.

The guided Q-learning first estimates the optimal control of the thermal camera $u_1^*, \ldots, u_K^*$ and the corresponding optimal sum of gains c^* by solving the MILP task. Then it evaluates the sum of gains c achievable for all possible control actions $u \in \mathcal{U}$ by successively fixing $u_1^* = u$ and solving the corresponding MILP instances from the following state. For each possible $u \in \mathcal{U}$ we compute the relative long-term sum of gains as $Q = c/c^*$, and store all such $(\mathbf{x}_k, u, Q)$ tuples. Eventually, either the optimal control u_k^* or Q-value-driven control $\arg\max_{u \in \mathcal{U}} Q(\mathbf{x}_k, u)$ is applied, and the process continues from the following state. If a sufficient number of transitions is collected, we perform the gradient descent on weights $\mathbf{w}$ of the regression network $Q_{\mathbf{w}}(\mathbf{x}, u)$ until the testing error stops decreasing.

In contrast to the standard Q-learning, the guided Q-value network is not forced to predict the absolute sum of gains, which is often loosely connected with the features observed in the current state. Guided Q-learning rather predicts the expected impact on the optimality. Another advantage stems from guiding the exploration of the state-action space close to the optimal trajectories. In all experiments, the guiding probability linearly decreases from 1 to 0.

3 Experimental results

First, we evaluated the guided Q-value policy (`GQ-policy`) from Sec. 2.3 on 64 randomly generated maze-like maps in terms of the gain accumulated along the path. It was compared to a (i) reactive control similar to [5], here denoted by `greedy`, which at each position chooses the viewpoint maximizing the current gain, (ii) Q-value policy (`Q-policy`) trained by Q-learning similar to similar to [4] and (iii) the `optimal` control from solving MILP. The gain achieved by the `optimal` control creates a theoretical upper bound and is used to normalize the results. The proposed `GQ-policy` achieved 0.85 of the optimum on average and thus outperformed the `greedy` policy, which reached to 0.66, and `Q-policy`, which reached to 0.72, by a large margin. We conclude that the proposed `GQ-policy` policy outperformed the `greedy` one by a factor of 1.28 and `Q-policy` one by a factor of 1.18.

Second, we evaluated `GQ-policy` on 14 indoor data sets from the mobile SAR robot, and compared it to the `greedy` policy, which served as the baseline in these experiments because the theoretical upper bound was not known. Here, the proposed `GQ-policy` outperformed the `greedy` one by a factor of 1.09 or 1.18, respectively, depending on whether the `CNN:DG→Q` subnet used the subsampled depth `D` as a separate feature or not.

While the performance of the proposed method in terms of the accumulated gain shows its suitability for real-time reactive control, a thorough evaluation regarding the classification performance itself is needed and remains as future work. Our current implementation, however, does not achieve real-time performance due to limited computational power on board; the CNN prediction, which is the most expensive part, takes 0.27 s on GeForce GTX TITAN X (off-board). The real experiments were conducted in a stop-and-go manner, which allowed to create a fair testbed for all methods.

Acknowledgment

The research leading to these results has received funding from the European Union under grant agreement FP7-ICT-609763 TRADR; from the Czech Science Foundation under Project GA14-13876S, and by the Grant Agency of the CTU in Prague under Project SGS15/081/OHK3/1T/13.

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105. Curran Associates, Inc., 2012.

[2] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-End Training of Deep Visuomotor Policies. *JMLR*, 17(1):1334–1373, 2016.

[3] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*, pages 3431–3440, 2015.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 02 2015.

[5] K. Shubina and J. K. Tsotsos. Visual search for an object in a 3D environment using a mobile robot. *CVIU*, 114(5):535–547, 2010.