Adaptive 3-D Object Classification with Reinforcement Learning

Jens Garstka and Gabriele Peters

*Human-Computer Interaction, Faculty of Mathematics and Computer Science, University of Hagen,
D-58084, Hagen, Germany*

Keywords: 3-D Object Classification, Reinforcement Learning.

Abstract: We propose an adaptive approach to 3-D object classification. In this approach appropriate 3-D feature descriptor algorithms for 3-D point clouds are selected via reinforcement learning depending on properties of the objects to be classified. This approach is supposed to be able to learn strategies for an advantageous selection of 3-D point cloud descriptor algorithms in an autonomous and adaptive way, and thus is supposed to yield higher object classification rates in unfamiliar environments than any of the single algorithms alone. In addition, we expect our approach to be able to adapt to subsequently added 3-D feature descriptor algorithms as well as to autonomously learn new object categories when encountered in the environment without further user assistance. We describe the 3-D object classification pipeline based on local 3-D features and its integration into the reinforcement learning environment.

1 MOTIVATION

In the field of human-robot interaction it is essential for an interactive system to recognize and classify objects in its environment. A number of 3-D object classification methods exist, which are successful in defined contexts such as scene understanding, navigation, or applications in robotics like grasping or manipulation. But none of them is flexible enough to satisfy the needs for arbitrary environments or in several different applications or contexts. At the same time in the field of machine learning approaches have been developed that are able to adapt to dynamic changes in environments and to learn behavioral strategies autonomously. One of these approaches is represented by methods of reinforcement learning. It seems reasonable to overcome some of the inflexibilities of state-of-the-art 3-D object classification methods by an appropriate fusion with an reinforcement learning approach.

There are different ways, how 3-D objects can be recognized and classified, e. g., by direct matching of global 3-D point cloud descriptions, pairwise matching of local descriptions with clustering, or dictionary based approaches, where local descriptions are summarized in histograms, before they are compared. Over the years local 3-D feature descriptor algorithms have emerged as the most promising means to compare 3-D point clouds.

In the present work we will focus only on local 3-D feature descriptor algorithms for point clouds without additional structural information like triangle meshes and surfaces. These local 3-D feature descriptor algorithms differ considerably with respect to recognition rates and computation times. However, in general the computational costs of calculation and comparison for local feature descriptors are high. A comparison of recent algorithms (Alexandre, 2012) and a survey of local feature based approaches for 3-D object recognition (Guo et al., 2014) show, that there is not a single best algorithm in the domain of 3-D object recognition and classification for all purposes. This raises the question which descriptors should be used in which cases. Furthermore, the question arises whether a combination of two or more different algorithms leads to better results.

This proposal provides a concept which may offer answers to these questions. We present a method, where we use reinforcement learning to learn sequences of 3-D point cloud descriptors to obtain high classification rates. Section 2 gives a short overview over the required components and section 3 describes our approach in detail.

Garstka J. and Peters G..
Adaptive 3-D Object Classification with Reinforcement Learning.
DOI: 10.5220/0005563803810385
In *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2015), pages 381-385
ISBN: 978-989-758-123-6

2 RELATED WORK

2.1 Reinforcement Learning

In general, reinforcement learning (Sutton and Barto, 1998) describes a class of machine learning algorithms, in which an agent tries to achieve a goal by trial and error. The agent acts in an environment and learns to choose optimal actions in each state of the environment. The strategy of chosen actions is called policy. Further, it is assumed that the goals of the agent can be defined by a reward function that assigns a numerical value to each distinct action the agent may take in each distinct state of the environment. In this environment the task of the agent is to choose and apply one of the available actions in the current state. This changes the environment which leads the agent to the next state, and the agent can observe the consequences (the immediate reward). While repeating these steps the reinforcement learning agent can learn a policy. Typically, it is desired to find a policy that maximizes the accumulated reward.

Watkins (Watkins, 1989) introduced a reinforcement learning algorithm called Q-learning. In his method the agent exists within a world that can be modeled as a Markov Decision Process, consisting of a finite number of states and actions. In each step the agent performs one of the available actions, observes the new state, and receives the reward. This reward and the expected future reward result in the so-called quality-value (q-value). With ongoing iterations all q-values for each possible state-action pair will be approximated. Watkins and Dayan (Watkins and Dayan, 1992) proved that the discrete case of Q-learning will converge to an optimal policy under certain conditions. A series of survey articles of reinforcement learning methods is collected in (Wiering and Van Otterlo, 2012).

2.2 3-D Keypoint Detectors

3-D keypoint detectors are essential for local 3-D feature based approaches. They reduce the computational complexity by identifying those regions of 3-D point clouds, which are interesting for descriptors in terms of high informational density. There has been a lot of research in this field in the last few years. A good overview of keypoint detector performances is provided in the comparative evaluations of Salti et al. (Salti et al., 2011) and Filipe and Alexandre (Filipe and Alexandre, 2013). Additionally, there is an overview of the available 3-D keypoint detection algorithms in (Guo et al., 2014). For our approach, we will choose the intrinsic shape signatures (Zhong, 2009) based on the results of Salti et al. and Felipe and Alexandre, where it yielded best results in terms of repeatability.

2.3 Local 3-D Feature Descriptors

The number of published local 3-D feature descriptor algorithms has grown considerably in the last few year. But there are two mentionable early approaches, the splash (Stein and Medioni, 1992) and the widely used spin image (Johnson and Hebert, 1998). These two methods stand for numerous subsequently developed algorithms that can be grouped into two categories: signature and histogram based methods. Alexandre and Gou et al. (Alexandre, 2012; Guo et al., 2014) provide extensive comparisons of state of the art local 3-D feature descriptor algorithms. The methods that we will use for an initial implementation will be the spin images, the 3-D shape context (3DSC) (Frome et al., 2004), the persistent feature histogram (PFH) (Rusu et al., 2008), the fast point feature histogram (FPFH) (Rusu et al., 2009), the unique signatures of histograms (USC) (Tombari et al., 2010b), and the unique shape context (SHOT) (Tombari et al., 2010a).

2.4 Classification

As already mentioned, there are different ways to match local 3-D features of an unknown object under inspection against previously experienced features. For our approach, it is assumed that the existing point clouds have already been segmented. Thus, the local feature descriptions can be used directly for classification without a previous clustering (hypothesis generation). In this case bag-of-words models with quantized local descriptors as used in recent classification pipelines (Toldo et al., 2010; Wu and Lin, 2011; Cholewa and Sporysz, 2014) can easily be used. In these pipelines a directory of keywords (quantized feature descriptions) is determined by clustering. By filling the keywords found with a nearest neighbor search for all local 3-D feature descriptions into a histogram, the number of each occurrence is counted. Finally, these histograms for an object can be learned and/or classified by support vector machines (SVM) or random trees (RT).

3 ADAPTIVE 3-D OBJECT CLASSIFICATION

As indicated in the introduction, the main goal of our approach is the autonomous learning of an opti-

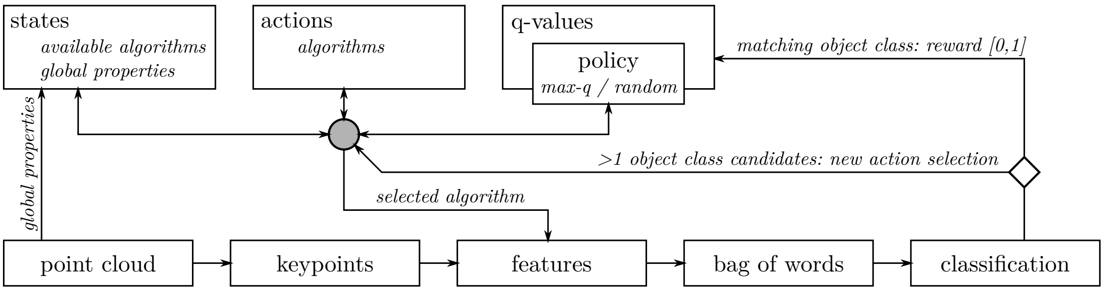

Figure 1: The proposed classification system. The basic classification pipeline is shown in the lower half. The reinforcement learning part used for the selection of a local 3-D feature descriptor algorithm (action) is shown above. During the classification phase the decision which algorithm should be selected first/next (represented by the circle), will be made based on the current state (the currently available actions and global properties) and the learned q-values (*max-q*). During the initial learning phase the algorithms will be chosen randomly.

mized combination and application of various local 3-D feature descriptor algorithms with the purpose to increase the overall classification rate of 3-D point clouds. Moreover, depending on some basic properties of a point cloud, the combinations of these algorithms are supposed to vary. The steps, how we intend to realize this, are described in the following sections.

3.1 The Basic Classification Pipeline

The basic classification pipeline consists of five main steps. Given a 3-D point cloud we start with the collection of global properties. These properties are the total number of points, the point cloud resolution, the eigenvalues to get the variance that is correlated with each eigenvector and the dimensions of the point cloud along the eigenvectors. These values will be used by the reinforcement learning agent to select the first algorithm. In the second step, the intrinsic shape signature keypoint detection algorithm (Zhong, 2009) will be used to determine points of interest. During the third step, one of the local 3-D feature descriptor algorithms stated in section 2.3 will be applied. As a result we will get a set of local 3-D feature descriptions. Each of the determined feature descriptions is quantized to be binned in a histogram (step four). In the last step the values of the histogram will be used as input vectors for a classifier, i. e., for an SVM, to identify an appropriate object class. This pipeline is depicted in the lower half of figure 1.

3.2 Fusion with Reinforcement Learning

The basic classification pipeline will be enhanced by a reinforcement learning agent.

States consist of sets of local 3-D feature descriptor algorithms that have not already been used (a descriptor will be applied only once) and the global properties mentioned above. In order to achieve a finite number of possible states, the continuous global properties will be divided into a fixed number of classes.

Actions correspond to the selection of a local 3-D feature descriptor algorithm and the application of the last three steps of the basic pipeline (see figure 1), i. e., the computation of local 3-D feature descriptions based on the selected algorithm and of the histogram which is fed as input vector into the SVMs of remaining object classes. Usually the object class with the highest score would be used as a single result. In our case, using the SVMs as binary classifiers trained with the responses -1 and 1, we will reject all classes with a corresponding output < 0 and keep all classes ≥ 0 as object class candidates.

Without any restrictions the reinforcement learning agent would stop, if there is no remaining object class or 3-D feature descriptor algorithm left. But this “natural” termination is not desirable, since we propose a time limit $t_{\max}$ how long a single object classification should take. Thus, the learning process breaks down into episodic tasks that end in four possible final states. These states are reached if all algorithms have been used once, if no object class is left, if exact one object class is left, or if the accumulated computation time exceeds the predefined time limit $t_{\max}$. In the only remaining situation, when there is more than one object class candidate left, the reinforcement learning agent continuous with the next action, i. e., with the selection of the next algorithm (see figure 1).

At this point we have to clarify among which conditions the reinforcement learning agent gets an immediate reward. The reward for all states except for the final states is 0. When the accumulated computation time is reached or no algorithm/object class is left, the reward is also 0. For the remaining case, when exact one object class is left, the reward depends on the phase of the reinforcement learning system.

3.2.1 Initial Learning Phase

The application of a reinforcement learning method is always coupled with the question, how much exploration and exploitation should be granted to the reinforcement learning agent. Typically, the reinforcement learning system starts with a random policy for maximum exploration. In this phase, we will use already known and classified objects from freely available data sets, e. g. from the large-scale hierarchical multi-view RGB-D object dataset of the University of Washington (Lai et al., 2011), since the decision whether the final category matches is straightforward. If the determined object class matches the input object class, the reward is a value $\in [0,1]$ depending on the accumulated computation time left with respect to $t_{\max}$. Otherwise the reward is 0, too.

3.2.2 Classification Phase and Adaptive Learning

If the q-values get more stable, the exploration is commonly reduced in favor of exploitation. This method is called ε-greedy, meaning that most of the time those actions are selected, where the expected reward is maximized, but with the probability of ε a random action is selected. However, instead of selecting single actions randomly, we will adopt this concept to completely random policy episodes with the following behavior: during the regular classification of (unknown) objects we will use a max-q policy selecting always that action with the highest proposed q-value. In these episodes no modifications of the q-values will be made – and thus a reward is irrelevant. Occasionally we will integrate random policy episodes with known objects. In this way, the system adapts to changes of the environment over time and allows us to add new descriptors to the system.

3.2.3 Handling New Categories

A big advantage of our approach consist in the fact that the number of object classes the system can recognize can increase dynamically. If no object class candidate is left during the classification phase, an explicit comparison with multiple accurate descriptors such as PFH and SHOT is envisaged. In case, the object can still not be assigned to one of the object classes at hand, a new unlabeled class is created, which means that the reinforcement learning agent has learned a new object class autonomously. New classes, of course, should be labeled and from time to time reviewed for consistency.

3.2.4 Evaluation

To evaluate the results, we will determine the recognition rates for each local 3-D feature descriptor algorithm individually, using the basic classification pipeline proposed in section 3.1. Subsequently, the individual results can be compared with the results of our adaptive 3-D object classification approach.

4 CONCLUSIONS

This proposal suggests a system which learns a strategy to select and apply 3-D point cloud descriptor algorithms with the goal to classify a point cloud with high accuracy within a preset time limit. The proposed approach is based on a reinforcement learning system with a 3-D classification pipeline in selecting local 3-D feature descriptor algorithms. Due to properties of reinforcement learning we expect the approach to be highly adaptive, e. g., allowing the integration of new descriptors and the on-line learning of new object categories.

REFERENCES

Alexandre, L. A. (2012). 3d descriptors for object and category recognition: a comparative evaluation. In *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal*.

Cholewa, M. and Sporysz, P. (2014). Classification of dynamic sequences of 3d point clouds. In *Artificial Intelligence and Soft Computing*, pages 672–683. Springer.

Filipe, S. and Alexandre, L. A. (2013). A comparative evaluation of 3d keypoint detectors. In *9th Conference on Telecommunications, Conftele 2013*, pages 145–148, Castelo Branco, Portugal.

Frome, A., Huber, D., Kolluri, R., Bulow, T., and Malik, J. (2004). Recognizing objects in range data using regional point descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., and Wan, J. (2014). 3d object recognition in cluttered scenes with local surface features: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11).

Johnson, A. E. and Hebert, M. (1998). Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, 16(9):635–651.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011). A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE.

Rusu, R., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *Robotics* *and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3212–3217.

Rusu, R. B., Blodow, N., Marton, Z. C., and Beetz, M. (2008). Aligning point cloud views using persistent feature histograms. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3384–3391. IEEE.

Salti, S., Tombari, F., and Stefano, L. D. (2011). A performance evaluation of 3d keypoint detectors. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pages 236–243. IEEE.

Stein, F. and Medioni, G. (1992). Structural indexing: efficient 3-d object recognition. *IEEE Trans. PAM*, 14:125–145.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press.

Toldo, R., Castellani, U., and Fusiello, A. (2010). The bag of words approach for retrieval and categorization of 3d objects. *The Visual Computer*, 26(10):1257–1268.

Tombari, F., Salti, S., and Di Stefano, L. (2010a). Unique shape context for 3d data description. In *Proceedings of the ACM workshop on 3D object retrieval*, pages 57–62. ACM.

Tombari, F., Salti, S., and Di Stefano, L. (2010b). Unique signatures of histograms for local surface description. In *Computer Vision-ECCV 2010*, pages 356–369. Springer.

Watkins, C. and Dayan, P. (1992). Technical note: Q-learning. *Machine Learning*, 8(3-4):279–292.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK.

Wiering, M. and Van Otterlo, M. (2012). Reinforcement learning. In *Adaptation, Learning, and Optimization*, volume 12. Springer.

Wu, C.-C. and Lin, S.-F. (2011). Efficient model detection in point cloud data based on bag of words classification. *Journal of Computational Information Systems*, 7(12):4170–4177.

Zhong, Y. (2009). Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696. IEEE.