FernUniversität in Hagen

Department of Mathematics
and Computer Science

Learning Strategies to Select Point Cloud Descriptors for 3D Object Classification: A Proposal

Jens Garstka and Gabriele Peters

2014

Research Report 1/2014
ISSN 1865-3944

Editor:	Dean of the Department of Mathematics and Computer Science
Type and Print:	FernUniversität in Hagen
Distribution:	http://deposit.fernuni-hagen.de/view/departments/miresearchreports.html

Learning Strategies to Select Point Cloud Descriptors for 3D Object Classification: A Proposal

Jens Garstka and Gabriele Peters

February 20, 2014

Abstract

We propose a reinforcement learning approach for an adaptive selection and application of 3D point cloud feature descriptors for the purpose of 3D object classification. The result of the learning process is an autonomously learned strategy of selection of descriptors with the property that the successive application of these descriptors to a 3D point cloud yields high classification rates among a large number of object classes. The order in which the descriptors are applied to an unfamiliar point cloud depends on the features calculated in previous steps of the descriptor sequence, i.e., the sequence of descriptors depends on the object to be classified, thus it is highly adaptive. Our approach starts with a given number of descriptors and object classes, but it is able to adapt dynamically to changes in the environment. For example, further descriptors can be added during the learning process, and new object classes are created autonomously if necessary.

1 Introduction

Due to a wide range of applications such as scene understanding, navigation or applications in robotics like grasping or scene manipulation, the classification and recognition of 3D point clouds has been a fundamental part of computer vision research in the last few years. Additionally, the appearance of cheap 3D cameras like the Microsoft Kinect made these fields of application available to a broader public.

Most of the current algorithms compare objects pairwise by matching the descriptions of whole objects or of local feature descriptions. Global descriptors like the frequency domain approaches from Saupe [SV01] and Vranić [VS02] are rather rare. Over the years local feature descriptions have emerged as the most promising means to compare 3D point clouds and surfaces. Mentionable approaches from the 1990s are "splashes" by Stein and Medioni [SM92] and "spin images" from Johnson and Herbert [JH98]. There are many other state of the art 3D point cloud feature descriptors with different recognition rates and time complexities, from which a selection is introduced in section 2. However, in general the computational costs of calculation and comparison for local feature descriptors are high. Furthermore, always more than only a few local feature vectors are necessary to accurately match corresponding point clouds. To reduce the computational costs of potentially useless and insignificant feature vectors, keypoint detectors are used to select regions of interest in the point cloud.

As with many other problems, there is not one best solution in the domain of 3D object recognition and classification [Ale12]. Especially when the feature vectors have to be compared with a large set of other feature vectors, for example if the classification structure contains many objects or object classes, the application of only a single feature descriptor, whether it is accurate but slow or fast but inaccurate, is questionable. This raises the question which descriptors should be used and in which order they should be applied.

This proposal provides a concept which may offer an answer to this question. We present a method, where we use reinforcement learning to learn an order in which point cloud descriptors have to be applied to obtain high classification rates, hereafter referred to as "sequences of point cloud descriptors". Section 2 gives a short overview over the required components and section 3 describes our approach in detail.

2 Related Work

In this section we give a short overview over existing 3D point cloud descriptors. We first present a selection of different global point cloud descriptors, which can be used by the reinforcement learning agent to perform the first comparisons in the sequence of descriptors with low computational costs. Afterwards we give a draft overview of state of the art local 3D feature descriptors. Finally we give a short introduction to reinforcement learning.

2.1 Global Point Cloud Descriptors

One straightforward way to describe a 3D point cloud, is a bounding box aligned along the principle axes. In this way, we will be able to describe a point cloud with just two length ratios in a very simple manner. Suzuki et al.[SKO00] use the PCA in a similar manner to get a stable orientation. They fit the point cloud into a unit cube, divide the cube into a coarse grid and count the points in each grid cell. Vranić and Saupe [VS01] and Lucchese et al.[LDC02] divide the 3D point cloud into voxel grids and use it as input for a 3D Fourier transform. While Vranić and Saupe use the absolute values of the obtained coefficients as feature vector, Lucchese et al. use the slice theorem to calculate radial projections, which they compare. There are several other approaches [Kei99, PR99, PRM$^+$00] which compare the similarity of voxel grids.

A different approach from Heczko et al.[HKSV02] consists in the creation of a parallel projection onto the 6 faces of a bounding cube aligned by PCA and the application of a Fourier transform on the so obtained silhouettes. The absolute Fourier coefficients are used for the feature vectors. In a second approach Heczko et al. create depth images on the 6 faces of the bounding cube, apply the Fourier transform and use the absolute values of the Fourier coefficients as feature vectors. Saupe [SV01] and Vranić [VS02] use a spherical projection of the inner centred point cloud. They calculate a depth image with the surface distance to the surrounding sphere and an image with the values of the scalar product of the surface normals and the projection rays. With the spherical harmonics representation of both 2D representations in a complex function, the Fourier transform is reversible.

In addition, there are many other approaches. However, there is one thing that most of the mentioned global methods have in common: they need the complete model for a successful classification, which is a problem particularly with regard to depth images taken from a single point of view, e. g., from the Microsoft Kinect.

2.2 Local Feature Point Descriptors

While in recent years only a few new global descriptors have been published, the number of published local feature point descriptors has grown considerably. Two early approaches were already mentioned in the introduction. The "splash" by Stein and Medioni [SM92] is a surface description based on surface normals along a geodesic circle. The widely used "spin image" by Johnson and Herbert [JH98] is created for an oriented point while counting the surrounding points of the point cloud in the bins of a 2D histogram. Regarding to the oriented point's normal vector, the bins are selected by the horizontal and vertical distance to the compared point. In [Ale12] Alexandre provides a wide comparison of state of the art local feature point descriptors. The methods in this paper are restricted to those provided within the point cloud library (PCL) version 1.6 [RC11], but work on pure 3D point cloud data. Without the use of the color information, PFH [RBMB08] and SHOT [TSDS10] perform best. In addition, Heider et al.[HPPLG11] compare a large number of different local shape descriptors. The conclude, that the distribution descriptor is consistently the best.

2.3 3D Keypoint Detectors

3D keypoint detectors are essential for local feature point descriptors. They reduce the computational complexity by identifying particularly those regions of 3D 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 the differences and the keypoint detector's performance is provided in the comparative evaluations of Salti et al.[STS11] and Filipe and Alexandre [FA13]. In our case scale invariant detectors are of particular interest, since we do not make any assumptions about the point cloud source. While Salti et al. prefer MeshDoG [ZBVH09], Filipe and Alexandre prefer SIFT3D [FDH07] as scale invariant detector. Since the latter algorithm is present in the PCL, we'll use this method for a first implementation.

2.4 Reinforcement Learning

In general, reinforcement learning (RL) [SB98, Kapitel 6] 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 this steps the RL agent can learn a policy. Typically, it is desired to find a policy that maximizes the accumulated reward.

Watkins [Wat89] introduced a RL algorithm called Q-learning. In his method the agent exists within a world that can be modelled as a Markov Decision Process, consisting of a finite number of states and actions, as well as transition probabilities, which reflect the probabilities that chosen actions result in particular states. In each step the agent selects one of the available actions, observes the new state and receives the immediate 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 [WD92] proved that the discrete case of Q-learning will converge to an optimal policy under certain conditions. These conditions are, that the learning rate $\alpha \in [0, 1]$ (the update ratio of q-values) should decrease over time, that each state should be visited an infinite number of times and that each action in those states has do be used an infinite number of times.

3 Our approach

As already mentioned, the computational costs of calculations and comparisons of local feature descriptors are high. On the other hand, global descriptors are too imprecise for an accurate assignment of a 3D point cloud to an object class in most cases. In particular with a rising number of different object classes, the number of possible result classes should be restricted first of all by efficient methods, so that the computational costs of subsequent calculations of local feature descriptors remain within tolerable limits.

One naive approach to reach this goal would be, to evaluate a huge bunch of descriptor sequences with global and local 3D point cloud descriptors. Maybe this evaluation results in one or more well-functioning sequences. But even if this would yield high classification rates for many objects, this approach has to be rejected on principle because of its lack of generalizability and its lack of adaptivity, e.g., in the case of adding new descriptors.

We think that our approach, using a RL framework to learn an optimal application order of descriptors online, is a way to solve this questions.

3.1 Environment

Our RL framework consists of a finite Markov Decision Process with a finite number of actions and a finite number of states, as described in section 2.4. Furthermore it includes a structure, with a small set of already classified 3D point clouds and preprocessed feature vectors for all implemented point cloud descriptors, as shown in figure 1.

A *state* consist on the one hand of the number of remaining candidates from the object-classes. In addition, it contains the set of remaining actions, since one descriptor will be applied only once. Finally it is envisaged, that descriptors should be able to contribute their results to the state. It is not intended, that this option is used for all descriptors, but it can be used where it makes sense.

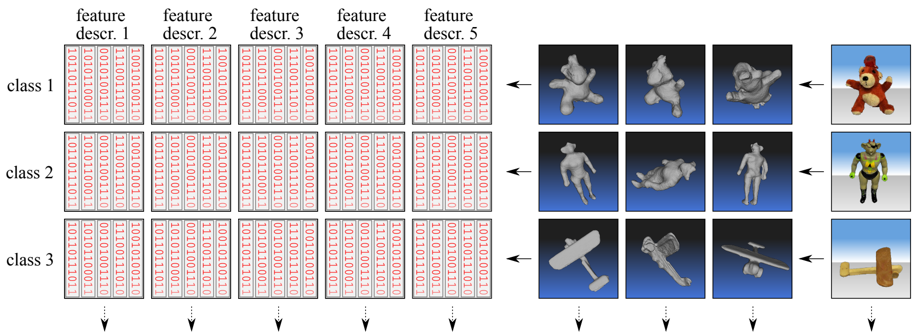

Figure 1: Building an initial structure of already classified 3D point clouds and preprocessed feature vectors for all implemented point cloud descriptors.

An example might be the transfer of basic shape attributes (e. g., flat, longish or uniformly), which can be obtained by PCA, to the state. The intention to embed the elementary results of different, probably predominant global descriptors, is to enable the RL agent to use different actions due to different descriptor results which correspond to different states, respectively.

An *action* corresponds to the application of a 3D point cloud descriptor. More precisely, it consists of three steps: 1) the selection of the descriptor, 2) the calculation of the feature vector and 3) the elimination of candidates from the object-classes by comparing the current feature vector with all feature vectors learned so far in the previous steps. This process is shown in figure 2.

The *transition probabilities* are unknown in advance. However, they arise from the actions, respectively, from the following three properties: firstly the number of the remaining candidates from object-classes, secondly the history of previously applied descriptors and thirdly the results of the already applied descriptors, if available.

3.2 Learning Process

Without any restrictions the reinforcement learner terminates naturally, if the number of remaining object-classes is zero or all 3D point cloud descriptors have been used. But the natural termination is not desirable, since we propose a time limit how long a single object classification should take. Without this limitation the RL framework would probably learn to use a descriptor with high accuracy like PFH [RBMB08] or SHOT [TSDS10], but it would take a very long computational time to compare the feature vectors with all objects in the classification structure. Moreover, it makes no sense to wait until the set of object-class candidates is empty. Thus, the learning process terminates at the latest when only one class remains.

At this point we have to clarify among which conditions the RL agent gets an immediate reward. Determining the quality of the states during the learning and classification process, is the goal of our approach. Thus, there will be no rewards for the achievement of any state. The only states at which an evaluation of the result is possible, are the terminal states. When a terminal state is reached where only one class remains and the class does match, the reward is 1. Otherwise, if there are more classes left or the class does not match, the reward

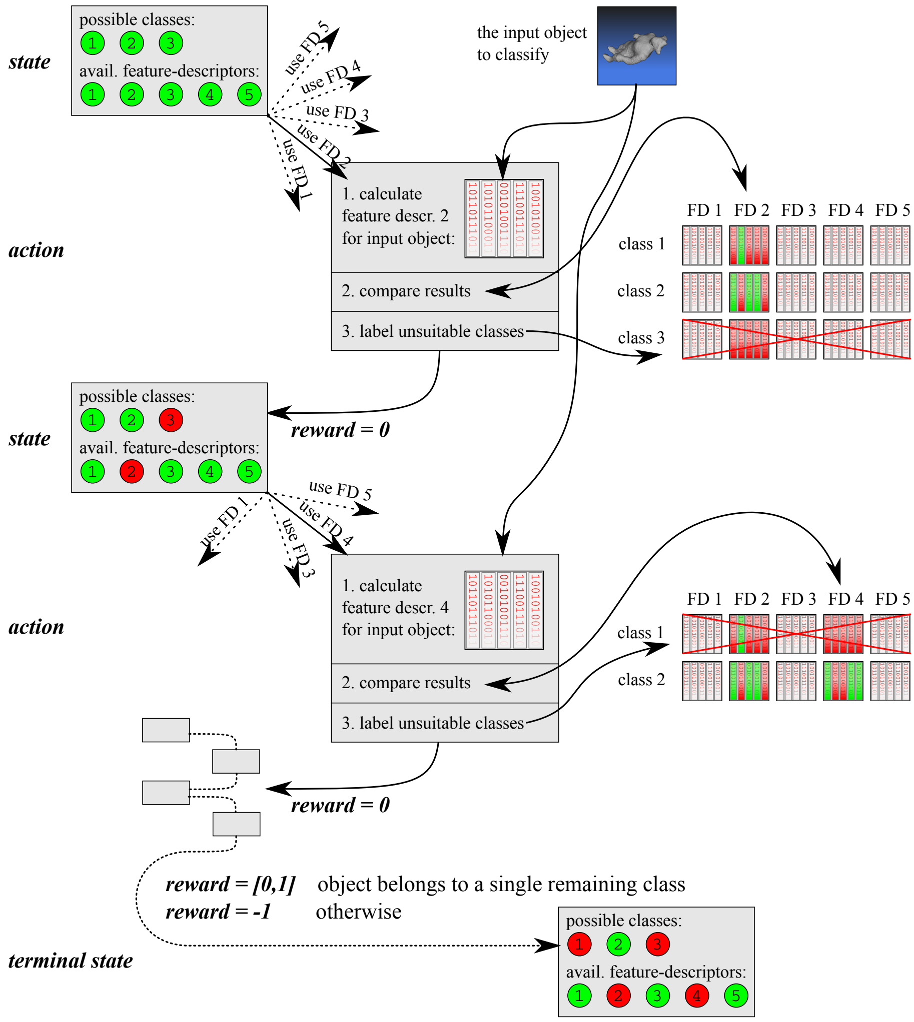

Figure 2: The basic process. From the current state the RL chooses one of the available descriptor algorithms. With this algorithm the feature vector(s) are calculated for an input object. By comparing this feature vector(s) to all classified feature vectors within the object-classes, an object class can be marked as unsuitable, if the quality and quantity of matching feature vectors is insufficient. This process will repeat until one or no suitable classes remain.

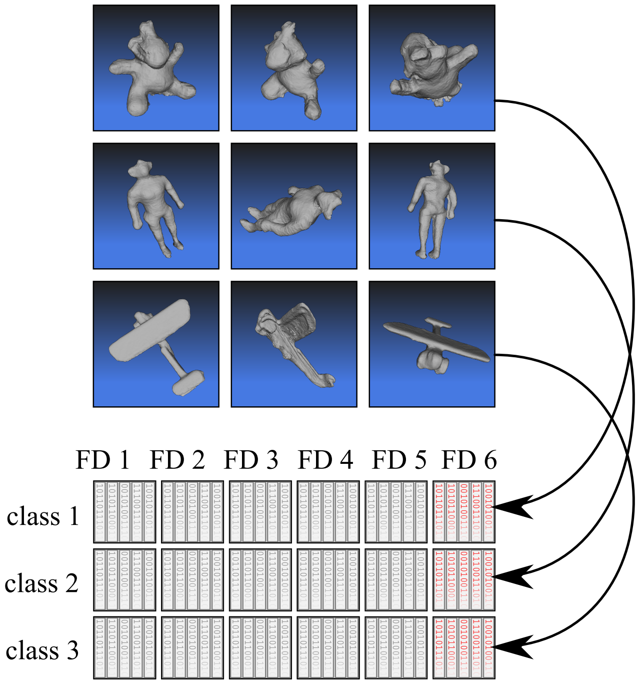

Figure 3: Adding new feature descriptors.

is -1. This so-called delayed reward and the trial and error learning are the most characteristic features of RL.

The application of a RL method is always coupled with the question, how much exploration and exploitation should be granted to the RL agent. Typically, the RL starts with a random policy for maximum exploration. In this phase, we start with the already known and classified objects from the classification structure, since the decision whether the final class fits or not is straightforward. If the q-values get more stable, the exploration is 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 probability ϵ a random action is selected. In this way it is possible that the system adapts to changes of the environment over time. This balance of exploration and exploitation allows the system to grow online and allows us to add new descriptors to the system (see figure 3). The new descriptors will be selected from time to time, so that the q-values will be adapted over time.

3.3 Handling New Categories

A big advantage of our approach compared to classical object classification approaches (e. g., neural networks based ones) consist in the fact that the number of object classes can change dynamically. Thus, the decision regarding a valid classification of a 3D point cloud must be independent from the explicit knowledge of an object-class, as used in the early exploration phase mentioned in the section above. For this purpose, an explicit comparison with accurate descrip-

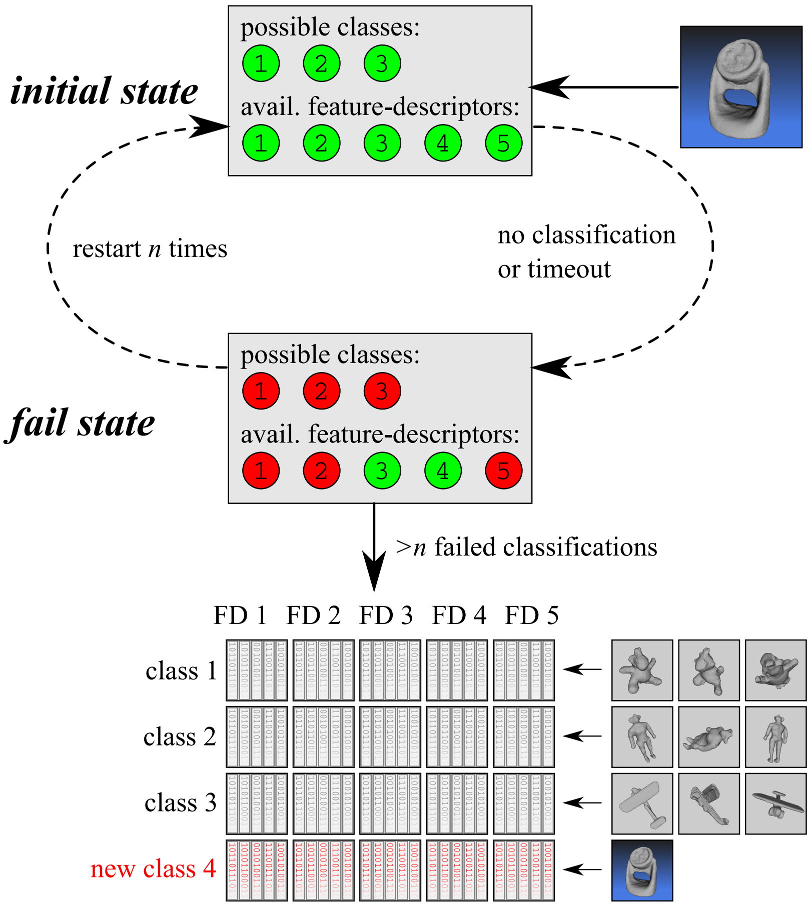

Figure 4: Automatically learned object classes. If the classification of an unclassified object fails multiple times while increasing random selections of descriptor algorithms (increasing ϵ), the RL will generate a new unlabeled object class.

tors such as PFH and SHOT is envisaged. To compensate for possible faulty decisions of the RL agent, the classification of an unclassified object will be repeated multiple times. In case, the object cannot be assigned to one of the object classes at hand, a new class is created, which means that the agent has learned a new object class autonomously, as shown in figure 4.

4 Conclusion

This proposal suggests a system which learns a strategy to select and apply 3D point cloud descriptors with the goal to classify a point cloud with high accuracy, namely among a large number of object classes and within a preset time limit. The proposed approach is based on reinforcement learning. The initial learning stage will be based on a 3D keypoint detector and a number of 3D point cloud descriptors. Due to properties of reinforcement learning we expect the approach to be highly adaptive, e.g., allowing the integration of new descriptors and the online learning of new object classes.

References

[Ale12] Alexandre, Luis A: 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*, 2012

[FA13] Filipe, Silvio ; Alexandre, Luís A.: A Comparative Evaluation of 3D Keypoint Detectors. In: *9th Conference on Telecommunications, Conftele 2013*. Castelo Branco, Portugal, May 2013, S. 145–148

[FDH07] Flint, Alex ; Dick, Anthony ; Hengel, Anton van d.: Thrift: Local 3d structure recognition. In: *Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on* IEEE, 2007, S. 182–188

[HKSV02] Heczko, M. ; Keim, D. ; Saupe, D. ; Vranic, D.: Methods for similarity search on 3D databases. In: *Datenbank-Spektrum (in German)* 2 (2002), Nr. 2, S. 54–63

[HPPLG11] Heider, Paul ; Pierre-Pierre, Alain ; Li, Ruosi ; Grimm, Cindy: Local shape descriptors, a survey and evaluation. In: *Proceedings of the 4th Eurographics conference on 3D Object Retrieval* Eurographics Association, 2011, S. 49–56

[JH98] Johnson, Andrew E. ; Hebert, Martial: Surface matching for object recognition in complex three-dimensional scenes. In: *Image and Vision Computing* 16 (1998), Nr. 9, S. 635–651

[Kei99] Keim, Daniel A.: *Efficient geometry-based similarity search of 3D spatial databases.* ACM, 1999

[LDC02] Lucchese, Luca ; Doretto, Gianfranco ; Cortelazzo, Guido M.: A frequency domain technique for range data registration. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24 (2002), Nr. 11, S. 1468–1484

[PR99] Paquet, Eric ; Rioux, Marc: Nefertiti: a query by content system for three-dimensional model and image databases management. In: *Image and Vision Computing* 17 (1999), Nr. 2, S. 157–166

[PRM+00] Paquet, Eric ; Rioux, Marc ; Murching, Anil ; Naveen, Thumpudi ; Tabatabai, Ali: Description of shape information for 2-D and 3-D objects. In: *Signal Processing: Image Communication* 16 (2000), Nr. 1, S. 103–122

[RBMB08] Rusu, Radu B. ; Blodow, Nico ; Marton, Zoltan C. ; Beetz, Michael: Aligning point cloud views using persistent feature histograms. In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* IEEE, 2008, S. 3384–3391

[RC11] Rusu, Radu B. ; Cousins, Steve: 3d is here: Point cloud library (pcl). In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on* IEEE, 2011, S. 1–4

[SB98] Sutton, Richard S. ; Barto, Andrew G. ; Dietterich, Thomas (Hrsg.) ; Bishop, Christopher (Hrsg.) ; Heckerman, David (Hrsg.) ; Jordan, Michael (Hrsg.) ; Kearns, Michael (Hrsg.): *Reinforcement learning: An introduction.* Cambridge Univ Press, 1998

[SKO00] Suzuki, Motofumi T. ; Kato, Toshikazu ; Otsu, Nobuyuki: A similarity retrieval of 3D polygonal models using rotation invariant shape descriptors. In: *Systems, Man, and Cybernetics, 2000 IEEE International Conference on* Bd. 4 IEEE, 2000, S. 2946–2952

[SM92] Stein, Fridtjof ; Medioni, Gérard: Structural indexing: efficient 3-D object recognition. In: *IEEE Trans. PAM* 14 (1992), S. 125–145

[STS11] Salti, Samuele ; Tombari, Federico ; Stefano, Luigi D.: A performance evaluation of 3d keypoint detectors. In: *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on* IEEE, 2011, S. 236–243

[SV01] Saupe, Dietmar ; Vranić, Dejan V.: 3D model retrieval with spherical harmonics and moments. In: *Pattern Recognition.* Springer, 2001, S. 392–397

[TSDS10] Tombari, Federico ; Salti, Samuele ; Di Stefano, Luigi: Unique signatures of histograms for local surface description. In: *Computer Vision–ECCV 2010.* Springer, 2010, S. 356–369

[VS01] Vranić, Dejan V. ; Saupe, D: 3D shape descriptor based on 3D Fourier transform. In: *Proceedings of the EURASIP conference on digital signal processing for multimedia communications and services (ECMCS 2001), Budapest, Hungary*, 2001

[VS02] Vranic, Dejan V. ; Saupe, Dietmar: Description of 3D-shape using a complex function on the sphere. In: *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on* Bd. 1 IEEE, 2002, S. 177–180

[Wat89] Watkins, Christopher John Cornish H.: *Learning from Delayed Rewards.* Cambridge, UK, King's College, Diss., May 1989

[WD92] Watkins, ChristopherJ.C.H. ; Dayan, Peter: Technical Note: Q-learning. In: *Machine Learning* 8 (1992), Nr. 3-4, S. 279–292

[ZBVH09] Zaharescu, Andrei ; Boyer, Edmond ; Varanasi, Kiran ; Horaud, Radu: Surface feature detection and description with applications to mesh matching. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* IEEE, 2009, S. 373–380