Boosting with a Joint Feature Pool from Different Sensors

Dominik A. Klein[1], Dirk Schulz[2], and Simone Frintrop[1]

[1] Institute of Computer Science III,
Rheinische Friedrich-Wilhelms-Universität,
53117 Bonn, Germany
[2] Forschungsgesellschaft für Angewandte Naturwissenschaften e.V. (FGAN),
53343 Wachtberg, Germany

Abstract. This paper introduces a new way to apply boosting to a joint feature pool from different sensors, namely 3D range data and color vision. The combination of sensors strengthens the systems universality, since an object category could be partially consistent in shape, texture or both. Merging of different sensor data is performed by computing a spatial correlation on 2D layers. An AdaBoost classifier is learned by boosting features competitively in parallel from every sensor layer. Additionally, the system uses new corner-like features instead of rotated Haar-like features, in order to improve real-time classification capabilities. Object type dependent color information is integrated by applying a distance metric to hue values. The system was implemented on a mobile robot and trained to recognize four different object categories: people, cars, bicycle and power sockets. Experiments were conducted to compare system performances between different merged and single sensor based classifiers. We found that for all object categories the classification performance is considerably improved by the joint feature pool.

1 Introduction

Object classification in sensor data is an important task for many applications. Especially autonomous mobile robots that have to act in a complex world rely on knowledge about objects in their environment. Imagine for example an automatically controlled car driving in city traffic or a universal housekeeping robot cleaning up your room. Various machine learning and pattern recognition methods have been studied to meet these demands. One area of active research in the field of object classification are boosting techniques [1–7]. An exhaustive survey can be found in [8].

While most approaches for object classification use camera data [9, 5, 6, 10–13], several groups also have investigated the use of other sensor data such as laser range finders [14, 15] or infrared cameras [16]. A reason for choosing different sensors is that each sensor has different strengths and drawbacks and some sensors capture information that others are not able to provide. Laser scanners for example provide accurate depth information and infrared cameras enable the detection of people or animals at night.

In this paper, we introduce an approach to automatically exploit the advantages of different sensors. We provide a feature pool that consists of a collection of feature candidates from different sensor layers. In a training phase, the boosting algorithm Gentle AdaBoost automatically selects the most distinctive feature at a time to obtain an optimal classification performance. Thus, it depends on the object type which features from which sensor layers are selected. To further improve the results, we introduce new corner-like features and a new measure to extract color information based on hue-distance.

We show the classification performance in various experiments for four different object classes: cars, people, bicycles and power sockets. Depending on the object type, different layers are chosen with different priorities. In all cases, the classification profited considerably from the fusion of data from different sensors; the classification performance was considerably higher than the classification rate of each sensor on its own.

The combination of data from different sensors has also been investigated by other groups. A straightforward solution is to train a classifier on the data from each sensor independently and in a second step combine the results. For example, Zivkovic and Kröse integrated a leg detector trained on 2D laser range data and a part-based person detector trained on omnidirectional camera images this way [17]. Frintrop et al. trained classifiers to detect chairs and robots on the range and the remission data of a laser scanner [18]. Nüchter et al. applied the same approach to people detection [19]. Here, the authors suggest two ways to join the two cascades: serial or interleaved. Both versions represent a logical "and" operator. In our approach instead, the boosting algorithm decides automatically which features to choose for an object type. It is thus a general approach to optimizing the sensor fusion for a certain object category. The result is a single, more concise classification cascade that achieves a faster classification with a better classification performance.

This paper comprehends results of the master's thesis[3] of Klein [20] and some further enhancements.

2 Adaptive Boosting with Haar-like Features

The Adaptive Boosting algorithm, short AdaBoost, forms a strong classifier as a weighted sum of as many weak-classifiers as are needed to reach a given precision on the training data [2]. Therefore it iteratively picks the weak-classifier out of a huge amount of possible candidate classifiers that performs best on a weighted training set. Subsequently it reweights the training set according to the outcome of the chosen weak-classifier: a failure raises the weight of the example, a correct match lowers its weight.

There are different versions of AdaBoost that differ on how weights are updated and how classifier performance is measured. We use Gentle AdaBoost with squared error metric to decide which candidate classifier is considered next, because it has been shown to outperform standard Discrete AdaBoost in [4], and

[3] available at www.iai.uni-bonn.de/~kleind/

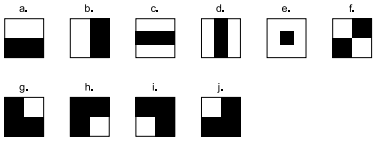

Fig. 1. Utilized Haar-like features. The sum of values in white regions is subtracted from those in black regions. a.-f.: standard features. g.-j.: new corner-like features.

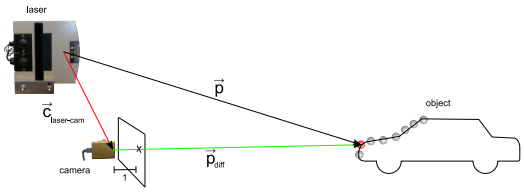

Fig. 2. Projection of a 3D point cloud. For every point $|\boldsymbol{p}_{diff}|$ is recorded at the intersection pixel between $\boldsymbol{p}_{diff}$ and the image plane of the camera.

we confirmed this result during our own experiments. In addition we arrange strong classifiers of increasing complexity in a cascade structure as proposed by Viola and Jones [5, 6] to speed up the system.

A common approach in computer vision to build weak classifiers is to use Haar-like features [9], which are inspired by Haar wavelet functions. In general, they consist of a positive and a negative area, whose values add to a common sum (cf. Fig. 1). For efficient computations, areas have upright rectangular borders, because it allows the use of integral images to compute this sum in a constant time. An integral image, also known as summed area table, is an intermediate step between per pixel values and sums of values in rectangular regions [5]. For every pixel position, the sum of all pixel values left and above this position is stored. This integral image can be computed in a single pass over the image by building the integral image in normal reading direction and just adding the current pixel value to sums computed before. With this integral image, the sum of any rectangular region can be computed with the four values at its corners.

Regions are combined to form simple templates that match to edge, line or center-surround features. To further enlarge the over-complete candidate pool and approximate diagonal features, we introduce new, corner-like features (cf. Fig. 1, g-j). In contrast to the diagonal features in [7] that are computed on rotated integral images, our features have the advantage that they can be computed as fast as the basic haar-like features of [9] with the standard upright integral image.

A feature is defined by its type, size and position with respect to a subwindow in an image. Variations in size do not only include scaling but also aspect ratio. This combinatorial multiplicity results in an over-complete set of some hundred thousands up to millions of different features. Every feature becomes a single weak-classifier by computing an optimal CART (classification and regression tree) for the training set. Because our training sets are rather small and generalization capability decreases by depth of the CARTs, we only use stubs. The coordinate system of the subwindow is also normalized to fit into the unit square (Fig. 4). This enables to handle differently stretched object instances in an elegant way.

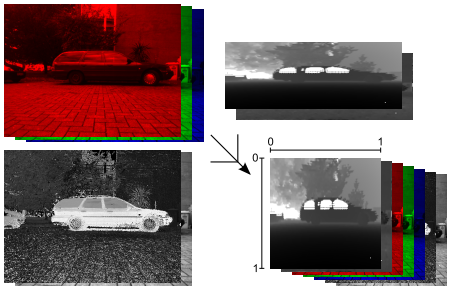

Fig. 3. Conversion of the coordinate system of the sensor layers to unit square. Sensor layers are red, green, blue, hue-distance, intensity, distance and remission.

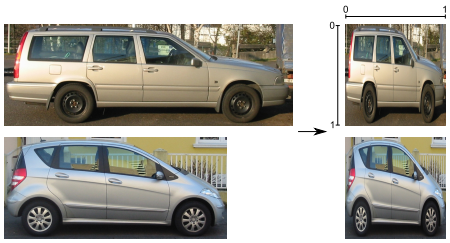

Fig. 4. Object coordinate systems normalized to unit square for handling different aspect ratios.

3 A Joint Feature Pool from Different Sensors

3.1 Sensor Fusion

Our robot is equipped with a color vision camera and a rotating SICK LMS 291 laser scanner. The beams of the laser scanner uniformly meter the encircled sphere. By clocking their time of flight a laser scanner supplies accurate distance information, and by quantifying the amount of laser-emitted light that is reflected or scattered back by the surfaces in the scene it provides remission information. Thus our robot is able to perceive an RGB image and a 3D point cloud of its surrounding area. After a conversion from spherical into Cartesian coordinates the visible part of the point cloud is projected onto the image plane of the camera (cf. Fig. 2). Note that the centers of reception of the sensors should be as close together as possible to avoid wrong assignments in case that an object is exclusively seen by one of the sensors. Although we use a lower resolution for the image layer of the laser, the layer is not densely-packed and we need to interpolate the missing values. In our case, we have to additionally correct the barrel distortion of the camera, before we obtain correctly correlated sensor layers.

The coordinate system of a sensor layer l is normalized by its width w_l and height h_l to fit into the unit square,

$$|(x, y)|_l = \left(\frac{x}{\mathrm{w}_l}, \frac{y}{\mathrm{h}_l} \right).$$

Thus, the position of an object is specified by the same coordinates in every layer, even if they vary in physical resolution (Fig. 3).

Altogether, we use seven sensor layers: red, green, blue, intensity and hue-distance from the camera, and distance as well as remission values from the laser (cf. Fig. 8). The hue channel from HSV color space encodes pure color information without saturation and brightness as an angle in a color wheel.

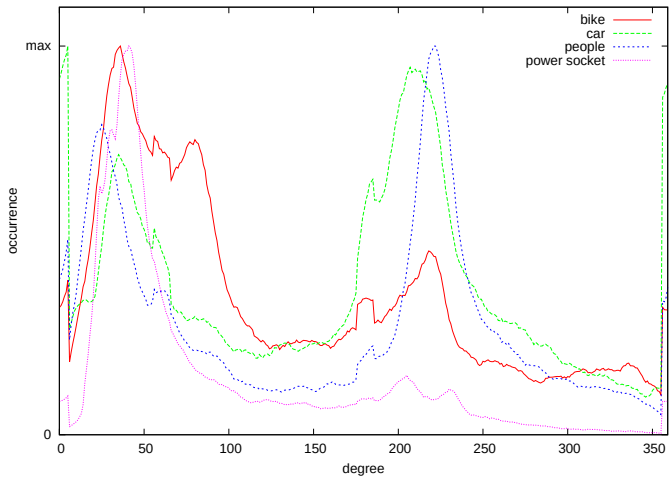

Fig. 5. Distribution of hue occurrence for object classes (10 degree smoothed average).

However, the computation of weak-classifiers from Haar-like features needs a totally ordered set which is naturaly given in case of continuous intensity values, but not in case of angles. Therefore we do not use angles directly. Instead, we use the distance between the hue angles and one specifically chosen hue angle. This particular value depends on the object type and is calculated from the positive training examples: we choose the most frequent hue by building a coarse additive histogram (Fig. 5). If the color distribution of an object type has only a few strong peaks, this choice tends to be more reliable than the use of a single predetermined hue value, because that constant reference hue could be located between two strong peaks and thus would not allow to discriminate between those peaks in the hue distance norm. Otherwise, if the color is uniformly distributed for an object type, there is no difference.

3.2 Integration of Different Sensors

A straightforward approach to exploit the information of different sensors is to train one classifier for each sensor and somehow join the outcome. For instance Frintrop et al. linked range and remission based classifiers by a logical "and" operator [18]. Instead of this, our approach is to learn only one classifier per object type that uses information of all sensors simultaneously (cf. Fig. 6). Because of our mapping to unit square coordinates, every feature is well defined on every sensor layer. Now it is a rather natural extension to enlarge the pool of candidate classifiers to all layers. It is up to AdaBoost to decide which weak-classifier from which sensor is best.

Thus, it only depends on the characteristics of the object type how much a certain sensor layer contributes to the final decision. The most distinctive features from every sensor are estimated and combined to supplement each other.

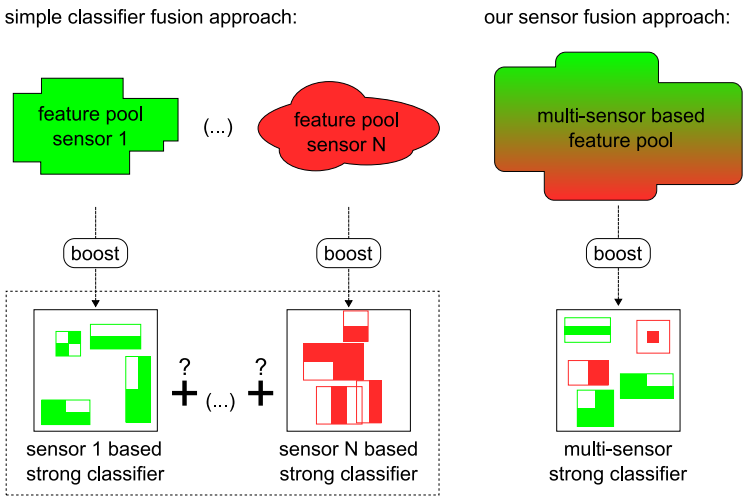

Fig. 6. Overview to our approach: utilizing AdaBoost to combine multiple sensors in a single strong classifier.

Furthermore the computational effort for classification is much lower. On average, one can expect that a multiple sensor classifier cascade is much smaller than each single sensor cascade used in the decoupled approaches, because they achieve a lower average classification error of selected weak-classifiers. A heuristic for combining the cascades of different sensors is no longer needed in this approach, therefore, it is also more general.

4 Experiments

We conducted experiments to discover if and how much a classifier benefits from our training on merged sensor data and from the new corner-like features. For this purpose, we built a training and a test set[4] for four different object categories: people, cars, bicycles and power sockets. Table 1 shows their configuration. An example is a labeled rectangular region inside of an image. During learning negative examples are bootstrapped from those parts of training images that do not contain positive examples. We also add images that do not contain positive examples but only clutter to enlarge the choice for negative examples.

First, we trained one classifier with all Haar-like features (cf. Fig. 1) on all sensor layers (cf. Fig. 3) for each object type. Then we decomposed those cascades into weak-classifiers and summed up their weights by sensor layer to discover how much a certain layer contributes to the final decision of the cascades. Fig. 7 shows the results of these experiments. It can clearly be seen that the laser distance

[4] Note that because of our special sensor fusion we cannot use common benchmark tests. More example pictures from training and test sets can be found in [20].

Table 1. Components of training and test sets (number of positive examples / images).

	car	people	bike	power socket
training	115 / 191	115 / 173	95 / 180	66 / 137
test	34 / 35	49 / 30	31 / 29	27 / 36

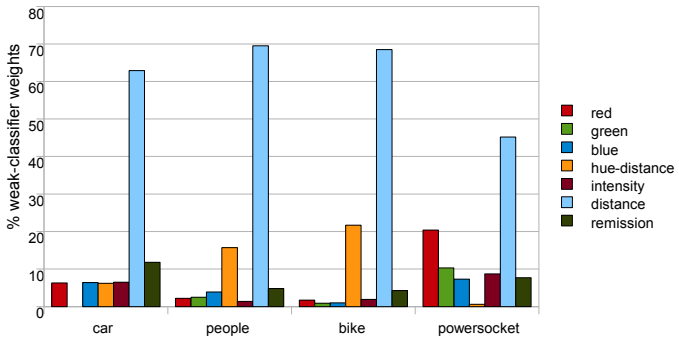

Fig. 7. Distribution of weak-classifier weights by sensor layers.

layer is favored over all others. Power sockets themselves are complanate at a wall and hardly visible in distance data, but the distance data is still useful for rejecting non-flat negative examples. In this case sensor fusion pays off most, weak-classifiers are shared equally between the sensor layers of laser and camera. Categories with a more protruding shape, namely car, people and bike, benefit still more from the distance layer. It can also be seen that hue-distance as a unified layer for color features is a worthwhile extension. While the utility of red, green and blue layers on their own seem to have strong correlation with each other and with intensity, hue-distance adds some unique information.

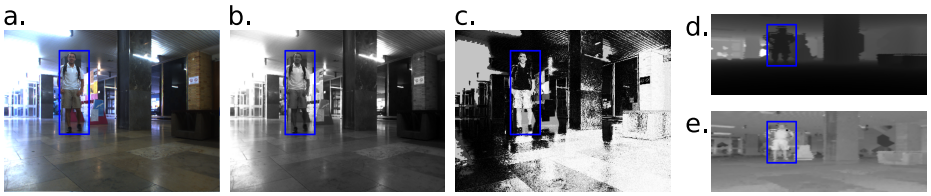

Fig. 8. (a.) Person classified with a cascade trained with features from intensity (b.), hue-distance (c.), distance (d.) and remission (e.) data.

In further experiments, we restrict the pool of candidate features to certain types or sensor layers to compare performances. Some of the sensor layers with

Table 2. Results of classifiers trained on all sensor layers. Length is number of combined weak-classifiers.

	car	people	bike	p.sock.
length	47	77	99	142
recall	1	0.98	0.97	0.85
precision	1	1	0.65	0.88
F-measure	1	0.99	0.78	0.87

Table 3. Results of classifiers trained on all sensor layers without the corner-like features.

	car	people	bike	p.sock.
length	62	111	162	142
recall	0.94	0.98	0.94	0.89
precision	0.94	0.98	0.69	0.63
F-measure	0.94	0.98	0.79	0.73

a classified person are shown in Fig. 8. We evaluate the performance with recall, precision, and the F-measure. Recall is a measure of completeness, whereas precision is a measure of exactness. Because they are interdependent opposite goals, the F-measure combines them into a single value by calculating the harmonic mean.

First, we evaluate the gain of corner-like features by training classifiers with and without them. We found that classifiers with all features are composed of 31% corner-like features on average. The classification results on our test sets are shown in Tables 2 and 3. As can be seen, the performance of cascades decreases and/or their length increases if corner-like features are omitted. Since we are interested in a real-time capable application for our mobile robot, shorter cascades are an important advantage. Thus, all subsequent experiments are carried out including our corner-like features.

The next experiment compares the exclusive ability of laser and camera for object recognition. For each object type one cascade was trained with the distance and remission layers from the laser sensor and one with the intensity and hue-distance layers from vision. Tables 4 and 5 show the results. Two observations can be made. First, performances of classifiers from single sensors are worse than those from all sensors and cascades grow considerably in length (cf. Tab. 2). For example the sizes of car classifiers increased by 236% (laser only) respectively 853% (camera only) while the F-measures slightly decreased. Second, performance of a sensor depends on object categories. It shows that the power socket category performs better with data from the camera than with data from the laser while car, people and bike categories show better results and shorter cascades with laser data.

After this we evaluate our sensor fusion approach against the approach to first learn separated classifiers and then fuse the results (cf. Fig. 6). We learned cascades with merged hue-distance, intensity, distance and remission layers and compared them to classifiers generated by a logical "and" concatenation of the cascades from camera-only and laser-only data (Tab. 4 and 5) as proposed in [18]. Tables 6 and 7 comprise the results on our test sets. It is self-evident that linking classifiers by a logical "and" can only improve precision, but possibly degrades the recall. This also arises in our experiments: while results from power sockets and cars perform comparably well with both methods, our approach shows superior results on the people and bike categories. It is able to exploit the pros of every sensor, thus can improve precision and recall. Furthermore,

Table 4. Results of classifiers trained only on laser data (distance and remission layers).

	car	people	bike	p.sock.
length	111	123	135	418
recall	1	0.96	0.97	0.85
precision	0.89	1	0.6	0.29
F-measure	0.94	0.98	0.74	0.43

Table 5. Results of classifiers trained only on camera data (intensity and hue-distance layers).

	car	people	bike	p.sock.
length	401	1008	373	172
recall	0.94	0.67	0.52	0.89
precision	0.89	0.97	0.59	0.86
F-measure	0.91	0.8	0.55	0.87

Table 6. Results of classifiers trained on distance, remission, intensity, and hue-distance layers.

	car	people	bike	p.sock.
length	58	128	108	87
recall	1	0.98	0.94	1
precision	1	1	0.78	0.75
F-measure	1	0.99	0.85	0.86

Table 7. Results of classifiers from Tab. 4 linked by logical "and" operator with classifiers from Tab. 5.

	car	people	bike	p.sock.
length	512	1131	508	590
recall	0.94	0.65	0.55	0.74
precision	1	1	0.74	1
F-measure	0.97	0.79	0.63	0.85

in theory a strong-classifier trained on the same examples but with a super-set of candidate weak-classifiers has at most the same size as one trained with less weak-classifiers. Therefore, the execution time of n concatenated single sensor classifiers is on average at least n times longer than the time spent by one classifier trained with a joint feature pool. In practice, the sensor fusion approach proposed here is more than twice as fast as the "and" concatenation of classifiers.

5 Conclusions and Future Work

In this paper, we have introduced a new approach to combine the significant parts of information from different sensors for object classification. Gentle AdaBoost selects the best weak-classifiers built from a joint feature pool from different sensors. In several experiments, we have shown that the presented approach outperforms results from separate sensor classifications as well as from a simple fusion of separately trained cascades. The proposed approach is generic and can be applied to other object types as well.

Moreover we have shown that corner-like features are a reasonable extension of the standard feature set. In our experiments, classifiers trained with corner-like features outperform those trained without. Already the fact that they have been selected by Gentle AdaBoost proofs their advantage. The same holds for our new feature channel based on hue-distances.

While the test and training sets in our experiments are comparably small (a fact that is explained by the time-consuming acquisition of sensor data with our mobile robot), our results imply that these sets are big enough to define the characteristics of object types and to classify objects reliably. In future work we

will investigate if larger training sets can even improve the performance. We also plan to use classifiers trained with this approach as initializer for object tracking to go one step further towards autonomous behavior of mobile robots.

We meet another challenge with improving the sensors. Acquisition time for the laser is much to long ($\approx$ 4 seconds per frame) to record data of fast moving objects or while driving with the robot. We will examine the applicability of TOF camera *ZCAM* or its successor *Natal* to tackle this drawback.

References

1. Schapire, R.E.: The strength of weak learnability. Machine Learning **5** (1990)
2. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Technical report, AT&T Bell Laboratories (1995)
3. Freund, Y., Schapire, R.E.: A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence **14**(5) (September 1999) 771–780
4. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Annals of Statistics **28** (2000) 2000
5. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. of CVPR. (2001)
6. Viola, P., Jones, M.: Fast and robust classification using asymmetric adaboost and a detector cascade. In: NIPS. (2002)
7. Lienhart, R., Maydt, J.: An extended set of haar-like features for rapid object detection. In: Proc. International Conference on Image Processing. (2002) 900–903
8. Meir, R., Rätsch, G.: An introduction to boosting and leveraging. In: Advanced Lectures on Machine Learning. LNCS. Springer (2003)
9. Papageorgiou, C.P., Oren, M., Poggio, T.: A general framework for object detection. In: Proc. of Int'l Conf. on Computer Vision, IEEE Computer Society (1998)
10. Huang, C., Al, H., Wu, B., Lao, S.: Boosting nested cascade detector for multi-view face detection. In: Proc. of the Int'l Conf. on Pattern Recognition. (2004)
11. Mitri, S., Frintrop, S., Pervölz, K., Surmann, H., Nüchter, A.: Robust Object Detection at Regions of Interest with an Application in Ball Recognition. In: IEEE 2005 International Conference Robotics and Automation (ICRA '05). (2005)
12. Barreto, J., Menezes, P., Dias, J.: Human-robot interaction based on haar-like features and eigenfaces. In: Proc Int. Conf. on Robotic and Automation. (2004)
13. Laptev, I.: Improvements of object detection using boosted histograms. In: Proc. British Machine Vision Conf. (BMVC'06). (2006)
14. Arras, K.O., Mozos, O.M., Burgard, W.: Using boosted features for the detection of people in 2D range data. In: Proc. of ICRA. (2007)
15. Premebida, C., Monteiro, G., Nunes, U., Peixoto, P.: A lidar and vision-based approach for pedestrian and vehicle detection and tracking. In: Proc. ITSC. (2007)
16. Zhang, L., Wu, B., Nevatia, R.: Pedestrian detection in infrared images based on local shape features. In: OTCBVS Workshop, CVPR. (2007)
17. Zivkovic, Z., Kröse, B.: Part based people detection using 2D range data and images. In: Proc. of IROS. (2007)
18. Frintrop, S., Nüchter, A., Surmann, H., Hertzberg, J.: Saliency-based object recognition in 3D data. In: Proc. of Int'l Conf. on Intelligent Robots and Systems. (2004)
19. Nüchter, A., Lingemann, K., Hertzberg, J., Surmann, H.: Accurate object localization in 3D laser range scans. In: Int'l Conf. on Advanced Robotics. (2005)
20. Klein, D.A.: Objektklassifizierung in fusionierten 3D-Laserscans und Kameradaten mittels AdaBoost. Master's thesis, University of Bonn (12 2008)