ALBERT-LUDWIGS-UNIVERSITY FREIBURG
FACULTY OF APPLIED SCIENCES

Department of Computer Science
Autonomous Intelligent Systems Lab
Prof. Dr. Wolfram Burgard

Scene Analysis from Range Data

Master's Thesis

Author: Felix Leon Endres

Submitted on: February 13, 2009

Supervisors: Prof. Dr. Wolfram Burgard
Dr. Cyrill Stachniss
Dr. Christian Plagemann

Abstract

In this thesis, we present a new approach to unsupervised discovery of object classes in 3D range scans. Our assumption is that a real-world scene is composed of a finite set of objects, each related to exactly one object class, plus irrelevant background structure. We do not require prior knowledge about the set of possible object classes other than that objects from the same class are more similar in shape than those from different classes.

We propose similarity-based clustering and classification of objects using the distribution of discretized local free-form surface signatures. In contrast to common methods, our approach does not rely on manually created models or training data. Instead, the class models will be discovered from the similarities in the feature distributions of the segmented range data. To learn the model, we apply latent Dirichlet allocation (LDA), a probabilistic framework for topic modeling. We show that LDA successfully learns a partitioning of the feature distributions into groups based on similarity. To evaluate the performance of this approach, we carry out experiments with two data sets of differing complexity and compare the results to an approach based on hierarchical clustering of the feature distributions. In our experiments, we demonstrate that the results based on LDA are superior to those based on hierarchical clustering, in particular for the more complex data set, containing similar object classes, varying object appearances and complex objects.

Zusammenfassung

In dieser Arbeit wird ein neuer Ansatz zur unüberwachten Aufteilung von 3D-Objektdaten in Gruppen vorgestellt. Dieser Ansatz wird auf Szenen angewendet, die mit einem Laserscanner abgetastet wurden. Wir gehen davon aus, dass die Szenen aus einer Menge von Objekten bestehen, sowie aus einem für unsere Zwecke irrelevantem Hintergrund. Die Objekte gehören dabei jeweils genau einer Bedeutungsklasse an. Der Ansatz bedarf keiner Information über die Eigenschaften der Klassen, es wird jedoch vorausgesetzt, dass die Objekte einer Klasse sich in ihrer Form ähnlicher sind, als Objekte unterschiedlicher Klassen.

Der vorgeschlagene Ansatz basiert auf lokalen Oberflächenmerkmalen, die beliebige Formen charakterisieren können. Die Merkmale werden diskretisiert und segmentweise zu Merkmalsverteilungen zusammengefasst. Diese werden in Gruppen aufgeteilt, ohne dass gruppenspezifische Informationen bereitgestellt werden müssen. Dieses Ziel wird unter Verwendung von “latent Dirichlet allocation” (LDA), eines statistischen Modells zur Gruppierung von diskreten Daten, erreicht. Zur Evaluierung der Methode werden Experimente mit verschiedenen Datensätzen vorgestellt. Die Datensätze unterscheiden sich dabei sowohl hinsichtlich der Komplexität der Objekte, als auch im Schwierigkeitsgrad bezüglich der Ähnlichkeit zwischen den Objektklassen und dem Variationsgrad innerhalb der Klassen. Im Vergleich mit experimentellen Ergebnissen, die unter Verwendung von hierarchischem Clustering erzielt wurden, erreichen wir bessere Resultate; sowohl hinsichtlich der Anzahl korrekt gruppierter Segmente, als auch bezüglich der Robustheit gegenüber Variationen in der Merkmalsextraktion.

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Felix Leon Endres — Freiburg, den 7. Januar 2009

Acknowledgment

At this point I want to thank everyone that helped me with this work, especially my supervisors Prof. Dr. Wolfram Burgard, Dr. Cyrill Stachniss and Dr. Christian Plagemann for the interesting topic, their support and ideas. I also would like to thank all people of the “Autonomous Intelligent Systems” group in Freiburg, in particular Jürgen Hess for the constant exchange of ideas and for sharing his set of tools and templates, Bastian Steder for his support with information and data and Kai Wurm for his help with data acquisition. Further I want to express my gratitude towards my family, who always supported me in my studies, and to my beloved girlfriend Lena Wenz. Many more helped in various ways and I am very thankful for that.

Contents

1 Introduction

A skill usually taken for granted by humans is the ability to recognize what we perceive with any of our senses. We will recognize things we saw, heard, smelled, tasted or felt before, mostly limited by our ability to memorize. If something has not been perceived before, we reliably relate our perceptions to similar things encountered in the past. This allows us to apply our experience with something similar to the new object. This capability is crucial for even the simplest tasks.

Unfortunately, the identification or classification of objects in sensor information is a very hard task for computers, if the input data does not follow a fixed set of rules (e.g. recognizing block letter with fixed font and size). For most objects we encounter and deal with in our everyday life, there are no definite rules of appearance. But although tasks, such as the identification of cars in photographs, poses no problems to us, we are not able to describe the subconscious process that lets us distinguish different classes of objects.

And while the state of the art in computer based object classification has progressed remarkably in recent years, our own superior capabilities show that much room is left for improvement.

Object recognition is of particular importance in the field of autonomous mobile robotics. Many tasks we would like to delegate to robots require the actor to distinguish between different kinds of objects. Such tasks can range from applications in service robotics, e.g a catering robot that has to identify specific kinds of tableware, to industrial applications where the robot has to distinguish a set of arbitrary products.

An additional benefit if objects can be identified from observation data is, that they can also be segmented from the rest of the image. This is especially useful in 2D images, when spatial information about the objects is limited. Yet it is also useful for automatic segmentation of 3D images where objects are connected or adjacent. Here clustering on spatial distance measurements would not be applicable. And even though object classification might appear to be easier for 3D range data than for 2D image data because the depth is known for all measurements, there is the lack of color information and data is often noisy and sparse. Figure 1.1 shows one example of an 3D scan. Objects of interest have been labeled with a color consistent to the object's class.

Further applications for recognition arise due to the increasing usage and availability of 3D models on the internet. Analog to image search engines, that can find and group images based on similarity the need for grouping and finding 3D data based on similarity, will grow with the amount of models available.

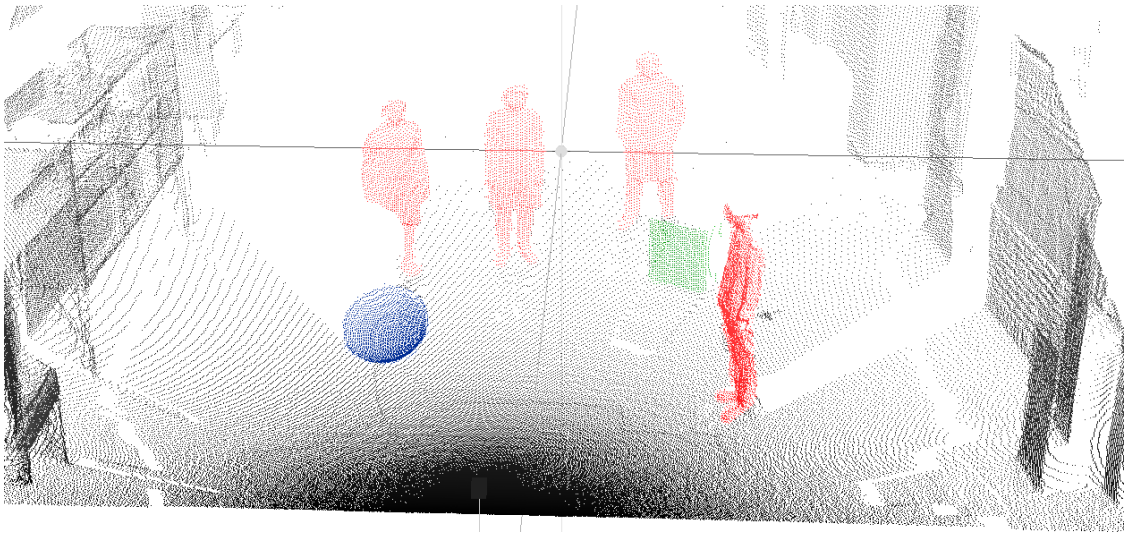


Figure 1.1: Example 3D range data with manually colored objects

In general, the first step for object recognition is to transform the data to a feature space. Instead of working on the raw data—a point cloud or a mesh—most recognition approaches extract features, that describe the characteristic object properties as concisely as possible. The classification is then performed solely upon those features. To allow for efficient comparison between data sets, most features reduce the data to a much simpler representation. There are many proposals for features differing in the characteristics described, in dimensionality and whether the whole data is described or only a local subset. Another important property of a feature type is, whether it can represent all kinds of shapes or just a subset, such as ellipsoids. The latter type is usually of considerably lower complexity than free-form feature types, but only suitable to describe certain kinds of objects.

From those features a *model*, can be built, against which new data can be compared and assessed. This thesis uses and enhances a suitable 3D shape descriptor for the representation of free-form objects and evaluates classification based on these representations. We will use the distribution of the features to calculate the class(es) the data belongs to. This is visualized in Figure 1.2, where the distributions of imaginary features are shown. We know the distributions for the object classes human and chair. Given a new distribution we need to decide what class it belongs to.

An important distinction between current classification methods is the way the models of the objects or classes to identify are created. The model can be engineered manually, learned from a set of labeled example data (which is called supervised learning) or learned from unlabeled data using an unsupervised clustering algorithm. While the former two categories have the advantage,

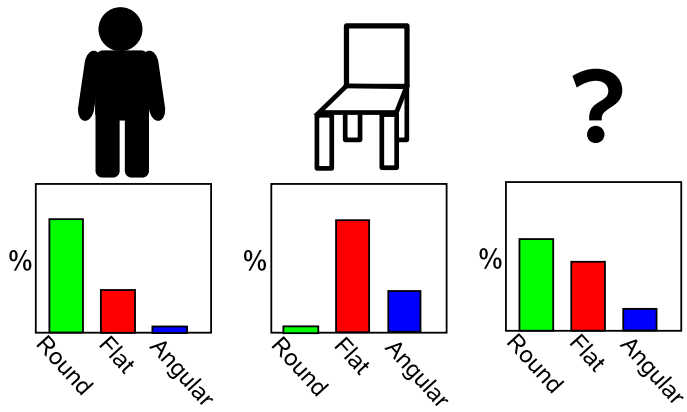


Figure 1.2: With features that only distinguish between round, flat and angular surfaces, we can estimate the type of object by analyzing the distribution of the feature

that detailed prior knowledge about the objects to identify can be included easily, the effort for manually building the model or labeling a significant amount of training data becomes infeasible with increasing model complexity and larger number of objects to identify. Furthermore, in applications where the objects to distinguish are not known beforehand, a robot needs to build the model by means of which it classifies the data on its own.

Unsupervised classification can be accomplished, e.g. by extracting features for which a distance measurement can be defined. The features are then assigned to classes upon a criterion such as minimizing the distance between features of the same class, while maximizing the distances to features of other classes. The main difficulty is then to find features and distance measures that reflect the information about the similarity of the underlying data.

Since data usually is structured in units, e.g. of separate images or 3D scans, and assuming that correlated features will mostly occur together it is possible to find classes upon the co-occurrence of features. This renders a feature specific distance measurement unnecessary, as long as we can decide whether two features are equal—or similar enough. There have been various models to describe such a problem in the text domain, where co-occurrences of words are regarded to find common topics. In this thesis we use a recent topic modelling approach—*latent Dirichlet allocation*—for unsupervised discovery of object classes, based on surface features.

1.1 Goal

The goal of this work is to develop an approach for classification of objects from 3D range data. Similar objects should be identified as belonging to the same class, and vice versa, different types of objects should be labeled distinctly.

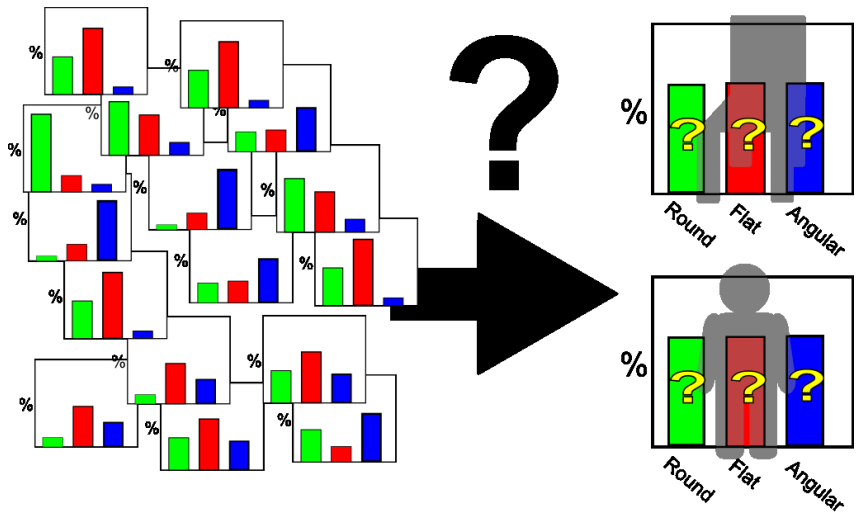


Figure 1.3: We want to organize a set of distributions of unknown origin (left) into classes based on similarity

Furthermore we will restrict our methods to not rely on labeled training data or a predefined model that will be matched, but the presented method should find the model from the data itself. So, the first problem we deal with is the mentioned transformation of the data to a suitable feature space, to generate meaningful feature distributions with respect to the represented objects. The second problem is the grouping of data sets into classes of objects and the determination of the distribution over the feature distributions that characterize the class. This latter task is illustrated in Figure 1.3.

1.2 Outline

This thesis is structured as follows: After this introduction, Chapter 2 presents work that is related to our approach in methodology or objectives. Chapter 3 describes the advanced methods we use for learning, estimation and feature extraction. Subsequently we present our approach in Chapter 4, including data preprocessing, feature extraction and model generation. In Chapter 5 the experiments conducted and the results found are described in great detail, including the equipment we used and objects scanned. The last chapter summarizes our findings and gives an outlook on future work and possible extensions to our work.

2 Related Work

Object recognition is a classic problem that has been well studied for over 60 years. Although it was first thought to be an easily accomplished task, it can not be considered a solved problem today. The approach presented in this thesis is based on a lot of preceding research and related to much of the work that is currently undertaken in the fields of computer vision (including 2D as well as 3D) and machine learning. This chapter is dedicated to the previous work we build up on and to the work of others that is similar in some way. For the latter we will shortly clarify the points of distinction to our work. The research area this work is embedded in, is illustrated in Figure 2. On top, objectives are shown that are related to our work. In clustering—also known as segmentation— the goal is to assign the data to a finite number of classes, such that criteria of interest are shared by the data within a class, but differ among classes. Hence, the properties that define each cluster have to be selected. A common criterion in clustering is spatial proximity. In this work, we cluster segments of 3D range data, based on their similarity in shape. This allows us to create groups of similar objects. Recognition is similar to clustering, in that data is classified on particular criteria, e.g. similarity in shape or appearance. Usually, in recognition the definition of the class that shall be recognized is either known beforehand or is learned by positive and negative examples. The definition of the class is also referred to as its *model*. Recognition can also be applied using the cluster definitions found by clustering as models. In this case, the cluster definitions need to be applicable to unseen data. In retrieval, no model needs to be learned. Here, given some set of data, the task is to find a corresponding—usually the most similar—data set from a database. Applications include image search by similar appearance [Siggelkow *et al.*, 2001] and retrieval of 3D models, based on shape [Shilane *et al.*, 2004]. If the database contains data with labels, this can also be used for recognition, by classifying the "search data", e.g. with the label of the nearest neighbor. In reconstruction, the goal is to combine sets of partial data to a complete data set. This can be used to construct models for recognition tasks from partial views, for instance due to (self) occlusion. The differences, with regard to prior knowledge about the class model are shown explicitly in the bottom right part of Figure 2. Section 2.2 presents current research with the mentioned goals.

At the bottom left of the illustration, data commonly encountered in problems in robotics and machine learning is shown. Next to it, the task of feature extraction is shown. Feature extraction is a common preprocessing step, if classification is based on shape or appearance. Features aim to efficiently capture

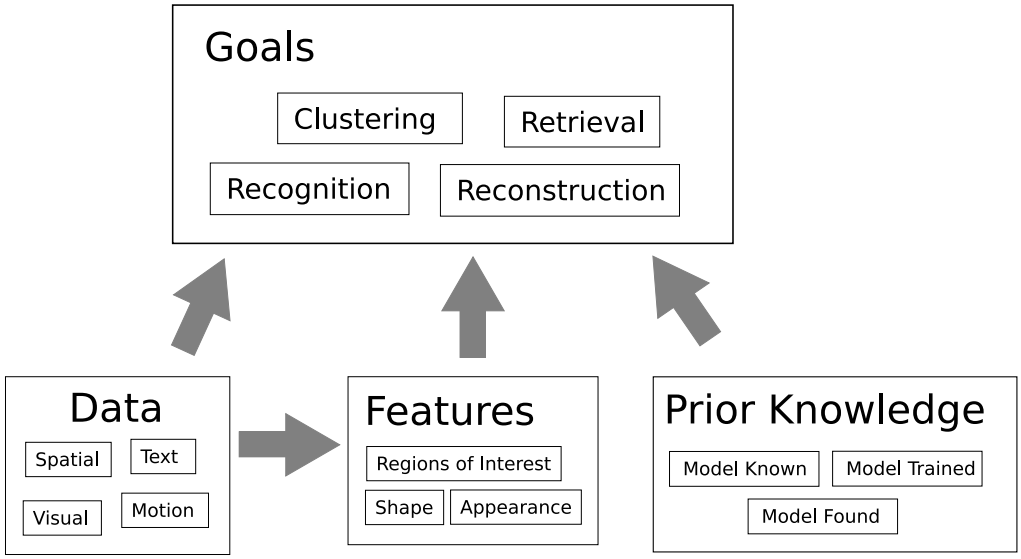


Figure 2.1: Illustration of related research areas.

characteristics of interest. Ideally a feature is invariant to variations that are irrelevant to classification. There is a lot of research in this area. An overview is given in Section 2.1.

2.1 Data and Feature Extraction

As language directly encodes semantic information in words, feature extraction in the text domain is mostly reduced to the task of suppressing words with no or little semantic value. This is usually done using lists of "stop words", which are deleted from the documents beforehand. On the contrary, the extraction of features with representive information about objects in 2D or 3D data is a much harder task and targeted by a lot of current research.

3D Data and Shape Features

Three dimensional data can be created manually, e.g using CAD software or computed from data, such as in construction of protein models. In the field of robotics, 3D data is mostly captured using time of flight sensors, such as laser range scanners, specialized cameras, radar or even sonar. Other possibilities include 2D sensors, where the third dimension is reconstructed, i.e. in multiple view computer vision. Unfortunately, these capturing methods yield rather sparse data, when compared with high resolution 2D imaging processes. Only in recent years, laser range scanners have become available that provide sufficient data density for applications such as object recognition from shape.

Usually this provides us with data in form of a point cloud or a polygonal mesh.

In contrast to artificially created data, two scans cannot be matched directly—even if they contain the same object. On the one hand the input data is not a perfect description of the scanned objects. Problems include smoothness (i.e. there is only a finite number of points/polygons) and precision. Therefore it is crucial to accommodate the matching process to discrete and erroneous input data, by using matching techniques such as the iterative closest point algorithm. Also incompleteness of the data, e.g. because of (self-) occlusion can rarely be avoided and has to be considered in the matching process. On the other hand, differences due to translation and rotation have to be disregarded, as the position of the object relative to the sensor is of no regard to the objects identity or class.

For these reasons a basic step in object recognition systems is the extraction of features which characterize the objects sufficiently. This allows us to efficiently distinguish between what is considered the same and what is not. The features should be robust to noisy data and—ideally—invariant under a similarity transform.

Important measures for such features are ambiguity, conciseness and uniqueness [Brown, 1981]. Ambiguity measures how exact the feature represents the data, i.e. the level of variation in the data that is mapped to the the same feature representation. Conciseness is a measure of the compactness of representation and therefore closely related to the efficiency with which operations such as matching can be performed. A feature is unique if there is only one feature representation for given object data. Non-uniqueness is usually introduced during the construction of the object, e.g. by aliasing.

Early works mostly segment and characterize point clouds or mesh data by fitting parameterized basic geometric shapes such as planes, spheres, quadrics, etc to the data (see [Hoover *et al.*, 1996] for an overview). The parameter set that results in the best fit is then used to characterize the objects shape. This proceeding yields very compact, yet ambiguous, features with particular difficulties to represent complex shapes.

Therefore the paradigm has shifted towards free-form surface representations. Spin images are a very popular kind of surface descriptor in this area. Spin images use the surface normal to devise a 2D coordinate system that is invariant under similarities and describe the surface of the object relative to this coordinate system. It was introduced by Johnson [1997] and has been applied successfully to supervised object recognition tasks in [Johnson and Hebert, 1996], [Johnson and Hebert, 1998], [Johnson and Hebert, 1999]. In this thesis, we apply spin images to our task and also propose an enhanced surface descriptor, which is very similar to spin images. We describe spin images in detail in Section 3.6. Our approach is described in Section 4.3. The basic difference is, that instead of storing a description of relative coordinates of the surface, we store the angle between the surface normals of two points.

Another proposal of a shape descriptors are presented in Ruiz-Correa *et al.* [2003], which describes a similar per-point-feature for recognition of object classes. It relies on symbolic labels that are assigned to regions. The symbolic values, however, have to be learned from a labeled training set beforehand. Stein and Medioni [1992] present a point descriptor that also relies on surface orientations. It focuses on the surface normals in a specific distance to the described point and models their change with respect to the angle in the tangent plane of the query point.

For more information on 3D shape descriptors the reader is refferred to [Bustos *et al.*, 2005] and [Campbell and Flynn, 2001].

2D Data and Appearance Features

Images are considerably easier to capture and process and visualize than 3D data. Image data is ubiquitous in our everyday life and huge amounts of image data is available through the internet. Object classification, identification and tracking in images are thoroughly studied problems and experience a lot of current research. Most current methods rely on feature detectors to find locations of interest and the subsequent extraction of features from these locations. The features are then used in learning algorithms, to find the correspondence between features and objects. As in this work, the feature distribution of an image often is considered be a "bag of words", i.e. the spatial distribution of the features is disregarded, to allow for a simpler representation [Csurka *et al.*, 2004], [Zhang *et al.*, 2007], [Bosch *et al.*, 2006].

As for 3D data, there is a huge variety of proposed features. Again, invariance to translation and rotation (around the principal axis) are required. Due to the loss of depth information 2D features should also be invariant to scale, as it is dependent on the distance of the object to the camera, and in the ideal case to perspective transformations. The former is often achieved using pyramids, i.e. extracting the features from the image in several scale levels. The latter, however, can only be approximated by robustness to affine transformations. Further important properties are robustness to different lighting conditions and to noise. Noise is usually dealt with by convolution of the image with a Gaussian function. Popular feature proposals include SIFT [Lowe, 1999], histogram of gradients[Dalal and Triggs, 2005] and SURF [Bay *et al.*, 2008]. A comparison of 2D image features can be found in [Mikolajczyk and Schmid, 2005].

2.2 Classification Approaches

Classification on 3D Data

Even though laser range scanners are very accurate, range data is always subject to small measurement noise. Features extracted from 3D data cannot per-

fectly describe an object with respect to ambiguity and uniqueness. Also, specific objects to be identified are usually subject to partial (self-) occlusion and might have a varied appearance. When a class of objects should be recognized these variations are part of the task and have to be considered. To accomplish recognition advanced techniques are necessary. As mentioned before, object recognition approaches differ in the process of model creation. Steder *et al.* [2009], for instance, use a reference scan segment as model and succeed in matching it to unseen data. An interesting approach to object recognition, using probabilistic techniques as well as histogram matching , has been presented in [Hetzel *et al.*, 2001]. The features used are particular low dimensional. Unfortunately the experiments have only been done on synthesized data. In contrast to our work, a complete model of the exact object to recognize is assumed to be available. A lot of research focuses on supervised algorithms that are trained to distinguish objects or object classes on a labeled set of training data. Anguelov *et al.* [2005] and Triebel *et al.* [2006] use supervised learning to classify objects and associative Markov networks to improve the results by explicitly considering relations between the class predictions. In contrast to the mentioned approaches, our goal is to find the object classes by the similarity of the data itself, such that no human supervision for model selection is necessary.

An approach to unsupervised segmentation of adjacent rectangular boxes in 3D scans has been presented in [Schroll, 2008]. Here, the point cloud is projected to a 2D height map. This height map is segmented on behalf of height and gradient using Markov random fields.

The work of Ruhnke [2008] proposes an approach to unsupervised model learning of models in 3D range data. His focus lies on reconstructing a full 3D model by registration of several partial views, while our approach focuses on the clustering of objects based on similarity. Ruhnke works on range images, from which he selects small patches with an region of interest detector. The patches are used to find candidates for the transformation of the partial views to a common coordinate system. Several heuristics are used to select the best transformation.

Streicher [2008] presents a method, aimed at model retrieval in databases of 3D models. In contrast to our work, Streicher first determines points of interest for each model, then clusters the features from these points into codebook entries. The features used in his work include spherical harmonics [Fehr and Burkhardt, 2007] and light field descriptors [Chen *et al.*, 2003]. Model retrieval is then done with a nearest neighbor search, based on histogram matching as distance function.

Triebel *et al.* [2007] presented an approach to supervised learning of 3D models, combining nearest neighbor classification with associative Markov networks to overcome limitations of the individual methods. The approach also uses spin images as surface descriptors. The method requires a training data set with the descriptor and the class label for all point measurements. From this data, the

classification algorithm is trained to predict the class of unseen data.

Classification on Text Data

The main focus of research on text data is information retrieval, but also topic modelling, i.e. semantic clustering of documents. In the domain of unsupervised classification of text documents, several models that greatly surpass mere counting of words have been proposed. Recent successful methods include probabilistic latent semantic indexing (PLSI) [Hofmann, 1999] and latent Dirichlet allocation (LDA) [Blei *et al.*, 2003] that both use the co-occurrence of words in a probabilistic framework to group words into topics. As shown by Girolami and Kabán [2003], LDA supersedes PLSI; the latter can be seen as a special case of LDA, using a uniform prior and using maximum a posteriori estimation for topic selection.

Classification on 2D Data

A common approach to locate objects in images is the sliding window method [Bosch *et al.*, 2007], [Ferrari *et al.*, 2008], [Rowley *et al.*, 1996], [Fritz and Schiele, 2008]. First a classification algorithm is trained with positive and negative examples. To locate an object in an image, a rectangular bounding box is "slid" over the image and a classification algorithm, such as a support vector machine or k-nearest neighbor, is applied in every position of the rectangle, deciding whether the object is located in this position or not. Unless the positions and sizes of the rectangles are rigorously restricted, this approach is extremely expensive computationally. Lampert *et al.* [2008] therefore proposed a new framework, that allows to efficiently find the optimal bounding box without explicitly applying the classification algorithm to all possible boxes.

Bosch *et al.* [2006] instead uses PLSI for unsupervised discovery of object distributions. These object distributions are then classified using a k-nearest neighbor classifier. In contrast to our approach, this classifies the whole scene, instead of the contained objects individually. LDA also has been successfully used in recent research for unsupervised categorization of objects in images. While usually few text documents contain a huge mixture of topics, a problem encountered in these papers is that images often show objects of many different categories. Wang and Grimson [2007] therefore introduce segmentation into the LDA model. Fritz and Schiele [2008] proposes the sliding window approach on a grid of edge orientations to evaluate topic probabilities on subsets of the whole image. In [Philbin *et al.*, 2008] the model is enhanced such that documents have explicit places for different topics.

3 Fundamentals

This chapter deals with fundamental methods applied in this thesis. Firstly, the Dirichlet distribution will be explained briefly, as it will be a key factor of the learning algorithm described later and the effects of its parameterization will be of concern in later chapters. Secondly, an introduction to the approximation of a probability distribution using Markov chain Monte Carlo with Gibbs sampling is given. Thirdly, we discuss latent Dirichlet allocation, a recent approach to topic modelling that is used for our classification purposes. In Section 3.4 we give an introduction to clustering methods. Following, the principal components analysis is explained briefly, as we use it to find the surface normals in feature generation. Finally we introduce spin images, a data level representation of surface patches which will be used as input to the learning algorithm.

3.1 Dirichlet Distribution

In this thesis, we use multinomial probability distributions to represent the distribution of discrete surface features for an underlying object class. Furthermore we describe the mixture of object classes within a data set by a multinomial probability distribution. A simple example of a multinomial distribution are the proportions of objects in data sets, e.g. 3D scans or a segments thereof. Having object classes “human”, “box” and “chair” we want to know how often each object class occurs. Assume we are sequentially given several such data sets and want to know the probability for each class. We might have an intuition of the distribution before seeing any of the data sets, e.g. because we know that the data sets were captured in an office at night, so there should be more chairs than humans. Yet, with every data set we want to update our belief to incorporate the observations.

Having a prior belief and updating it according to subsequent observations is a common setting in Bayesian statistics. After each observation we calculate the probability distribution for the next observation (the posterior distribution) from the prior belief and the previous observations. For this purpose it is important that the prior distribution can be easily updated to incorporate observations. In mathematical terms this means that we want to express our prior belief as a distribution $P(\theta)$ that is conjugate to the likelihood of our observations $P(x|\theta)$, i.e. the posterior distribution

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{\int P(x|\theta)P(\theta)d\theta} \tag{3.1}$$

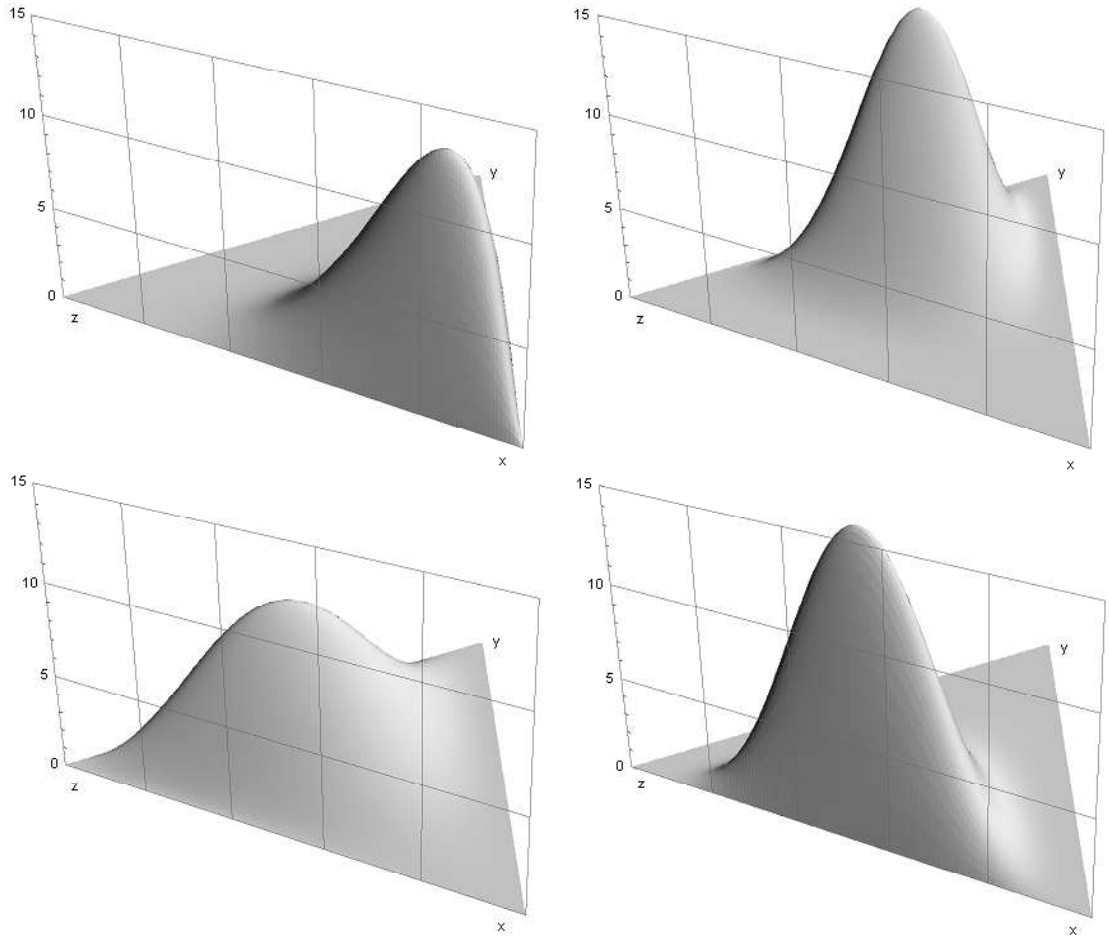


Figure 3.1: Dirichlet Distributions for Different Choices of α. Picture taken from Wikipedia.

is in the same family as $P(\theta)$ itself. For multinomial distributions the conjugate prior is the Dirichlet distribution, a distribution over multivariate probability distributions, i.e. a distribution assigning a probability density to every possible multivariate distribution.

For the multinomial variable $\mathbf{x} = \{x_1, \ldots, x_K\}$ with K exclusive states x_i the Dirichlet distribution is parameterized by a vector $\alpha = \{\alpha_1, \ldots, \alpha_K\}$. The elements of α can be thought to represent $\alpha_i - 1$ observations of state i. The distribution can then be calculated by

$$f(\mathbf{x}) = \underbrace{\frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)}}_{\text{Normalization}} \prod_{i=1}^{K} x_i^{\alpha_i - 1},$$

where $\Gamma(\cdot)$ is the standard gamma function. Note that the elements of $\mathbf{x}$ have

to be positive and sum up to one. Using the Dirichlet as prior for our earlier example, we could express the prior e.g. by $\alpha = \{2, 3, 4\}$, giving lower probability to the class "human", than to "box" and "chair". The resulting Dirichlet distribution $Dir(\alpha)$ is shown in the lower left plot in Figure 3.1. Every corner of the triangle represents a class. The corners itself represent the distributions where only the respective class occurs. The center point represents the uniform distribution over all classes. The triangle itself is the n-simplex containing all points where the $n + 1$ coordinates of all axes sum up to one (which is required for the multivariate distribution).

After observing one human, four boxes and a chair the posterior distribution would become $Dir(\{3, 7, 5\})$ and is shown in the upper right plot in Figure 3.1. This result is easily obtained by adding the observation counts to the elements of α. The same result would of course occur when calculating the posterior with Equation 3.1.

In our use of the Dirichlet distribution, however, we restrict ourself to the subset of symmetric distributions with $\alpha_i = \alpha_j$ for all i and j. Thus, we ignore the possibility to have different probabilities for the classes. Our use will focus on drawing samples from Dirichlet distributions that assign high probabilities only to the outer regions, i.e. distributions that favor a subset of classes instead of a mixture. This is achieved by choosing the α_i to be in the range $(0, 1)$. The resulting distribution will be a valley with peaks at the corners. Thus, when sampling a multinomial probability distribution from the Dirichlet, we expect to draw a distribution that assigns high probability to one or few of the classes and very low probability to the rest. In our example above, we therefore expect to either see only humans, only chairs or only boxes—but would be very surprised to see all of them.

The calculation of the expected probability distribution over the states and its variance can be easily performed based on the values in α. With $\alpha_0 = \sum_j \alpha_j$, the expected probability for a state i is given by α_i/α_0. The variance of the probability for state i is given by $\alpha_i(\alpha_0 - \alpha_i)/(\alpha_0^2(\alpha_0 + 1))$. Thus the variance increases with lower α_i. Note that these are the same calculations as for the beta distribution, and in fact beta distributions are a special case of a Dirichlet distribution with a two-dimensional parameter vector $\alpha = \{\alpha_1, \alpha_2\}$.

3.2 Markov Chain Monte Carlo

Computing a complex probability distribution often proves to be computational infeasible, for instance if it involves integrals that cannot be solved analytically but have to be calculated numerically. However, in most cases it is sufficient to approximate the target distribution instead of computing it exactly. There are many approaches to the approximation of probability distributions. We will present a Markov chain Monte Carlo technique in the following.

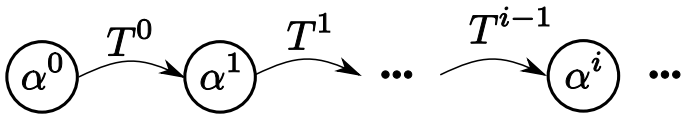


Figure 3.2: Graphical illustration of a Markov chain. The states α^i depend only on their direct predecessor and the transition probability function T^{i-1}: $P(\alpha^i) = \sum_{\alpha^{i-1}} P(\alpha^{i-1})T(\alpha^{i-1}, \alpha^i)$

Markov chain Monte Carlo methods approximate a target distribution by constructing a Markov chain and subsequently sampling new states for the next chain link with a Monte Carlo transition function. The transition function leads to the target rule, i.e. has the target distribution as equilibrium distribution.

In a Markov chain, every chain link represents one state of all the variables. The chain is assumed to obey the Markov property, so the assignments in every step are dependent on the assignments of the last step, yet independent of everything before that. Hence the new state α^i is selected according to a probabilistic transition function $T(\alpha^{i-1}, \alpha^i)$ that depends only on the previous state. This assumption simplifies the algorithm considerably. Figure 3.2 illustrates a Markov Chain.

To converge to the target distribution, when sampling the next state of the variables, the transition function needs to prefer states that are more probable given the model. There are several Monte Carlo methods to do this. Commonly used algorithms for such a setting are Metropolis-Hastings and its special case Gibbs Sampling. Metropolis-Hastings updates the states of the variables sequentially. Each time, the quality of the new state is tested with respect to its probability. If the quality is unchanged or increased, the new state is accepted immediately. Otherwise it is accepted only with a probability that decreases exponentially with the decrease of quality. If rejected, the new state is resampled from the set of states. This is very efficient for systems with few states and when many states are either rejected or a step towards the target distribution. There is a drawback for situations where many of the new states are neither bad nor good. In this case variables might go through many of these states, until one is selected that is advantageous. In our setting this can slow convergence, if a 3D scan or a segment thereof contains few of the many object classes in the corpus. The problem occurs, for instance, with a variable of which the state corresponds to neither of the class labels in the belonging scan. It will be switched to any of the other labels randomly, as none will be worse than the current one.

In Gibbs sampling this procedure is changed. The new state for each individual variable is sampled successively. However, the probability distribution over the possible new states (i.e. the object class labels) of the variable is calculated, conditioned on the current state of the other variables. The new state is then sampled from this discrete distribution. Therefore the new state of a variable is biased towards a better (i.e. more probable given the last state) assignment,

given the current state. As described above, this is advantageous in settings with many states that are equally good than the previous state. The drawback here is the expense of precomputing the probability of all possible new states and thus the conditional probability distribution should be efficiently computable . In settings where the state of a variable is strongly dependent on the other variables, the difference between the computed selection probabilities and the uniform sampling-with-rejection in Metropolis-Hastings may pay off. In our case the strong interdependencies of the variables are due to the Dirichlet priors preferring fewer object classes within one scan segment and equal class labels for equal data items (see Section 3.3).

In order that the chain converges to the target distribution, two requirements have to be met by the transition function (see [Bishop, 2006]):

- The target distribution needs to be invariant under the transition function
- The proposal distribution needs to be ergodic, i.e. every possible state can be reached in a finite number of steps.

The former is an obvious requirement, since the chain cannot converge to the target distribution if it changes to some other distribution upon arrival. In Gibbs sampling, the transition function for a state α, consisting of random variables $\alpha_1, \alpha_2, ..., \alpha_n$, is $p(\alpha_i|\alpha_{-i})$. Here α_{-i} denotes the variables in α except for α_i. This is always an invariant of the joined target distribution $p(\alpha)$ since $p(\alpha_{-i})$ is unchanged and α_i is drawn conditioned only on α_{-i}, such that $p(\alpha_i|\alpha_{-i})$ is also unchanged. Since these together form the joined distribution of the states, i.e. $p(\alpha) = p(\alpha_i|\alpha_{-i})p(\alpha_{-i})$, it is also invariant.

For ergodicity it is sufficient that the transition probabilities are larger than zero for every state change possible.

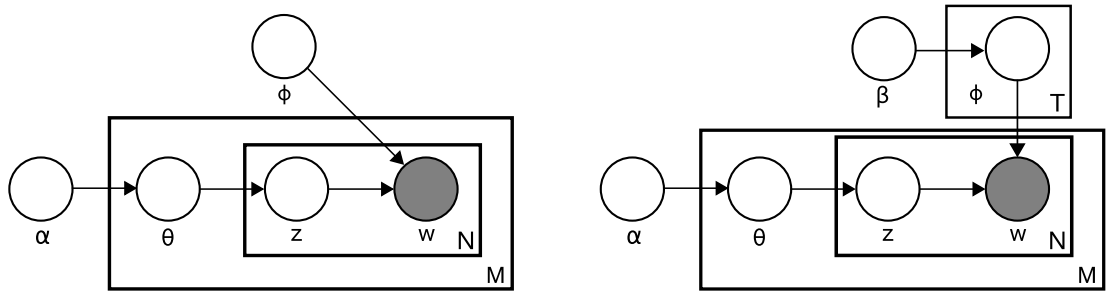
An important fact to keep in mind is the dependency of the samples on their predecessor. Therefore, to get independent samples, a sufficiently large number of sampling steps should be performed (and their results discarded) between two samples, to consider them independent.

3.3 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a fully generative probabilistic model to semantic clustering of discrete data. It was introduced by Blei *et al.* [2003] and has been applied to finding topics in the text domain, e.g. [Griffiths and Steyvers, 2004], as well as for image data [Wang and Grimson, 2007], [Fritz and Schiele, 2008], [Philbin *et al.*, 2008]. In the following description of LDA, we will adopt the terminology of the text domain, even though the algorithm is in no way restricted to text data and will be used to cluster features extracted from 3D range scans. For easier readability, the mathematical notation used will not be explained in detail until after a rough description of the model.

In LDA, the input data is assumed to be organized in a number of sets of discrete data. The individual sets are referred to as "document". All sets together form the "corpus". The data items in the documents are "words". LDA assumes the documents to be "bags of words", i.e. the order will be disregarded. Instead of relying on a distance measure for generating clusters, semantic clustering relies on the co-occurrence of the words in documents to assign them to certain "topics" (i.e. clusters).

The original LDA model of document generation, as proposed in [Blei *et al.*, 2003], is shown in Figure 3.3a. See [Wikipedia, 2008] for a concise explanation of the plate notation used.



(a) LDA model as described in (Blei *et al.*, 2003)

(b) LDA model as described in (Griffiths and Steyvers, 2004)

Figure 3.3: Graphical representation of the LDA document generation process in plate notation. The outer plate represents the M documents of the corpus, the inner plate represents the N words in each document. α denotes a Dirichlet distribution, θ is the multinomial distribution over the topics that is drawn for each of the N documents. z is the topic drawn from θ for every word. ϕ contains the probabilities $P(w|z)$ for every word and topic. β is the Dirichlet prior to this distribution.

Being a generative probabilistic model, the basic assumption made in LDA is that documents are generated by random processes. Each of these represents a different topic j. A random process generates the words in the document by sampling them from its own specific discrete probability distribution over the words $\phi^{(j)}$. A document can be created by one or more topics, each having associated a distinct probability distribution over the words.

To represent the mixture of topics in a document, a multinomial distribution θ is used. For each word in the document, the generating topic is selected by sampling from θ. The topic mixture θ itself is drawn from a Dirichlet distribution, once for every document in the corpus. The Dirichlet represents the prior belief about the topic mixtures that occur in the corpus, i.e. whether the documents are generated by single topics or from a mixture of many topics and which topics prevail.

The model from Blei *et al.* [2003] is modified by Griffiths and Steyvers [2004], by specifying a Dirichlet prior β on the conditional distribution ϕ. The effect of this prior could be a preference for certain words; but we also restrict the prior, such that it only affects whether only few words are included in a topic or as much as possible. Figure 3.3b shows this addition in the graphical representation.

The following detailed description of how we compute the assignment of topics to the input data makes use of a rather complex notation. We summarize the notation in Table 3.1.

d	The document index, $d = [1, D]$, D is the number of documents
w_i	One of N occurrences of the W unique words, $i = [1, N]$. Belongs to a document d
w	A unique word from the vocabulary. $w = [1, W]$
j	The topic index, $j = [1, T]$, $z_i = j$
z_i	The topic that generates w_i. One of T topics.
$\mathbf{w}$	The corpus. $\mathbf{w} = \{w_1, w_2, ...w_N\}$
$\mathbf{z}$	The topic assignment vector. Contains the generating topic for each word occurrence. $\mathbf{z} = \{z_1, z_2, ...z_N\}$
$n_j^{(w)}$	The number of occurrences of word w that have been assigned to topic j by $\mathbf{z}$.
α	A T-dimensional vector used as parameter for the Dirichlet prior of the topic distributions. For our purpose restricted to $\alpha_i = \alpha_j$ for all i, j
$\theta^{(d)}$	A multinomial distribution over the topics. Drawn independently for each document d from Dirichlet(α)
θ	The set of all θ^d, $\theta = \{\theta^1, ..., \theta^D\}$
β	A W-dimensional vector used as parameter for the Dirichlet prior of the word distributions. For our purpose restricted to $\beta_i = \beta_j$ for all i, j
$\phi^{(j)}$	A multinomial distribution over the words. Drawn independently for each topic j from Dirichlet(β)
$\phi_w^{(j)}$	The probability of topic j to generate an occurrence of w: $P(w\|z = j)$
ϕ	The set of all ϕ^d, $\phi = \{\phi^1, ..., \phi^T\}$

Table 3.1: Notation

To put LDA to use, we want to find the assignments of topics to words that are probable (ideally the most probable) under this model of document generation. Given a corpus $\mathbf{w} = \{w_1, w_2, ...w_n\}$, where each word occurrence w_i belongs to some document d, we are looking for the most probable topic assignment vector $\mathbf{z}$ for our data $\mathbf{w}$. Hence, we need to know $P(\mathbf{z}|\mathbf{w})$. From this distribution we

would like to get a topic assignment, which reflects the underlying objects in the digitized scenes and therefore lets us distinguish the points in the point cloud by means of the object class they belong to. Using Bayes' rule we know

$$P(\mathbf{z}|\mathbf{w}) = \frac{\mathbf{P}(\mathbf{w}|\mathbf{z})\mathbf{P}(\mathbf{z})}{\mathbf{P}(\mathbf{w})}. \tag{3.2}$$

Unfortunately the partition function $P(\mathbf{w})$ is not known and cannot be computed directly because it involves T^N terms, where T is the number of topics and N is the number of word instances in the corpus. A common approach to approximation of a probability distribution, of which the partition function is unknown, is the use of Markov chain Monte Carlo sampling methods. For LDA, this approach is introduced in [Griffiths, 2004] as described in the following. For the reasons stated in Section 3.2, in our setting it is advantageous to use Gibbs sampling to transition between the states. We therefore sample from the distribution in the numerator on the right hand side of Equation 3.2, by successively sampling the topic assignment for each word occurrence in the corpus, with the distribution conditioned on the topics of all other words. The distribution over the topics when sampling the topic z_i for word occurrence w_i is given by

$$P(z_i = j|\mathbf{z_{-i}}, \mathbf{w}) = \frac{\overbrace{P(w_i|z_i = j, \mathbf{z}_{-i}, \mathbf{w}_{-i})}^{\text{likelihood of } w_i}\overbrace{P(z_i = j|\mathbf{z}_{-i})}^{\text{prior of } z_i}}{\sum_{j+1}^{T} P(w_i|z_i = j, \mathbf{z}_{-i}, \mathbf{w}_{-i})P(z_i|\mathbf{z}_{-i})} \ . \tag{3.3}$$

We can express these conditional distributions without knowledge of the hidden variables ϕ and θ by integrating over them. The likelihood term on the left side of the numerator depends on the probability of the word distribution of topic j, so we need to integrate over all these distributions $\phi^{(j)}$, i.e.

$$P(w_i|z_i = j, \mathbf{z}_{-i}, \mathbf{w}_{-i}) = \int \underbrace{P(w_i|z_i = j, \phi^{(j)})}_{\phi^{(j)}_{w_i}} \underbrace{P(\phi^{(j)}|\mathbf{z}_{-i}, \mathbf{w}_{-i})}_{\text{posterior of } \phi^{(j)}} d\phi^{(j)} \ .$$

We have chosen a Dirichlet prior, that is conjugate to the family of multinomials, which is the family $\phi^{(j)}$ belongs to. Therefore the posterior distribution of $\phi^{(j)}$ can be easily computed from the prior and the observations by adding the observations to the respective elements of the parameter vector β of the prior Dirichlet distribution. See Section 3.1 for a detailed description. We therefore have a Dirichlet posterior with parameter vector $\beta + n^w_{-i,j}$ where the elements of $n^w_{-i,j}$ are the number of occurrences of each word w, assigned to topic j by the assignment vector $\mathbf{z}_{-i}$. Again the subscript $-i$ denotes that the word occurrence w_i, of which the topic is sampled in the current step, is not counted. Since the terms we integrate over are just one part of the multinomial, weighted by

the posterior of the multinomial, this boils down to a calculation of the expectation of $\phi_{w_i}^{(j)}$. From the properties of Dirichlet distributions we know that the expectation of the ith element of the multinomial variable X_i is

$$E(X_i) = \frac{\alpha_i}{\sum_i \alpha_i} \ .$$

Thus having an occurrence w_i of word w, the integral can be computed by

$$P(w_i|z_i = j, \mathbf{z}_{-i}, \mathbf{w}_{-i}) = E(\phi_{w_i}^{(j)}) = \frac{n_{-i,j}^{(w)} + \beta_w}{n_{-i,j}^{(\cdot)} + W\beta_w} \ . \tag{3.4}$$

Where $n_{-i,j}^{(\cdot)}$ denotes how many words have been assigned to topic j in total. As mentioned earlier, we assume all elements of β to be equal, thus the normalization in the denominator can be written $\sum_w n_{-i,j}^{(w)} + \beta_w = n_{-i,j}^{(\cdot)} + W\beta_w$.

In the same way we integrate over the multinomial distributions over topics θ, to find the prior of z_i from Equation 3.3. Sampling the topic for word occurrence w_i belonging to document d_i, we can write

$$P(z_i = j, \mathbf{z}_{-i}) = \int \underbrace{P(z_i = j|\theta^{(d_i)})}_{\theta_j^{(d_i)}} \underbrace{P(\theta^{(d_i)}|\mathbf{z}_{-i}}_{\text{posterior of } \theta^{(d_i)}})d\theta^{(d_i)} \ .$$

Again this is the expected value of $\theta_j^{(d_i)}$ that—due to the choice of the conjugate Dirichlet distribution as prior—can be calculated by adding the observation vector $n_{-i}^{(d_i)}$ to the parameter vector of the prior α and dividing the element from the resulting vector of interest by the sum of all of its elements,

$$P(z_i = j|\mathbf{z}_{-i}) = \frac{n_{-i,j}^{(d_i)} + \alpha_j}{n_{-i,\cdot}^{(d_i)} + T\alpha_j} \ . \tag{3.5}$$

Here $n_{-i,j}^{(d_i)}$ is the number of words in document d_i, that are assigned to topic j. $n_{-i,\cdot}^{(d_i)}$ is the number of words in the document excluding w_i. Putting the results from equations 3.4 and 3.5 into Equation 3.3, we find the proposal distribution for the sampling of z_i in the form of

$$P(z_i = j|\mathbf{z_{-i}}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w)} + \beta_w}{n_{-i,j}^{(\cdot)} + W\beta_w} \frac{n_{-i,j}^{(d_i)} + \alpha_j}{n_{-i,\cdot}^{(d_i)} + T\alpha_j} \ .$$

After a random initialization of the Markov chain, a new state is generated by drawing the topic for each word successively from the proposal distribution. As mentioned in Section 3.2 the chain states are not independent from their predecessor, so when recording samples to estimate the posterior distribution

we use a subset of the states with sufficient intermediate steps to consider them independent. From these samples the values of θ and ϕ can be estimated using the sampled topic assignments in $\mathbf{z}$ by

$$\phi_j^{(w)} \sim \frac{n_j^{(w)} + \beta}{n_j^{(\cdot)} + W\beta} \ ,$$

$$\theta_j^{(d)} \sim \frac{n_j^{(d)} + \beta}{n_\cdot^{(d)} + T\beta} \ .$$

These values can be used to predict the topic assignments for unseen data.

3.4 Clustering

Another way to subdivide a data set into groups is clustering on a distance measurement. Which can also be an approach to semantic clustering, if the distance measurement encodes semantic relationship. However, in this thesis we will primarily use spatial clustering on the euclidean distance of the scan points. Thus there is no way to tell whether two clusters apart are similar or not. There are several different clustering methods, of which k-means clustering and hierarchical clustering are very popular. In k-means and variations like k-medians, k cluster ids are assigned to the data (3D scan points in our case). This is done randomly for initialization. Then an iterative algorithm such as expectation maximization (EM) algorithm is used to find a better partitioning in each iteration. For EM, one iteration consists of the following steps:

1. Calculate the center (mean, median, ...) of each cluster.
2. For each point calculate the closest cluster center.
3. Reassign each point to the closest cluster center.

This is repeated until no reassignments occur anymore. So in a setting with two scanned objects, the scan points of each are randomly assigned one of two cluster ids in the initial assignment. In the following iterations the cluster centers will be shifted towards the object that had more assignments of the respective cluster id. However if one object is considerably larger, e.g. a man and a dog, the feet of the man might be nearer to the center of the dog, than to the center of the man. Thus, the feet would be assigned the same cluster id as the dog, shifting the center of the man's cluster further up. Another problem arises in a setting with a three objects if one is further apart than the others. In this case if the initial assignment is such that the distant object is the center of two cluster centers, both remain in that object, as the points in the other objects are all assigned to the third cluster.

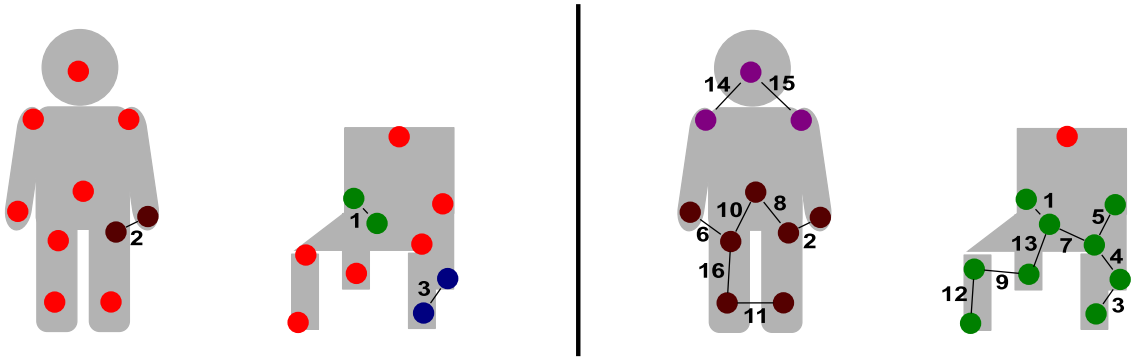


Figure 3.4: Hierarchical clustering on euclidean distance between laser measurements after 3 iterations (left) and 13 iterations (right)

Hierarchical clustering on the other hand does not rely on a randomized initial assignment. Instead the following deterministic iterative procedure is applied:

1. Calculate a distance matrix, containing the distance between each data pair.
2. Group the two data items with the least distance.
3. Consider this group as one item for the next iteration.

Due to the last step the distance matrix changes with respect to the distances to the newly grouped data items. There are several variations in the way distances are calculated for groups. Having two groups, their distance can be defined by

- The distance of their centers (mean, median, ...).
- Their minimum or maximum distance.
- The average (mean, median, ...) of all pairwise distances.

Under the assumption that objects are apart from each other and do not contain gaps, usage of the minimum distance between groups is a good choice for objectwise grouping. In such a setting, the longest distances remaining will be those between the objects.

3.5 Principal Components Analysis

The principal component analysis (PCA) is a vector space transform. For a given data set, a transformation to a new orthogonal coordinate system is computed. The special properties of this coordinate system are, that the origin is located at the center of mass and the axes are ordered by the variance of the data set. I.e. the direction of the first axis is the direction of the highest variance within the

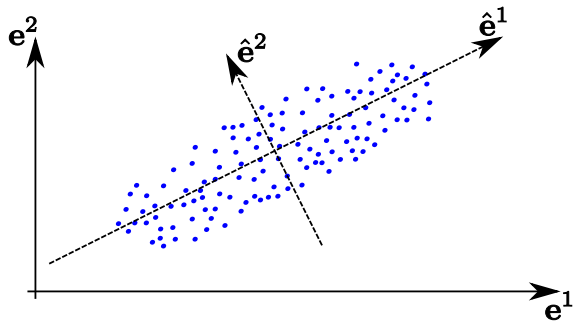


Figure 3.5: Principal component analysis on a two dimensional dataset. The eigenvectors $\hat{\mathbf{e}}^1$ and $\hat{\mathbf{e}}^2$ form a new basis aligned with the variance of the data set.

data set, the direction of the second axis is the direction of the highest variance orthogonal to the first axis. This is accomplished by first translating the mean of the data set to the origin by subtracting the mean vector from all data points. Assuming each point to be on of N column vectors $\mathbf{x}_i$ the mean vector $\bar{\mathbf{x}}$ can be computed by

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \ .$$

with translated vectors $\hat{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}_i$. With vectors in $\mathbb{R}^n$, where n is the dimensionality of $\mathbf{x}_i$, we determine the $n \times n$ covariance matrix $\mathbf{C}$ of the data set:

$$\mathbf{C} = \mathbb{E}\left[\hat{\mathbf{x}}_i\hat{\mathbf{x}}_i^T\right] = \frac{1}{N}\sum_{i=1}^{N}\hat{\mathbf{x}}_i\hat{\mathbf{x}}_i^T \ .$$

Since each $\hat{\mathbf{x}}_i\hat{\mathbf{x}}_i^T$ is symmetric, $\mathbf{C}$ is also symmetric. Therefore we can compute the eigenvalues and corresponding eigenvectors of the covariance matrix by finding $\mathbf{V}$ that diagonalizes $\mathbf{C}$

$$\mathbf{D} = \mathbf{V}^{-1}\mathbf{C}\mathbf{V} \ ,$$

where $\mathbf{D}$ is a matrix with non-zero entries only on the main diagonal and $\mathbf{V}$ is an orthogonal matrix. The element d_{jj} on the diagonal of $\mathbf{D}$ is the eigenvalue to the eigenvector in the jth column of $\mathbf{C}$. When used as basis vectors for a coordinate system as described above, the eigenvectors have to be normalized and ordered by the magnitude of their eigenvalues. For two dimensional data the result is shown in Figure 3.5.

3.6 Spin Images

In recent years, sensor systems to capture 3D range data have become more and more common in research as well as in the industry. Along comes the

need to describe and distinguish free-form 3D shapes for tasks such as surface registration, object alignment and of course recognition and classification. Some feature proposals are briefly described in Section 2.1. A good overview of proposals can be obtained from [Campbell and Flynn, 2001].

Spin images are a data level representation of surface patches described in [Johnson, 1997] that have been used successfully for object recognition by surface matching ([Johnson and Hebert, 1996], [Johnson and Hebert, 1998], [Johnson and Hebert, 1999]). The generation of a spin image for a given point on a 3D surface can be thought of as aligning the side of a raster image to the surface normal of that point, with the "up" direction of the raster image being equal to the orientation of the normal. Then the raster image is rotated around the surface normal, "collecting" every point of the surface in the raster element that intersects with during this rotation. This process is shown in Figure 3.6. Spin images give a usable representation of the surface property. Due to the alignment with the surface normal, spin images are translation- and orientation-invariant. Full rotational invariance comes through the "spinning" around the normal which can be seen as an integration over the remaining degree of freedom.

To accommodate for differences in the density of the data we get a canonical representation of the spin image by dividing the value of every raster element by the maximum value in the raster. This value is then discretized.

Spin images are adaptable in precision by choosing the *raster resolution* (how many raster elements) and a suitable *discretization* of the stored values. By the choice of the *support distance* parallel and orthogonal to the surface normal, the size of the spin images can be controlled, to explicitly represent local or global properties of the objects.

A drawback is that the uniqueness of representation depends on the accuracy of the surface normal. Unfortunately measurement noise greatly affects the direct calculation of surface normals from the nearest neighbors only, such that smoothing techniques become necessary.

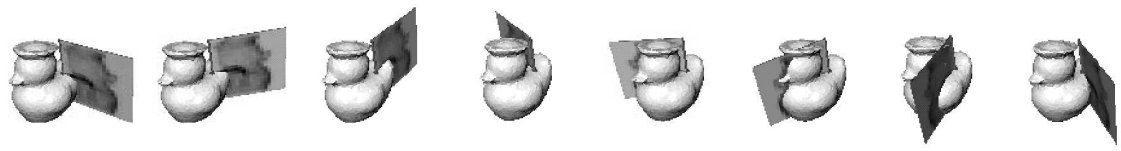


Figure 3.6: Eight steps in the generation of a spin image on the model of a rubber duck. Image taken from (Johnson, 1997)

4 Methodology

This chapter describes our approach to clustering and classification of objects from 3D data using the techniques described in the previous chapter. In the following we introduce definitions of the problem we face, the task we want to accomplish and the approach we propose. After these definitions, we describe our data preprocessing steps, followed by our approach to feature extraction and clustering.

4.1 Definitions

In this work, we assume the following problem description:

- We are given a set of captured *scenes*—the *corpus*. Each *scene* is in form of a point cloud and contains a set of *objects* plus irrelevant background structure.
- Each *object* is an instance of exactly one *object class*.
- We consider the number of *object classes* contained in the *corpus* to be known.
- For each *scene*, we assume the minimal distance between point measurements within one *object* to be smaller than the minimal distance between points of different *objects*.

Given a corpus where these assumptions hold true, we want to achieve the following task:

- Learning a model for object classes in an unsupervised way.
- Consistent classification of the objects in the corpus.
- Correct classification of unseen objects belonging to one of the known object classes.

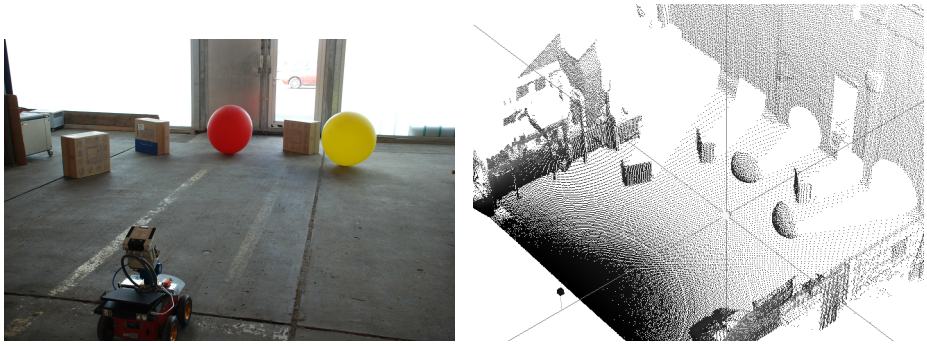
Note that “unsupervised” in this context does not mean, that our approach does not require human interaction. In particular, insights about the parameters for feature generation will be found through the experiments in Chapter 5. However, the parameters determined in this way will be applicable to a wide range of object classes and are not limited to the exact set of objects, we use. Importantly, the model for each object class will be learned without supplying the

learning algorithm with information about the classes, e.g. in form of training examples. Our approach is structured as follows:

- Spatial Segmentation of the scenes into *segments*.
- Usage of discretized feature that describe the data based on shape.
- Use of LDA for learning the class models based on feature histograms.
- Use of LDA for classification of seen and unseen data.

4.2 Data Preprocessing

When recording 3D data with a range scanner, the output is a stream of distance measurements associated the 3D rotation angles — pitch, roll and yaw — and the coordinates of the origin of the laser beam. This data we use to compute a point cloud of 3D coordinates of the endpoints of the beams. The recorded data encompasses the distances of everything within the field of view of the scanner. A sample scene and the resulting point cloud is shown in Figure 4.1



(a) Picture of the Scene **(b)** Point Cloud

Figure 4.1: Example Scene with three boxes and two balloons.

To be able to concentrate on the objects we extract the floor, walls and other objects in the background of the scan.

In preliminary classification experiments, we tried to learn the similarity clusters on the remaining foreground objects. However, the results did not meet our expectations, which we attribute to problems of LDA to learn from documents with mixed topics. A similar problem has been faced in [Wang and Grimson, 2007] for visual images containing different objects and has been solved by segmenting images in various ways. In 3D range data the distinct advantage is, that segmenting can be done on a full Euclidean distance measure, as all

Figure 4.2: Artifacts (highlighted by yellow background) often occur at the edges of an object, where the laser beam hits the object only partially

spatial dimensions of the measured points are known. Therefore another preprocessing step is introduced: clustering the scan with a minimal gap between scanpoints as separation criterion. Note that oversegmentation is of no concern here, since splitting of one object into more than one document in general will not affect the performance of the algorithm, as it should be able to find the similarities in the object parts to other documents with different segmentation. Also for hierarchical clustering, which we will use for comparison, segmentation is required, as this method yields no point-wise labeling.

No efforts have been undertaken to remove artifacts from the scanning process. These generally occur at the edge of objects when only part of the laser beam hits the object while the rest is reflected by something behind the object. If the two measurements are nearby, the scanner interpolates between the two times-of-flight; this results in frayed out edges, as visible in Figure 4.2.

4.3 Feature Extraction

As described in chapter 3.3 we need to extract discrete features from the input data, that reflect the characteristics of the objects classes. In the case of 3D range data the main property to be extracted is a description of the surface shape. Therefore we investigated shape descriptors for applicability to our setting. The requirements were:

- Translational invariance
- Invariant under rotation (at least about the Z Axis axis)
- Unique in representation
- Representation of free-form objects

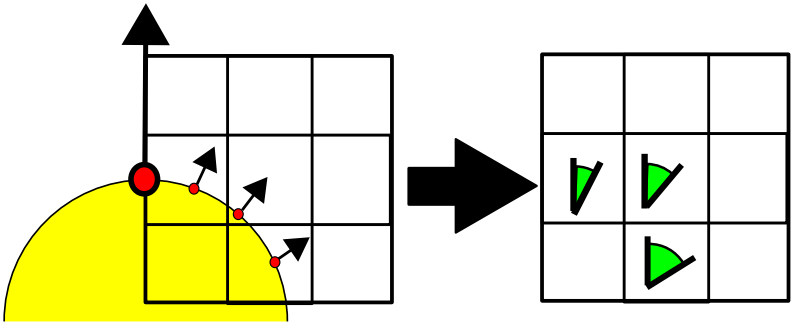


Figure 4.3: Illustration of our proposal for an advanced surface signature: Instead of counting scan points (red) on a surface, the angles (green) of their surface normals (small arrows) relative to the normal of the query point (big point and arrow) are stored.

- Usable with parts of objects
- Discretization is possible

Invariance under rotation about X and Y axes yields the advantage of having the same class for, e.g. standing and lying people, yet makes it impossible to distinguish a wall from floor and ceiling. A unique representation will always be the same for the same underlying non-invariant properties. Features which do not have a unique representation, for instance the 3D shape context in [Frome *et al.*, 2004], require that one representation from the input data is compared against all representations of the feature in the model. Since there is no predefined model in unsupervised learning all representations would be necessary for every feature, thus greatly increasing the size of the feature dictionary.

A feature that fulfills all the above requirements is the spin image described and discussed in Section 3.6.

To further improve the expressiveness of the spin images with respect to the local shape of a point, we introduced changes to the generation process. Instead of counting the points intersecting with the "spinning image" and possibly normalizing the image later, we compute the angle between the normal of the query point and the normals at the collected points. Finally all angles in one bin are averaged over. The process is illustrated in Figure 4.3 for a simplified 2D example.

This proposal follows the lines of [Stein and Medioni, 1992] in that it explicitly uses the orientation of the surface in the neighborhood of a point. In experiments this kind of spin image led to considerably better consistence for the feature histograms from different object classes (see Section 5.2). Experiments with a similar surface signature, based on the up vector instead of the surface normal of the query point, did not meet our expectations.

As mentioned in Section 3.6 the calculation of the surface normal is prone to measurement noise and slight imprecisions in the scanner pose. Computing

the surface normal of a point using its nearest neighbors only, therefore proved prohibitive. There are many different ways to approach this problem. The most common approach is to apply the principal component analysis as described in Section 3.5, by using the eigenvector in direction of the lowest variance. Since we cannot measure surfaces with a normal oriented away from the sensor, we always know the correct orientation. We therefore applied the principal component analysis on a patch of 15 cm radius for normal calculation, which yielded quite stable results, yet introduces undesired smoothing of edges and corners. Experiments with a smaller radius and with a mesh based calculation showed worse results because of the inaccuracies in the sensor data, especially for sparse data, distant from the sensor.

4.4 Clustering

After transforming the raw input data to a discrete feature space, we want to cluster the scan segments created in the preprocessing step on basis of their feature histograms. Inspired by previous work we want to use Latent Dirichlet allocation, as described in Section 3.3 for the unsupervised discovery of object classes. Latent Dirichlet Allocation is a probabilistic framework. It has been successfully used for the semantic clustering of distributions of discrete data in the text domain and for appearance based clustering in visual data. We will apply it to cluster the distributions of surface features. As mentioned in Section 3.3, we need to supply the values for the hyperparameters α and β. The hyperparameters define the prior distributions for the mixture of object classes in a data set, and the mixture of features in an object class respectively. In the next chapter we experimentally determine the optimal range for both hyperparameters.

For comparison of our classification results, we also implement a clustering method based on hierarchical clustering. Hierarchical clustering is a spatial segmentation approach, thus we need to compute a distance measure between the feature distributions. To compare two documents d_1 and d_2 we normalize their histograms of feature occurrences $\mathbf{f_d} = \{\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_V}\}$ such that each sums up to one. We then compute the histogram intersection, as described in [Hetzel *et al.*, 2001], by summing for each feature the smaller occurrence count. Hence, if two feature distributions do not have any features in common, the smaller value of two corresponding histogram bins is always zero, and thus the sum is zero. In contrast, for similar distributions the minimum of corresponding bins will be high for shared features, resulting in a intersection close to one. The calculation is illustrated in Figure 4.4.

We also considered the χ^2 and Kullback Leibler divergences as distance measures. Both would need smoothing for features occurring only in one of the segments, as probabilities of zero cannot be handled. Furthermore, both are heavily influenced by features with few occurrences of which there are many in

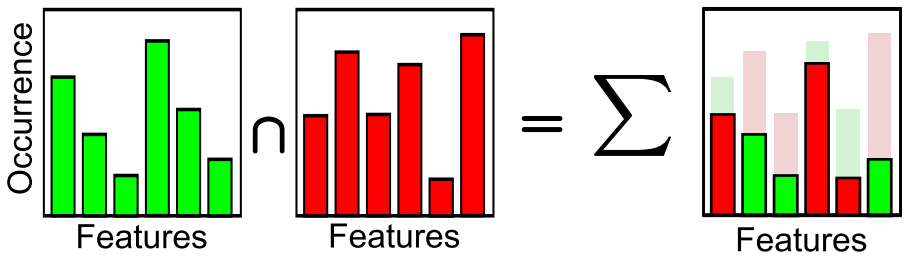


Figure 4.4: The intersection of two histograms is defined as the sum of the smaller elements

our data. This led to considerably worse results with respect to the distances between distributions that represent the same object class. As mentioned before, there are several methods to determine the distance of two clusters, which we address in Chapter 5.

LDA does not need an explicit distance measure, as the feature occurrences are treated as observations of random variables. LDA is not affected by features with few occurrences and will most likely assign the predominant label in the document they belong to.

5 Experiments

In this chapter we describe our experiments of object classification, using the features described in Section 3.6 and Section 4.3. For semantic clustering, we use latent Dirichlet allocation and hierarchical clustering as described in Chapters 3 and 4. The goals of our experiments are

- to find a range of parameter settings, that achieve optimal classification results and
- to demonstrate the capabilities of our approach.

In the following section we describe the data we use for our experiments, including the capturing process and the objects we want to classify. With these objects, several scenes are set up and scanned. The resulting point clouds are then preprocessed and the features are extracted. For feature extraction we need to determine a reasonable subset of the parameter space. This is described in detail in Section 5.2. Section 5.3 describes the classification experiments with both clustering methods. The focus will be on the exploration of the determined parameter range. The results will be discussed and compared in Section 5.4.

5.1 Input Data

We use two sets of 3D laser range scans that were recorded with the robot shown in Figure 5.1. The laser range scanner unit is a SICK LMS 291-S05 with an angular resolution of 0.25 degrees and a standard deviation of 10 mm.

Figure 5.1: The robot used to capture the range data

Figure 5.2: Objects of corpus one: Balloon and box

The unit is mounted onto a Amtec PowerCube hinge on a Pioneer 2-AT robot base. To capture a 3D scene the hinge changes the pitch of the scanner unit, such that subsequent range scans cover the whole scene. The angular resolution for the pitch is about 0.33 degrees. A sample scene and the resulting point cloud is shown in Figure 4.1

All scans have been preprocessed as described in 4.2, such that the later feature extraction step takes place on segments of the scans, containing only individual objects.

5.1.1 Corpus One - Simple Object Classes

For the first set of scenes two object classes were scanned: cardboard boxes of size 30 cm x 60 cm x 60 cm and balloons with a diameter of about 30 cm. Figure 5.2 shows an example of each class. These two primitive object classes were the basis for a first evaluation of our approach. The resulting features for the objects were rather homogeneous. The surface patches on the approximately spherical balloons are almost uniformly shaped, with respect to the position and orientation of the surface normal. The feature distribution is therefore dominated by only few features. The same holds true for the sides of the boxes which dominate the features generated by the box, though the edges and corners give rise to diverse features.

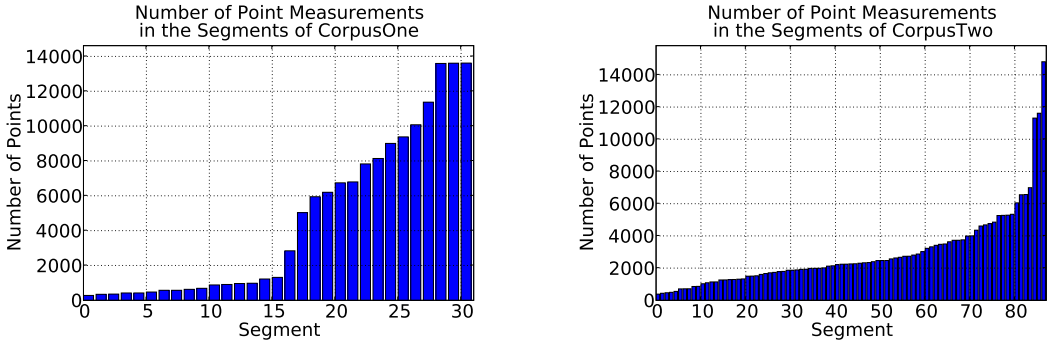
This set of scans consists of twelve scenes containing 31 individual objects. Figures 5.5a and 5.5b exemplify the resulting point clouds. Figure 5.4a shows the number of measurements for each of the individual objects. The huge difference is mainly due to different distances between object and sensor. Differing object size and occlusions also affect the number of scan points.

5.1.2 Corpus Two - Complex Object Classes

For the second set, we chose to include more complex object types than in the previous set and also have more similar classes. So besides ballons and boxes the scenes contain humans and two types of chairs. Example objects are de-

(a) Human, balloon and box **(b)** Chair **(c)** Swivel chair

Figure 5.3: Pictures of the test objects in corpus two



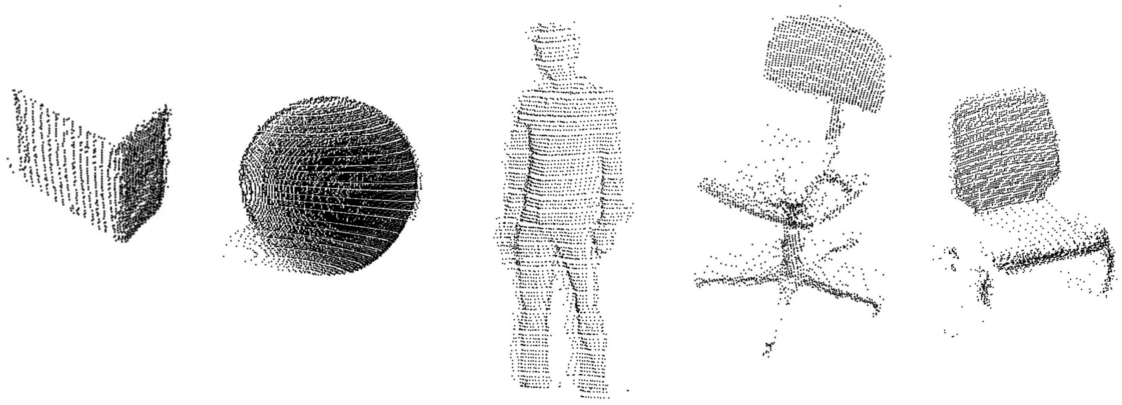
(a) Corpus one **(b)** Corpus two

Figure 5.4: Number of point measurements in the documents after segmentation

picted in Figure 5.3 and some resulting point clouds in Figure 5.5. Visualization of more scan segments can be found in Appendix A; there, the difficulty of the data set is apparent, e.g. in artefacts from noisy measurements. In particular, the legs of the chairs are extremely noisy, due to reflection of the laser beam. Also particular difficulties arise from the self-occlusions, inherent to the viewing perspective. In the two chair classes, the seating area is often occluded. Boxes reduce to a single plane, dependent on the point of view. The features extracted from the scans of humans are very diverse compared to the simple objects. Furthermore the scans differ because they include four different persons in slightly varied stances. The two kinds of chairs and the boxes will provide information on how well we will be able to distinguish and generalize similar classes.

This second set consists of 39 scenes segmented into 82 documents. Figure 5.4b shows the measurement counts of the documents.

This corpus serves to find limitations of clustering based on latent Dirichlet allocation. While we still use relatively simple scenes to allow for the spatial



(a) Box **(b)** Balloon **(c)** Human **(d)** Swivel Chair **(e)** Chair

Figure 5.5: Example point clouds for objects after spatial segmentation

clustering to find a good clustering, the scenes contain very difficult challenges, as mentioned above.

5.2 Feature Distributions

As described in Sections 3.6 and 4.3 we need to determine a set of suitable parameters for the generation of spin images:

- Raster Resolution
- Discretization Resolution
- Support Distance

All these parameters influence how shapes can be distinguished, yet with different ramifications. Raster and discretization resolution mainly affect the precision of discrimination and are therefore important with regard to error robustness and the ability to differentiate small variations in shape. The choice of these parameters depends strongly on the density of the point cloud — and thus on the angular resolution of the scanner and its distance to the objects. As stated before we use a maximum angular resolution of 0.33 degree along the pitch and 0.25 degree along the yaw. The minimum distance (assuming the surface to be orthogonal to the laser beam) of a point to its nearest neighboring scan point can be computed by $d_{ij} = D \cdot tan(\alpha_{ij})$ as shown in Figure 5.6. For objects within a range of 5 m this results in a minimal distance of 2.9 cm vertically

and 2.2 cm horizontally. Therefore a smaller bin size (size of the raster elements) would be prone to contain bogus empty bins that are between two scan points. A slightly larger bin size might still be prone to misrepresentations if the laser beams hit a distant object in a steeper angle. Therefore, the maximum $k \times k$ raster resolution should be chosen to have $k < \frac{\text{support distance}}{D \cdot tan(\alpha)}$.

The problem of choosing a good discretization resolution is different for the two types of spin images. For the standard spin images, the discretization resolution defines how fine-grained the relative density of surface points is captured. Here it is important to have a lower resolution than the typical maximum number of points collected in a bin. Otherwise distant objects will not be represented correctly as the points are sparse and their numbers not representative anymore.

For the point descriptor we proposed in Section 4.3, resolution denotes the how we subdivide the relative angle between the surface normals. So for a resolution of two discrete values, we would only be able to tell whether it is lower or higher than 90 degrees. For reasonable differentiation the value should be higher than that. If chosen too high, the exact position of the measurements will influence the result. However, this effect is not as problematic as with the standard spin images and we would like to retain a degree of imprecision, as it allows for slight variations in shape.

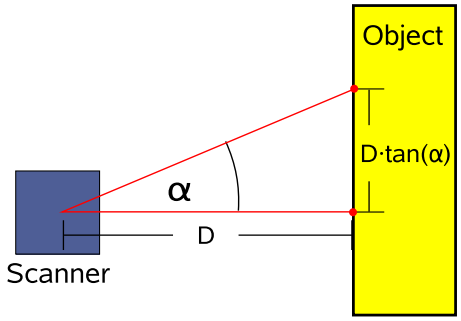


Figure 5.6: Computing the distance between neighboring scanpoints given the distance to the laser scanner

The support distance, i.e. the size of the spinning image, is important to capture the context of a shape. If it is chosen too small, the information captured by the spin image will only include the most nearby changes and most spin images will look similar, i.e. flat, since the variation of a shape is usually limited. If chosen too large, local changes in the object will affect the spin images in a large area which might be problematic, especially if there is occlusion or clutter in the scene. Also, local features will be more robust when trying to discriminate the class “human” and all the spin images are affected by the exact position of the arms.

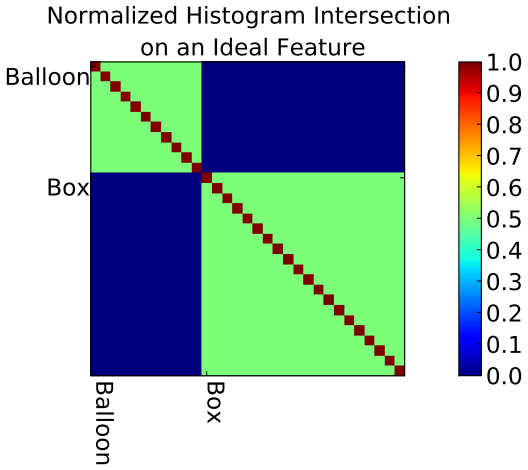
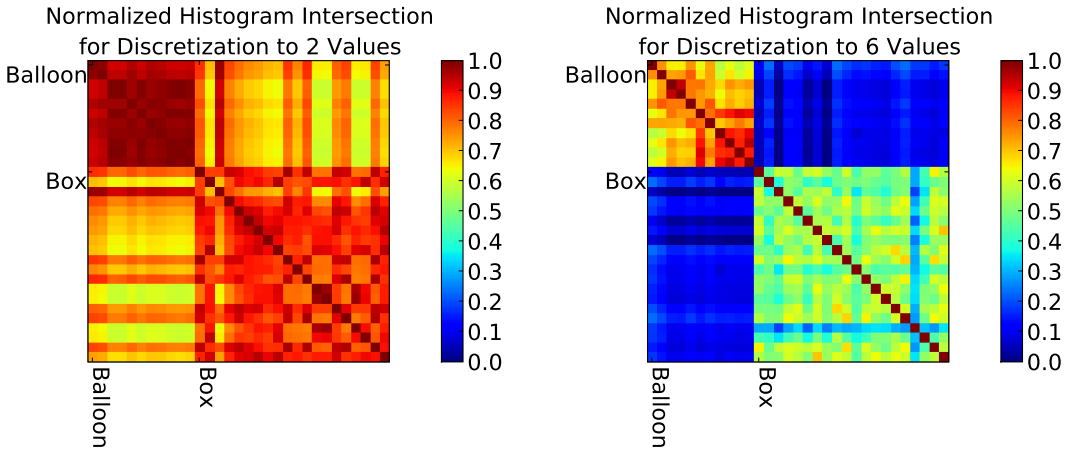


Figure 5.7: Document similarities under a hypothetical feature that perfectly separates the classes

To evaluate the distinctiveness of features generated with a certain parameter set, we first use the histogram intersection to measure the similarity of the documents. This is of course a good estimation for hierarchical clustering, as it depends on distances between the scan segments, that allow to separate the classes. It is also suitable for LDA, since the histogram intersection concentrates on the amount of words shared between two documents. Upon these quantities the learning algorithm decides whether two documents are of distinct topics or not. Assuming an ideal feature, that can perfectly discriminate between the desired classes, the intersection would result in high similarity for objects belonging to the same class and no similarity for objects of different classes. This ideal discrimination is shown in Figure 5.2. Here, the similarity value obviously allows for separation of the two object classes. Note, that the exact value shown is of no importance, as long as the interclass similarity is always lower than the similarity within the classes.

Figure 5.8 shows two example plots of the similarities between the documents in corpus one for the standard spin images as described in Section 3.6 and our surface representation as in Section 4.3. The plot shows generally high similarity between the documents within one class and less between documents of different classes. However, choosing a higher discretization resolution decreases the intraclass similarity of documents. Lowering it will increase the interclass similarities.

To find a good compromise we calculate a ratio of interclass similarity to intraclass similarity. The intraclass similarity is computed by averaging over each similarity measure in a class. For the interclass similarity we also average over the similarities between the documents of two classes, then choose the result of



(a) Spin images: Raster resolution of 3×3 and discretization to two values

(b) Spin images type 2: Raster resolution of 5×5 and discretization to six values

Figure 5.8: Example document similarities for a support distance of 10 cm

the most similar pair. This proceeding proved to be an important choice, since otherwise the distinction between the most similar classes were not emphasized. However these numbers are of course only a rough guide. They do not reflect that using a high resolution, some scans do hardly resemble any other scan. For this kind of information we need the full documentwise plots.

Figure 5.9 shows these cumulated statistics for the objects in the first corpus at a support distance of 10 cm, raster resolution of 3×3 which are the parameters where the standard spin images performed best. On the left side the results for the intraclass similarity are shown. The diagram shows that the intraclass similarity is much higher for the surface descriptor we proposed. Also the interclass similarity is lower than when using the standard spin images. This relation holds true for all parameter sets we computed.

The second column suggests, that the features generated with the chosen parameters allow a good distinction between the two classes and thus we expect the clustering algorithm to be able to classify accurately. Learning results for the first corpus are presented in Section 5.4. However, a similar quality of distinction is achieved for a wide variety of parameters for this corpus. Therefore our attention lies on the second corpus, where the problem of finding a suitable parameter set is harder.

Hence we did the calculations described above for the second corpus. Sample similarity plots can be seen in Figure 5.10. Particular problems visible are the high similarity between chairs and boxes and the low similarity within the classes “human” and “swivel chair”. If we create the surface signatures with a high level of detail (in terms of raster and discretization resolution) and limited locality, we expect the classes “human” and “swivel chair” to be missclassified

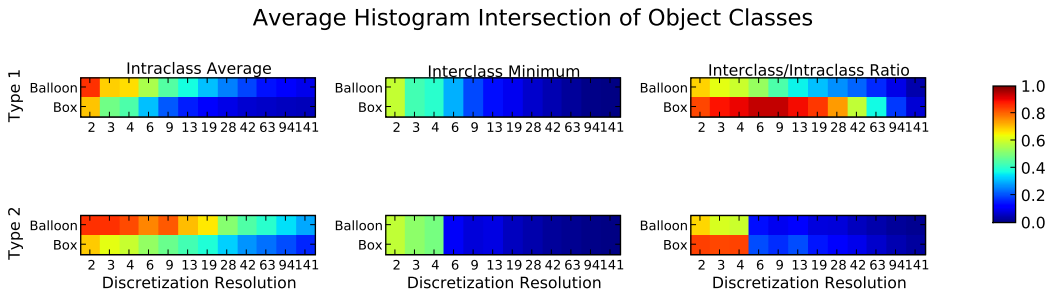
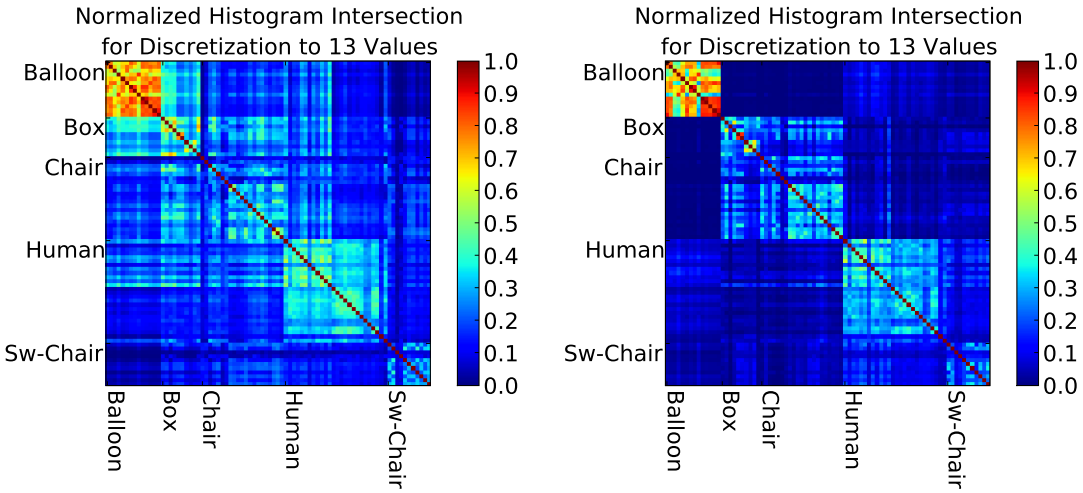


Figure 5.9: Comparison between histogram intersections for standard spin images (top) and our proposal (bottom). On the left side are the intraclass values (higher is better), centered are the interclass values (lower is better). The right figure shows the ratio (lower is better)

often. With very rough, localized features, we expect the documents of the classes “chair” and “box” to become a common class.

For this corpus the choice of good parameters for the surface descriptors is crucial to the success of the learning algorithm. We evaluated a variety of parameters, with respect to the histogram intersection ratio, as described above. We parameterize the support distance in the range of 10 cm to 50 cm. The number of scanpoints for distant objects does not allow a sensible choice of raster resolution for smaller support distance. A higher support distance contradicts the advantages of local features mentioned earlier. We evaluate raster resolutions of 3×3, 4×4, 5×5 and 7×7.

Figures 5.11, 5.12 and 5.13 show the evaluation of the intra-/interclass similarity ratio for this corpus, for different parameter combinations of support distance, raster resolution and discretization resolution. The figures only depict the results for the surface signature we proposed, as the results of standard spin images are consistently worse. From these statistics we can deduce which parameters sets result are promising. Again, on the left side the average similarity of documents of the same class are shown. The plots support our expectations of reduced similarity with higher detail and larger support distance. The center column shows the decrease of similarity for documents of different classes. And as we can see in the right column, the decrease in the intra- and interclass similarity is not proportional. Note that the plot colors only reflect a range from zero to one. If the documents of a class have a higher similarity to those of another class than to other documents of the same class, their ratio is above one and displayed with magenta regardless of the magnitude. This allows to concentrate on the relevant range and is of no disadvantage here. A ratio of one or above, that there can be no distinction between those classes, such that all scan segments are classified correctly. Values below 0.01 are shown in



(a) Standard spin images

(b) Spin images type 2

Figure 5.10: Example document similarities for a support distance of 20 cm, discretization to 13 values, raster resolution of 3×3

black, to emphasize that there is less than one percent correspondence between the documents. As these results are averaged, this is a sign that there is hardly any match for many documents anymore.

From the plots we deduce, that for small support distances up to 10 cm it is extremely difficult to distinguish some objects — in particular chairs and humans — from object of other classes. For chairs we see that the ratio becomes better when the discretization resolution is higher. However the class of humans has a meaningful self-similarity for less precise resolutions. For a larger support distance we obtain a window of resolutions where the ratio goes down considerably. For support distance values of 0.5 m and above the intraclass similarity is very low, such that for many documents a sensible classification is impossible.

In conclusion, the shown subset of the parameter space seems sufficient for a thorough analysis of the capabilities, as we cover the sensible spectrum for locality and precision. As argued above, we deem values outside this spectrum to be problematic. Hence the clustering experiments in the following section only consider the parameter ranges established in this section.

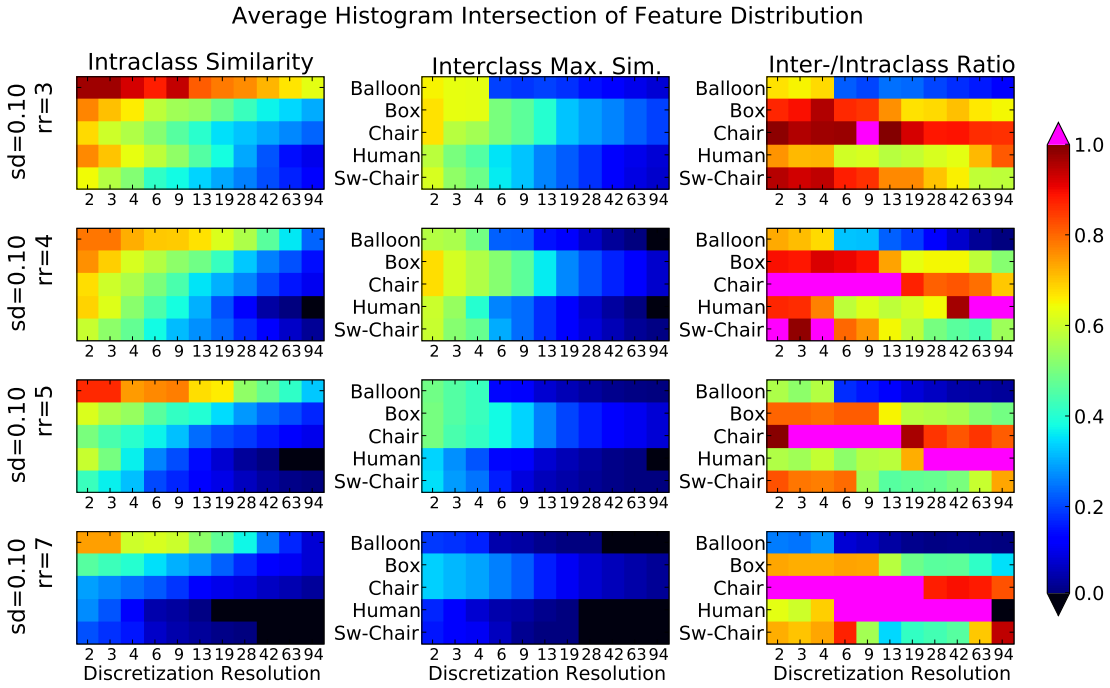


Figure 5.11: Analysis of intra- and interclass similarity for the second corpus. Various parameter sets are shown in rows. Here "sd" stands for "support distance", "rr" stands for "raster resolution"

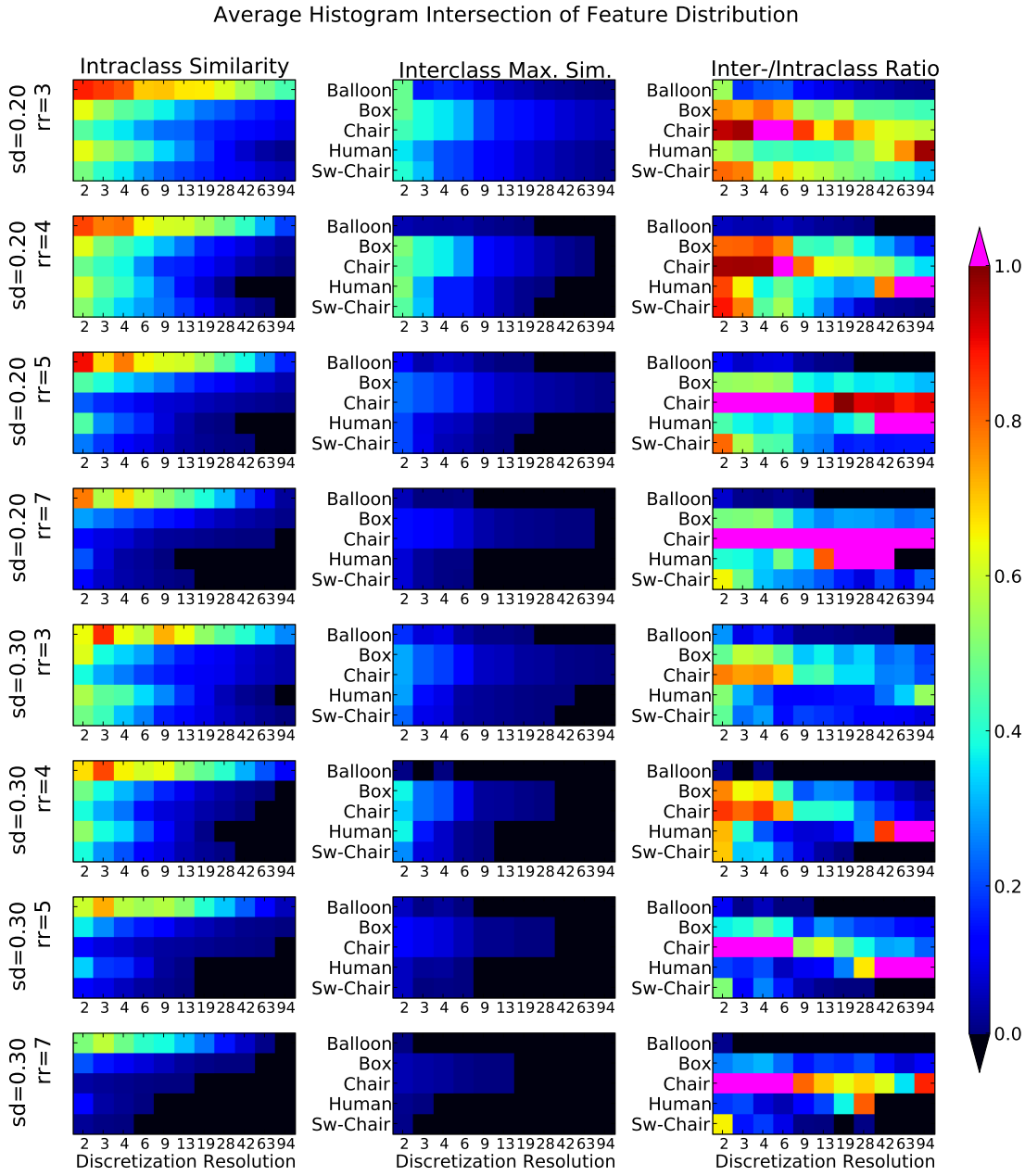


Figure 5.12: Analysis of intra- and interclass similarity for the second corpus. Various parameter sets are shown in rows. Here "sd" stands for "support distance", "rr" stands for "raster resolution". (Continued)

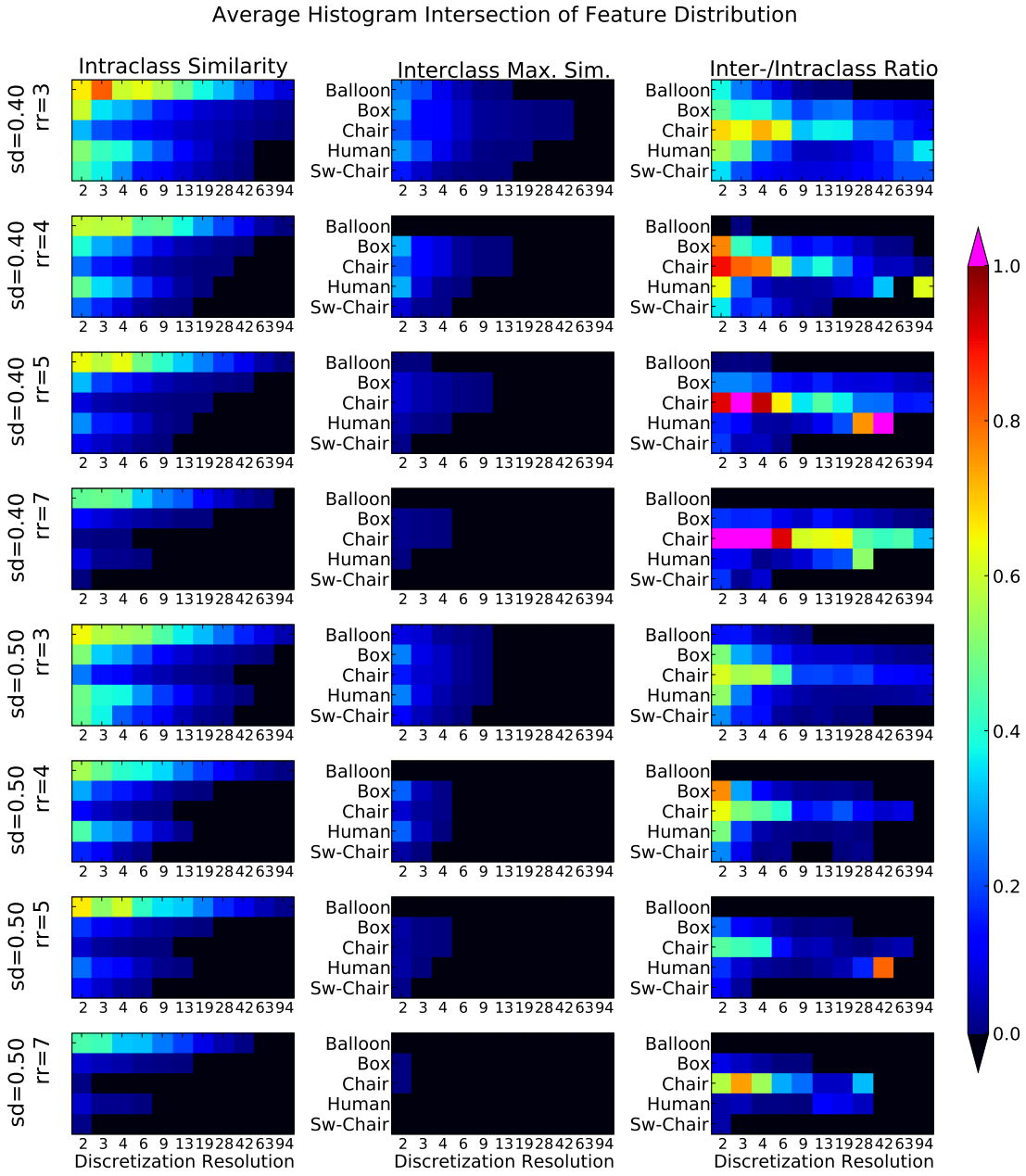


Figure 5.13: Analysis of intra- and interclass similarity for the second corpus. Various parameter sets are shown in rows. Here "sd" stands for "support distance", "rr" stands for "raster resolution". (Continued)

5.3 Clustering

After feature extraction, we classify the feature distributions. First the experiments using hierarchical clustering are presented, followed by the experiments based on latent Dirichlet allocation. We evaluated the clustering results using the intuitions for the feature generation parameters as described above. The goal of these experiments is to find a range of parameter settings with excellent results, rather than finding a particular setting to generate a perfect clustering result. Identifying a suitable range instead of one perfect setting is important, as it demonstrates, that the results generalize to deviations in the setting of the experiments. This includes, for instance, varying complexity of the objects and differences in data capturing.

5.3.1 Hierarchical Clustering

There are two important choices to be made, when using hierarchical clustering: The distance measurement and — based on that — the method to determine the distance for a cluster. The first choice has been described in Section 4.4. For the reasons stated there, we will make use of histogram intersection as distance measurement. For the second choice, however, we will find the best method experimentally. Also we need to find parameters for feature generation, that yield a good separability of the classes when using hierarchical clustering. Therefore, we generated feature distributions for both corpora, using the parameter combinations analyzed in the previous section. For each parameter combination we clustered the scan segments into two categories for the first corpus and five categories for the second corpus. For evaluation we determine which clusters represent which object class and calculate the percentage of correctly assigned segments.

Corpus One

Figure 5.14 shows the results, accumulated over the various experimental settings. The x-axis shows how many scan segments have been correctly classified, the y-axis shows for how many parameter combinations a result was achieved. There are two peaks: The peak at 100 % shows that the scan segments have been perfectly grouped according to whether they contain a balloon or a box. The peak around 60 % is mainly due to parameter sets, where all but one segments are assigned to the first cluster and only the last segment is in the second cluster. Since the first cluster corresponds to the box class, the box segments are counted as correctly classified.

For a thorough analysis we computed the clustering for the parameter combinations and both feature types mentioned in Section 5.2. For hierarchical clustering we use the following linkage types for cluster distances:

- Minimum Pairwise Distance (also “single linkage”).
- Average Pairwise Distance.
- Maximum Pairwise Distance (also “complete linkage”).

As stated before, we want to find a parameter range, that yields optimal results. Here we have many parameter combinations that achieve perfect classifications, i.e. each cluster contains only objects that belong to the same class. Having so many perfect results, we disregard the non-perfect parameter combinations for now and count the remaining results with respect to the parameters. The most influential parameter seems to be the feature type. As Figure 5.15a shows, more than twice as many perfect results were achieved using our feature proposal, than using the standard spin images. In Figure 5.15b the perfect clustering results are shown with respect to the linkage method, where we see that maximum distance linkage is considerably inferior to the other two methods. In consequence we only investigate further on the perfect classification results with feature type two and without complete linkage. The dependencies for this reduced result set are shown in Figure 5.16, where we see that discretization resolutions of six and nine discrete values perform best. The right graph suggests a tendency towards coarse raster resolutions, while the center plot shows that a support size of 20 cm most often produces a perfect classifications. Restricting this approach to features of

- feature type two, with
- 0.2 m support distance and
- discretization of the raster elements to between six and 27 distinct values,

we find the results shown in Figure 5.17. Note that raster resolution has not been restricted, as the results with respect to the given support distance and

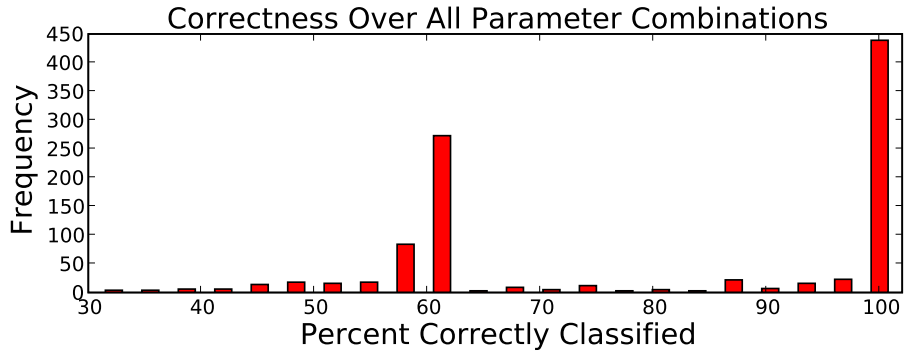
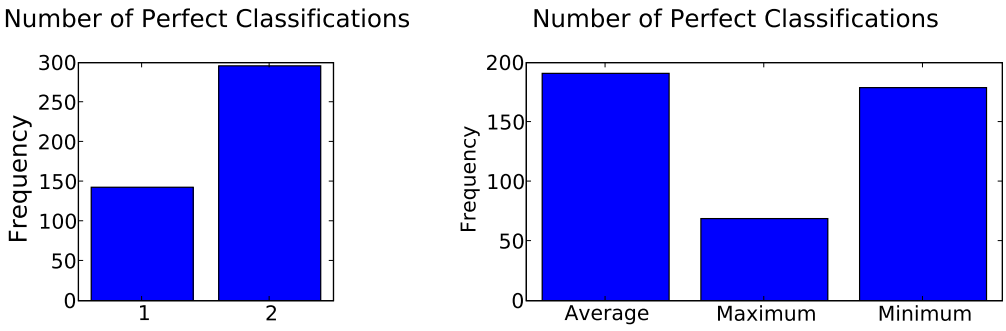
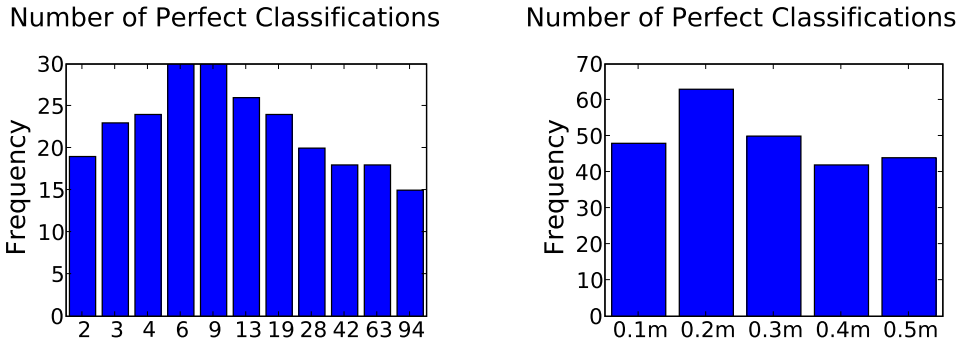


Figure 5.14: Aggregation of the classification results.

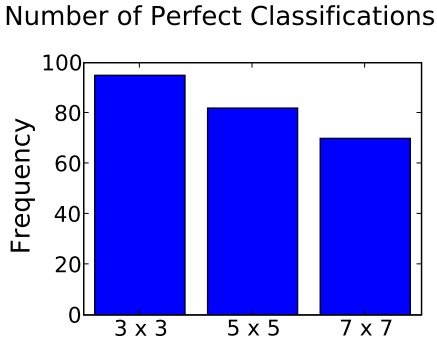


(a) By Feature Type **(b)** By Linkage Method

Figure 5.15: Number of perfect results with respect to different parameters



(a) Discretization Resolution **(b)** Support Distance



(a) Raster Resolution

Figure 5.16: Number of perfect results with respect to different parameters using only feature type two and exclusive the maximum linkage method

discretization have shown no preference anymore. Other parameter subsets could be chosen where raster resolution plays a role. Overall the clustering performance is very good.

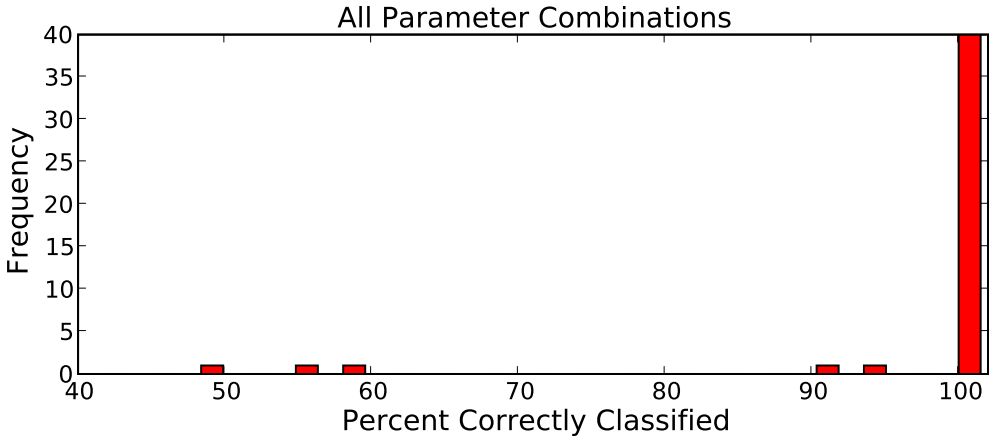


Figure 5.17: Accuracy of hierarchical clustering for the first corpus, with restricted parameter range (see text)

Corpus Two

For the second corpus the grouping is far more difficult as the object classes are less uniform, have a higher resemblance between classes and there is increased variation within classes. The clustering results accumulated over all parameter sets are shown in Figure 5.18. Clearly most parameter combinations result in unacceptable results. Again we try to find parameter settings that work well in a variety of parameter combinations. In contrast to the experiments on the first corpus, there are no perfect classification results; we therefore rather identify and exclude parameter settings, that lead to inferior results.

Figure 5.19 shows the classification accuracy with respect to the different methods of measuring the distance between clusters (see Section 3.4). Contrarily to the first corpus, using the maximum pairwise distance between points in the clusters yields the best results. As depicted in Figure 5.10 some scans segments of a class usually resemble a segment of another class. Under the minimum pairwise distance method, this serves as a "bridge" between classes. If the similarity between two segments of different classes is higher than the similarity of another segment to any of the segments, the former will be connected—possibly joining their classes—while the latter will remain as an individual class. This effect is very comprehensible in a setting depicted in Figure 3.4. If the person stood directly next to the chair, its foot would be connected to the leg of the chair, joining the clusters, before the head and shoulder would be included in the "person" cluster.

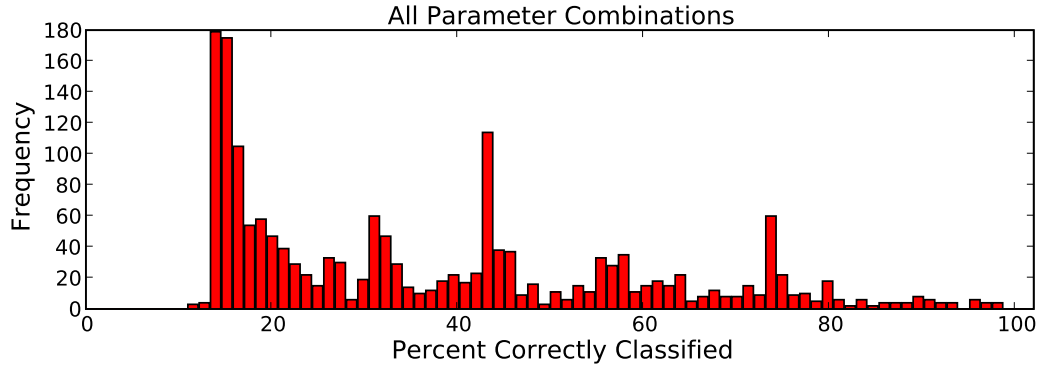


Figure 5.18: Accumulated results for the second corpus.

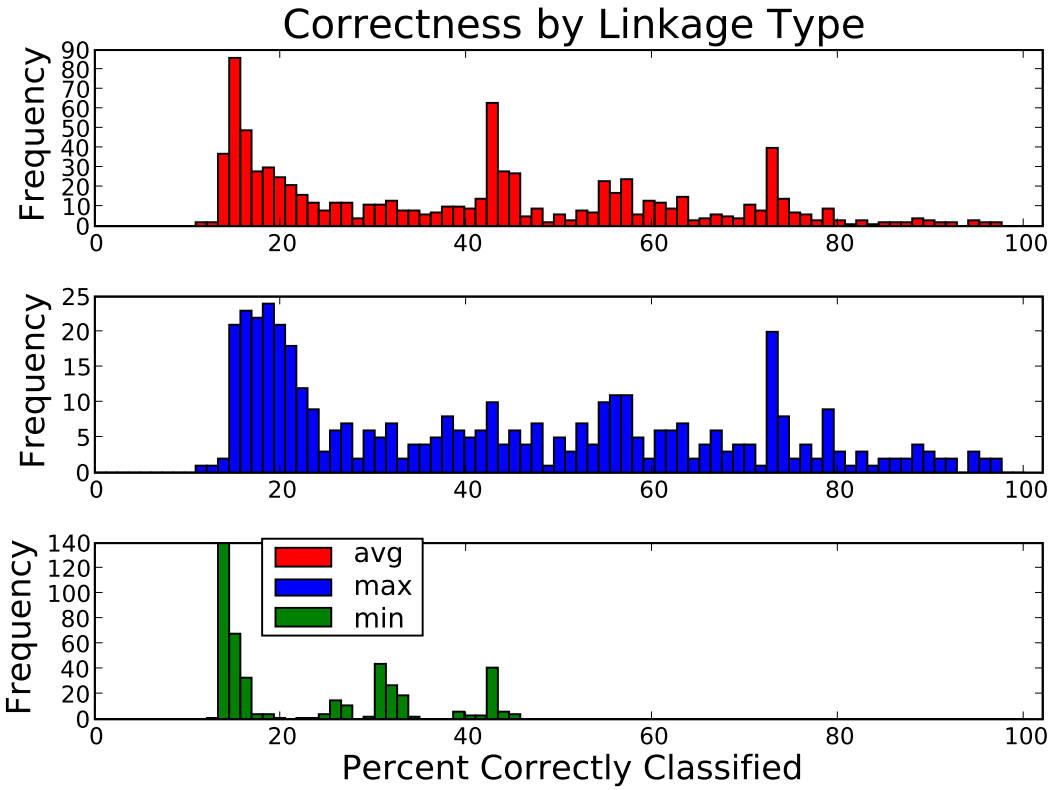


Figure 5.19: Classification results for hierarchical clustering of the feature distributions, with respect to the different methods of distance calculation for clusters.

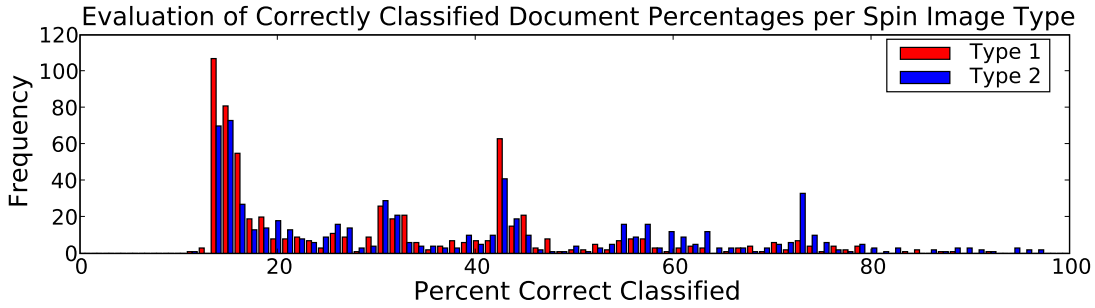
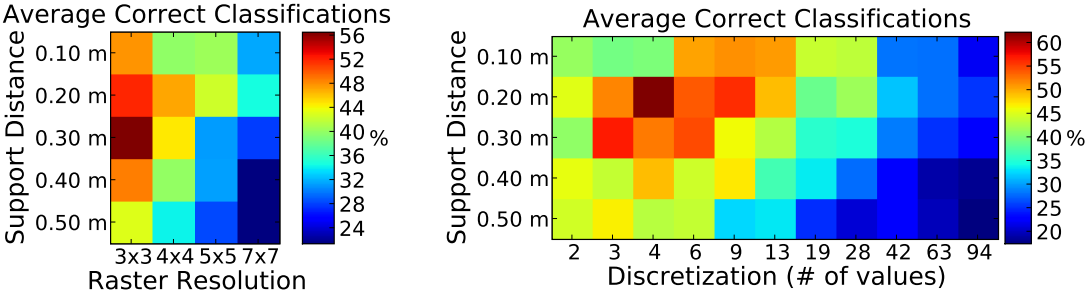


Figure 5.20: Classification results for hierarchical clustering of the feature distribution with respect to the feature type.

This effect leads to the considerably bad results in the bottom chart. Percentages below 20 % in general signify, that nearly all scans have been assigned to a single class. In contrast, in maximum pairwise linkage, the most similar segments of distinct classes do not lead to "bridges" as often, because their distance to the most distinct segment of the other class is still high. Because of the high difference in accuracy, we concentrate on the classification results of the maximum linkage clusters in the following analysis. In the next step we verified our prediction of the distinctive qualities of the feature types. Figure 5.20 shows a bar chart over the classification accuracy for the used feature types, with type one corresponding to standard spin images, while type two denotes our proposal. On the left (worse classification accuracy) we see a prevalence of the first type, on the right (more correct classifications) we find more results based on the second feature type. Figure 5.21 depicts several plots, that show



the average accuracy with respect to the feature generation parameters support distance, raster resolution and discretization resolution. The charts on the right side emphasize that there is no benefit in choosing a higher discretization resolution than nine distinct values, which corresponds to a resolution of 20 degree in the second feature type. The left and lower right chart show that the best results have been achieved using a support distance of 0.2 m. For the raster resolution the picture is not as clear.

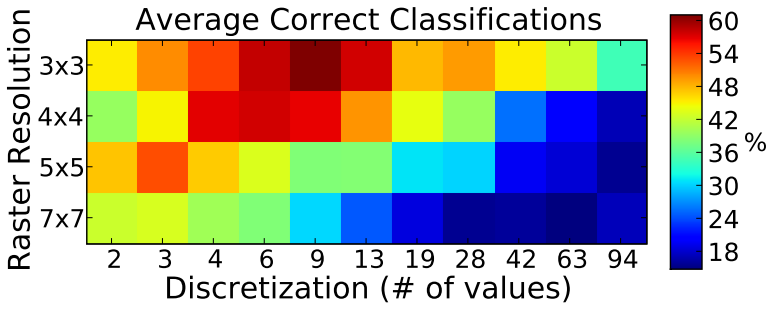


Figure 5.21: Analysis of parameters for feature generation

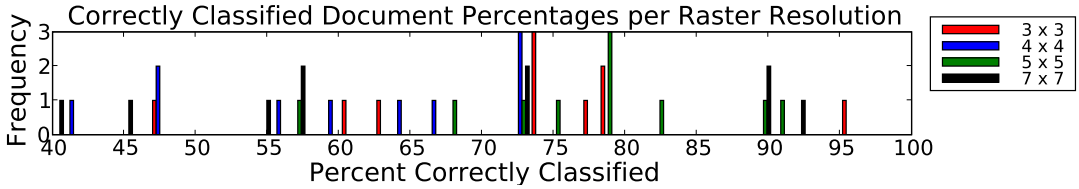


Figure 5.22: Results for a support distance of 20 cm and discretization of less that 10 values

Therefore we disregard the resolutions above nine and the support distances other than 0.2 m and plot the remaining results with respect the raster resolution. The results are shown in Figure 5.22. Unfortunately, there is still a huge variation from 40 % to 96 % in the results, meaning that a very specific parameter set has to be selected to get good results. This is in contrast to the goal stated before, that the results should be robust over a variety of parameter settings.

5.3.2 Latent Dirichlet Allocation

For semantic clustering under the latent Dirichlet allocation model, we use the implementation developed by Phan [2007] to calculate the topic assignments $\mathbf{z}$ and the topic distribution θ. The learning algorithm requires three hyperparameters: The number of topics and the parameter vectors for the prior Dirichlet distributions over the words and the topics respectively. As mentioned in Section 3.3, larger values for α favor the occurrence of many topics in each document, while low priors result in less topics per document. Similarly, the lower the hyperparameter β for the Dirichlet distribution over the words, the stronger the preference for fewer words per topic and unambiguous words. Due to the segmentation in the preprocessing we strongly assume that there are few topics per document. However, a value chosen too low endorses the assignment of the prevailing topic to all features in a document, giving rise to missclassifications.

Also for β, a tradeoff is needed. One the one hand different objects can yield the same discrete features (yet in distinct distribution). On the other hand we need the words to be strongly related to specific topics. From the intuitions for the Dirichlet parameters, as described in Section 3.1, we expect better performance if both parameters, α and β, are between zero and one. We verified this assumption empirically.

Choosing the value for α near one, the topic count per document slowly tends towards a uniform distribution, introducing topics into the document, which are not related to the underlying object. However, choosing α too low will cut out topics from documents rigorously. In the worst case this leads to the elimination of less dominant topics, if they only occur in combination with other topics. Their characteristic words will be assigned to the co-occurring topics.

In the same manner, higher values of β result in bad differentiation, since the probability, for occurrences of a particular word to be assigned to different topics in different documents, is rising. Using a low penalty for word ambiguity, the words of topics with less co-occurrences are assigned to the more dominant topics. Too low values of β enforce a single topic for all occurrences of a word, disregarding the possibility of a word to represent more than one object. Thus, introducing the topic with the higher probability for the word, even though not appropriate. Following these arguments, we experiment with hyperparameters combinations of $0.8 \geq \alpha \geq 0.012$ and $0.8 \geq \beta \geq 0.012$. To compare LDA to the results from hierarchical clustering, we use a maximum likelihood classification in the following experimental evaluation, i.e. we only consider the prevailing topic for each scan. More sophisticated approaches (e.g. k-nearest neighbor, as in Bosch *et al.* [2006]) are thinkable, yet this proceeding will suffice for our purpose.

Corpus One

As for hierarchical clustering we begin our experimental evaluation with the classification of the feature distributions of the first corpus. The results are shown in Figure 5.23. Compared with the corresponding statistics of hierarchical clustering (see Figure 5.14) we notice the missing peak around 40 %, that was due to the assignment of almost all segments to a single class. Although the results are already very good, we still want to exclude parameters leading to bad results. An analysis, analog to our proceeding for hierarchical clustering, confirms again the choice of the second feature type. Also the features with support distances up to 30 cm perform better than less local features. For the remaining feature parameters a compromise between precision and generality should be found, e.g. very coarse discretization resolutions work better with higher raster resolutions and vice versa. The same holds true with respect to support distance. Thus, by concentrating on the parameter combinations in the table below, we have a perfect clustering in 97.5 % of the calculations, with few

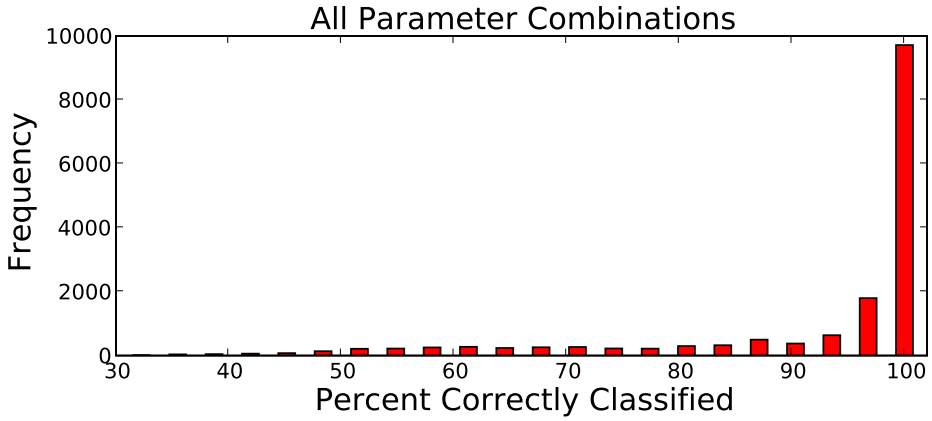


Figure 5.23: Accumulated results for clustering the feature distributions of the first corpus using LDA.

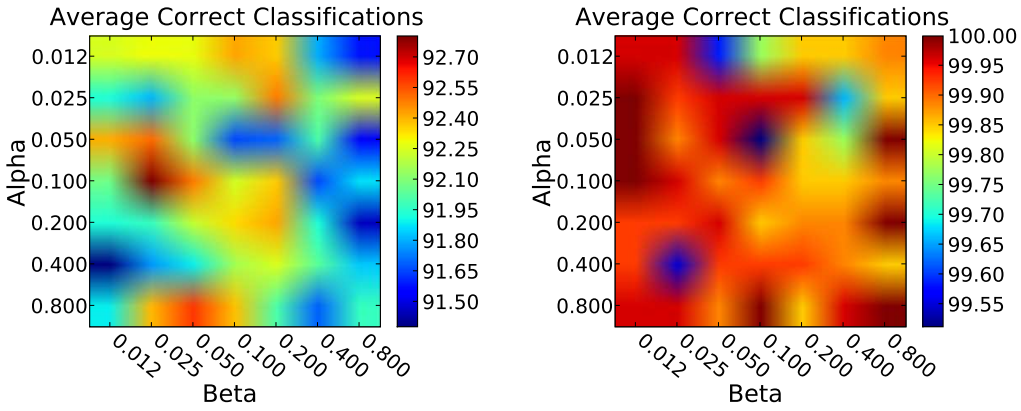
missclassifications for the remaining 2.5 %.

Support Distance	Raster Resolution	Discretization v
0.1 m	3×3	$6 \leq v \leq 94$
	5×5	$3 \leq v \leq 94$
	7×7	$3 \leq v \leq 63$
0.2 m	3×3	$3 \leq v \leq 94$
	5×5	$2 \leq v \leq 94$
	7×7	$2 \leq v \leq 63$
0.3 m	3×3	$2 \leq v \leq 94$
	5×5	$2 \leq v \leq 42$
	7×7	$2 \leq v \leq 27$

These limitations are not particular stringent, as only a minority of the parameter combinations used in the experiments are excluded and the remaining parameter space is quite flexible.

The results with respect to the hyperparameters of LDA, α and β, are shown in Figure 5.3.2. The left hand plot shows the correctness of clustering, averaged over all settings. The right hand side is averaged only over the subset of parameter space defined above. Unfortunately, no particular region seems to be consistently favorable, yet the variation is rather small, such that the particular choice of hyperparameters within the given bounds seems to be neglectable.

To illustrate the clustering results, Figure 5.3.2 shows a point-wise color labeling of two example input scenes. The labels assigned to the points are taken from a sample of the posterior distribution $P(\mathbf{z}|\mathbf{w})$, as generated when learning the clustering. The results for this corpus are perfect for the maximum likelihood classifier, but also nearly perfect for the pointwise classification. Misclassified points, if any, are very sparse and could easily be filtered, e.g. using



(a) All Settings **(b)** Selected Settings

Figure 5.24: Average accuracy

a Markov random field.

Corpus Two

Feature Parameters

As for hierarchical clustering we evaluate the parameter combinations by calculating the percentage of correctly classified documents for each corpus and each combination of learning parameters. This gave us 21560 percentages, 10780 for each feature type. The distribution of these percentages for the feature types are shown in Figure 5.26. The higher red bars on the left side show that classification on standard spin images ("Type 1") yields bad results more often, while our feature proposal ("Type 2") often allows for correct classification of 80 percent or more of the documents (high blue bars at the right side). This again validates our conclusion to concentrate on the second type in further comparisons.

In the next step we measure classification accuracy, with respect to support distance and raster resolution. The charts on the left of Figure 5.27 again show, for each percentage of correctly classified documents (along the x-axis), how often it was achieved with the mentioned parameters. It is clearly visible that the features with support distance of 0.5 m and 0.4 m are inferior because often less than 50 % of the documents are correctly classified. In contrast with a support distance of 0.1 m, very few parameter combinations result in bad classification rates. Most corpora are classified correctly to 80 % and more. Similarly for higher raster resolutions the results are worse than for coarse rasters. The matrix on the right shows the interdependencies of the combinations of the parameters, endorsing what can be seen in the individual graphs. In conclusion the following step only includes the parameter combinations with an average

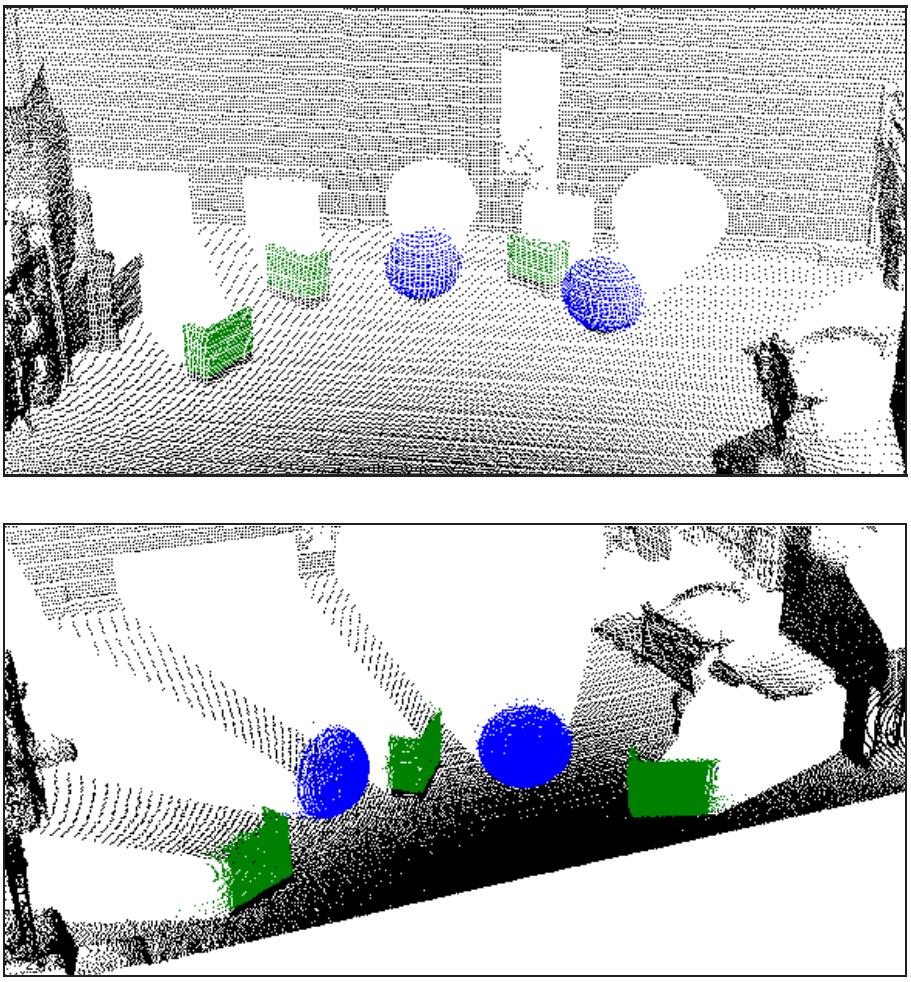


Figure 5.25: Illustrations of the point-wise labeling of the range data with samples of the LDA topic assignment vector $\mathbf{z}$

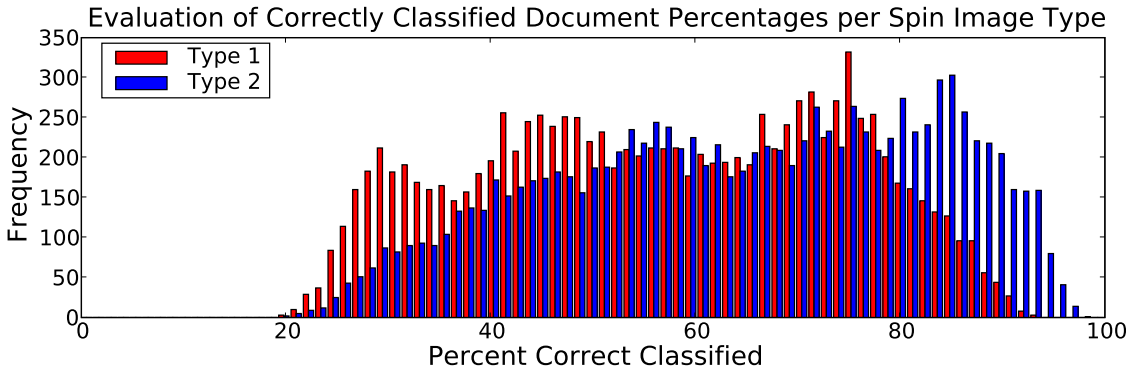


Figure 5.26: Classification upon standard spin image features ("Type 1" shown in red) generally labels less documents correctly than classification upon the features we proposed ("Type 2", blue).

of 70 % and above. Figure 5.28 depicts the performance with respect to the remaining parameter sets and discretization resolutions. We see that for smaller support distance, higher discretization resolutions work well and vice versa. This is due to the need for generalization, as feature distributions with a large support and a very accurate discretization have too precise features, that do not matched to the distributions of other segments. Here the size of the data set plays a role, as for a sufficiently large data set, every feature, however precise, occurs often enough, such that it can be matched with features from other distributions.

The best results in our setting are obtained for strongly localized features, with a discretization resolution between five and 27. In conclusion we see, that choosing such parameters for the feature generation, we can achieve over 85 % correct classifications, averaged over the learning parameters. This is quite impressive given the difficulty of the data set.

LDA Hyperparameters

Having identified suitable parameters for feature generation, we still need to find a good compromise for the hyperparameters of LDA. As mentioned earlier we assume symmetric Dirichlet priors, because we neither anticipate certain topics, nor particular features. Also the priors should be below one, since we assume few objects to fall into one scan segment and expect the features to be typical for few object classes. Therefore, for both α and β, we evaluated all combinations of values ranging from 0.012 to 0.8. Figure 5.29 shows the classification results with respect to α and β. The right plot shows the accumulation over all feature generation parameter combinations. On the left, results are accumulated over the feature sets chosen in the previous section. Note the difference in scale. In both plots the value of β seems to be of minor importance. For α, the left

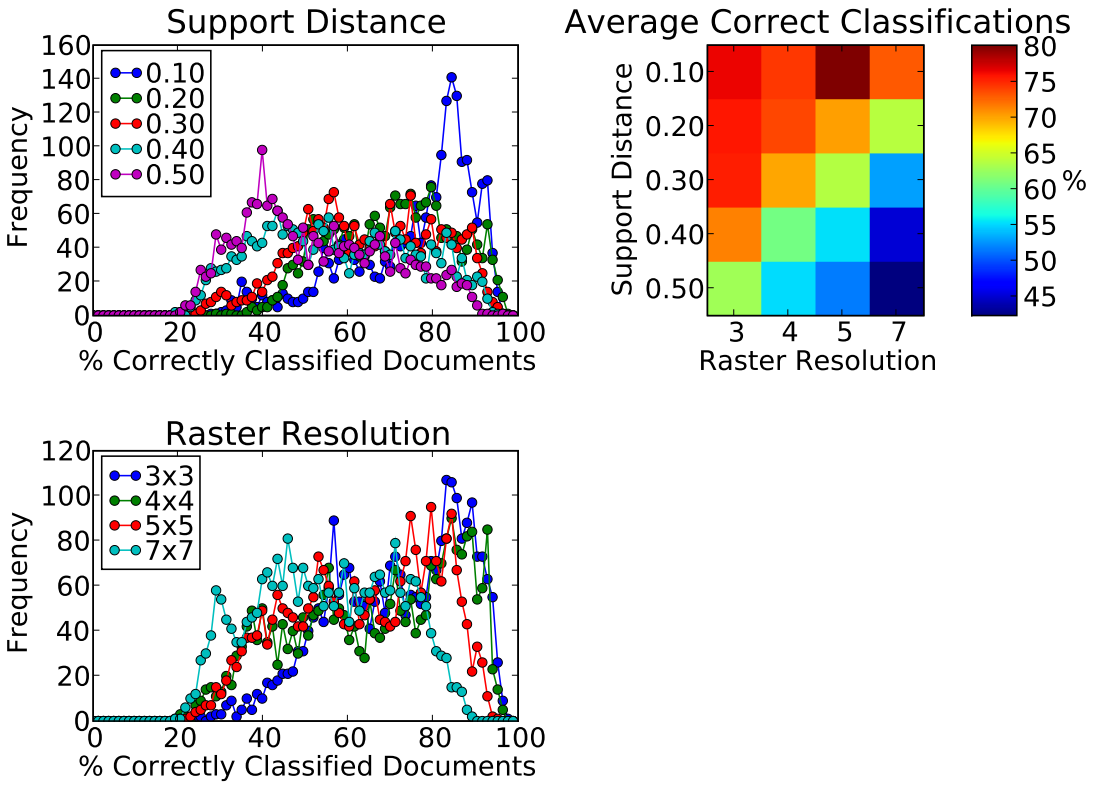


Figure 5.27: Frequency of classification results per support distance (upper left) and raster resolution (lower left). Smaller (0.10 cm) and less detailed (3×3) features result in better classification rates. The plot at the right confirms this relation for various combinations of parameters

side suggests values between 0.025 and 0.05. On the left we see, that this restriction becomes less important when selecting appropriate parameters for feature generation, such that only large values of α lead to drastically worse clustering results.

To illustrate the clustering results for the second corpus, Figure 5.3.2 shows a point-wise color labeling of two example scenes. Again, the labels assigned to the points are taken from a sample of the posterior distribution $P(\mathbf{z}|\mathbf{w})$, as generated when learning the clustering. While clearly visible that the majority of the points has been labeled correctly, such that the correct topic has the maximum likelihood, the differentiation is not as perfect as for the two simple object classes of the first corpus. Some interesting aspects of the feature to class assignments can be seen: The box contains two stripes of points, that are classified the same as the balloon. This is due to the smoothing effect when calculating the surface normal using the PCA method. Since we average over

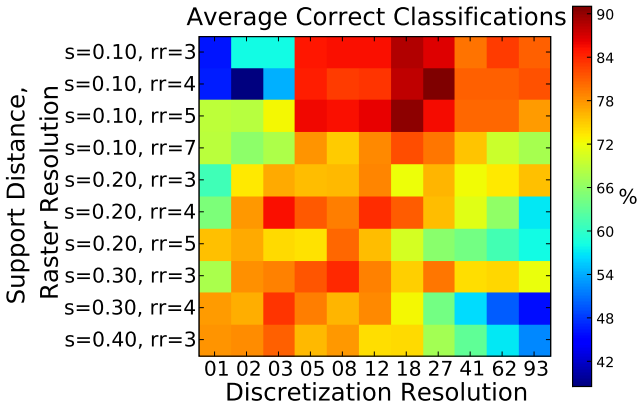
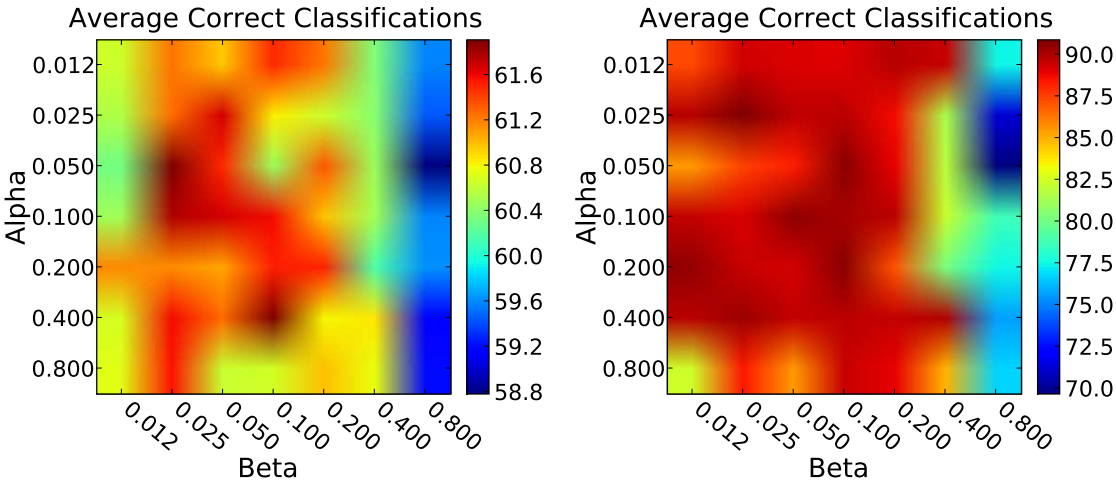


Figure 5.28: Classification accuracy for the discretization resolutions with respect to support distance and raster resolution

the surface normals intersecting with the "spinning image" of the feature, in that particular area, the averaging of the smoothed corner with the flat sides of the boxes is similar to the curvature of the balloon. This also explains why the edge itself is not similar to the balloon, as the averaging here lacks the flat parts. Other missclassifications include the upper corner of the back of the chair, which is labeled like the box. Knowing that features with a radius of 10 cm are used, this seems not too unreasonable, as the backrest is completely flat in this area. Also the partial "balloon" classifications of the humans seem sensible enough, considering the locality of the features.

5.4 Evaluation

Comparing the unsupervised clustering methods presented in this chapter, we see that the usage of latent Dirichlet allocation clearly leads to better and more stable results than the usage of hierarchical clustering. For the first corpus, both methods mostly lead to perfect classification results. The advantage of LDA here, is the better performance across different support distances. Also hierarchical clustering tends to put outliers (scan segments dissimilar to all other segments) into an individual cluster. This does not occur using LDA for clustering. The second corpus is difficult for both methods. Neither was able to classify all scan segments perfectly. Using hierarchical clustering, we were not even able to find a range of parameters with reliable results. With LDA, however, we successfully determined a suitable range of parameters, where classification results are correct for more than 85 % of the scan segments; individual param-



(a) All Parameter Sets **(b)** Optimal Range (see Text)

Figure 5.29: Evaluation of classification accuracy for various values of alpha and beta

eter combinations even classify up to 98% of the segments correctly. Given the difficulty of the input data, this is an highly satisfactory result. We also found some limitations of LDA-based classification. These include the rather high computational complexity, that is particular problematic in applications with many object classes and high measurement densities. The randomized initialization of the Markov chain influences the clustering result. This renders exact reproduction of results difficult, which might be critical for some applications. Our evaluation of the parameter space showed that the feature-specific parameters depend on the characteristics of the corpus. This is unfortunate—yet not surprising—as it causes the need for application specific adaptation of the parameters. On the other hand, we found the optimal range of hyperparameters of LDA to be valid for both data sets, which indicates that no adjustments are necessary here. However, due to the small number of corpora we used, this conclusion needs further experimental evidence.

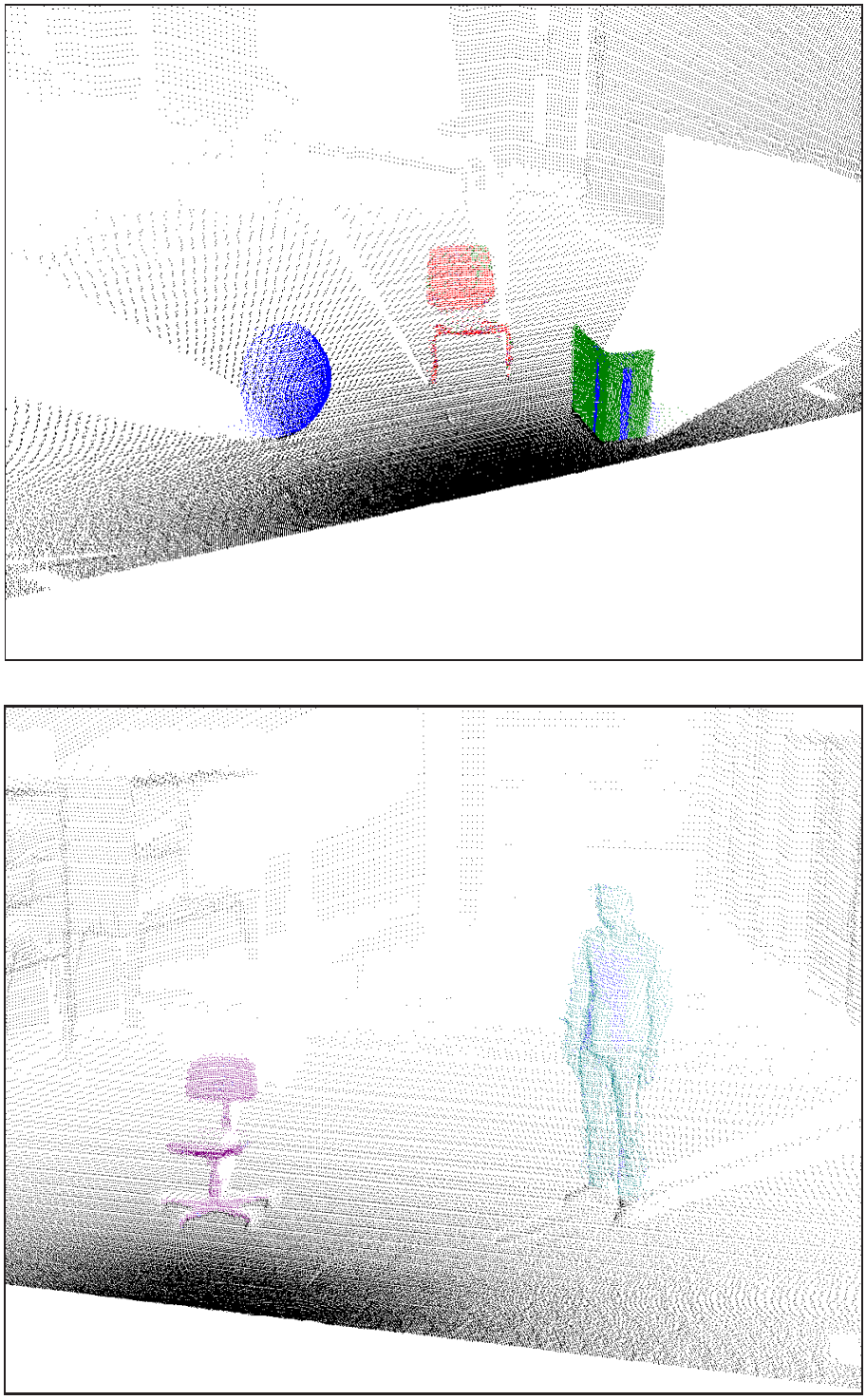


Figure 5.30: Illustrations of the point-wise labeling of the range data with samples of the LDA topic assignment vector $\mathbf{z}$ for scans of corpus two

6 Conclusions and Outlook

6.1 Conclusions

In this thesis, we presented a new approach to scene analysis from range data. We perform unsupervised discovery of object classes in 3D range data, based on feature distributions. For the purpose of shape based clustering, we extract surface descriptors from 3D point clouds. We evaluate spin images and develop an enhanced, local, free-form surface descriptor, that shows better discriminative performance in our experiments. Instead of summing points projected to a coordinate system relative to the query point, our proposal is to consider the angle between the surface normal at the query point and the surface normals of the projected points.

We apply latent Dirichlet allocation to the feature distributions, to learn a clustering of 3D objects according to similarity in shape. The learned feature distributions of the clusters can subsequently be used as models for classification of unseen data. An important advantage of our approach is that there is no need for labeled training data to learn the partitioning.

In experiments on laser range scans, we evaluate the accuracy of our approach. We apply it to two data sets of differing difficulty. For simple, distinct objects, we achieve perfect classification on a wide variety of settings for the feature generation process. For the second data set, containing more complex objects with varied appearance, we still achieve a robust performance with over 85 percent correctly classified objects. We compare the results to another approach in which hierarchical clustering is used, instead of latent Dirichlet allocation, to determine the partitioning of the data. According to our expectations the performance of our approach is superior in terms of achieved accuracy, as well as with respect to robustness to variations in the feature generation step.

6.2 Future Work

In this section, we want to present some proposals for future improvements of our approach. First of all, being a feature based approach, further improvements in surface signatures will directly improve the results of classification, in particular for difficult data. Also using a mixture of features, covering different object characteristics could improve the categorization quality. However, a mixture of the two feature types used in this thesis, did not improve the results.

In data preprocessing we relied on a spatial clustering to get scan segments that contain a single object class. This bases on the assumption, that there is a gap in-between objects and thus requires steps like floor extraction, to ensure suitable gaps are there. A more sophisticated proceeding would possibly be favorable, with respect to applicability to range scans where such preprocessing is difficult, e.g. in case of highly cluttered scenes. Furthermore we determine the object class by choosing the prevailing topic as assigned by LDA. In many cases an object class is a mixture of several kinds of feature distributions. A good example is the chair class, that has a lot in common with the box class, yet the presence of some features strongly suggest an object belongs to the chair class. Here a more sophisticated choice of the final object class might improve the results for classes that share features.

In terms of computational cost, currently the most limiting factor is feature generation. We generate one surface descriptor per point measurement, which generates a lot of redundant features. Enhancements could be achieved using point-of-interest or region-of-interest detectors. Also, a good sampling strategy would lead to improved efficiency. Due to its ability to work on discrete data, regardless of the information content, our approach is easily adaptable to include feature information from other sensors. Thus, the inclusion of e.g. visual information is very likely to improve the results of the presented method.

A Classified Scan Segments

At this point we show learning results for the second corpus, generated using our feature proposal to build the feature distributions and latent Dirichlet allocation for clustering. The parameters used for feature generation are

Support Distance:	0.2 m
Raster Resolution:	3×3
Discretization Res.:	13 values

For LDA we set the topic count to five, which is the number of classes present in the corpus, and chose the parameters for the prior Dirichlet distributions to be

Alpha:	0.025
Beta:	0.050

Figure A.1 shows the topic distributions found for the individual scan segments. Each color corresponds to a topic id. There is a strong tendency towards few dominant topics in the distribution; This is due to the low value for α. The following pages show the sampled topic-to-word—i.e. class id to point measurement—assignment vector $\mathbf{z}$ for a selection of scan segments. The id to color correspondence is the same as in Figure A.1.

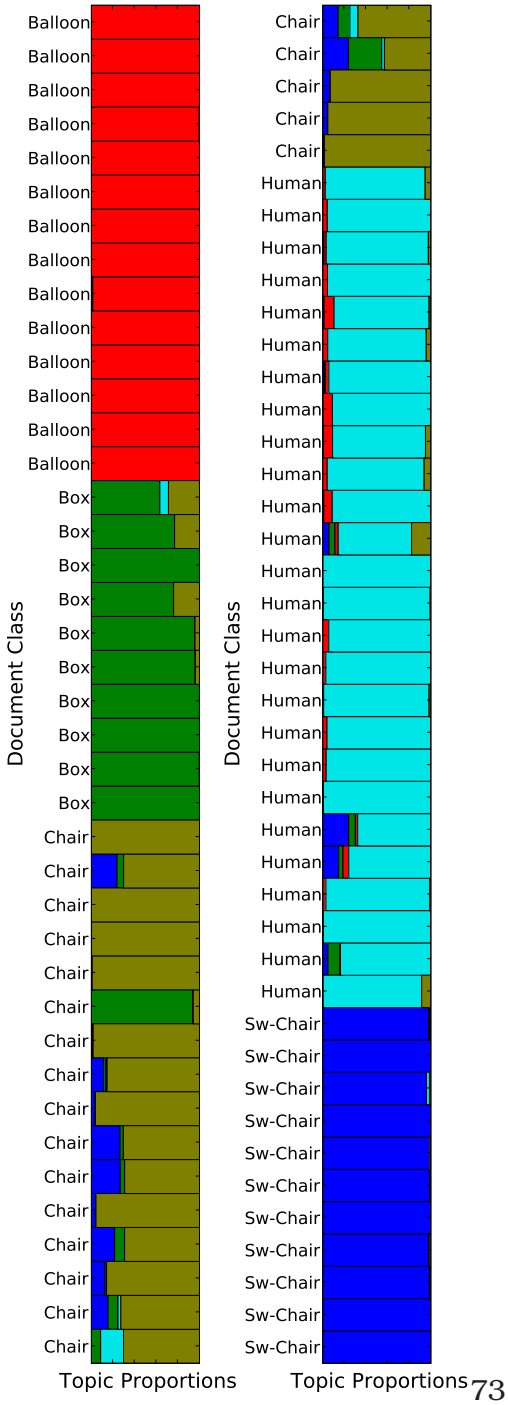


Figure A.1: Learning results, topic distribution

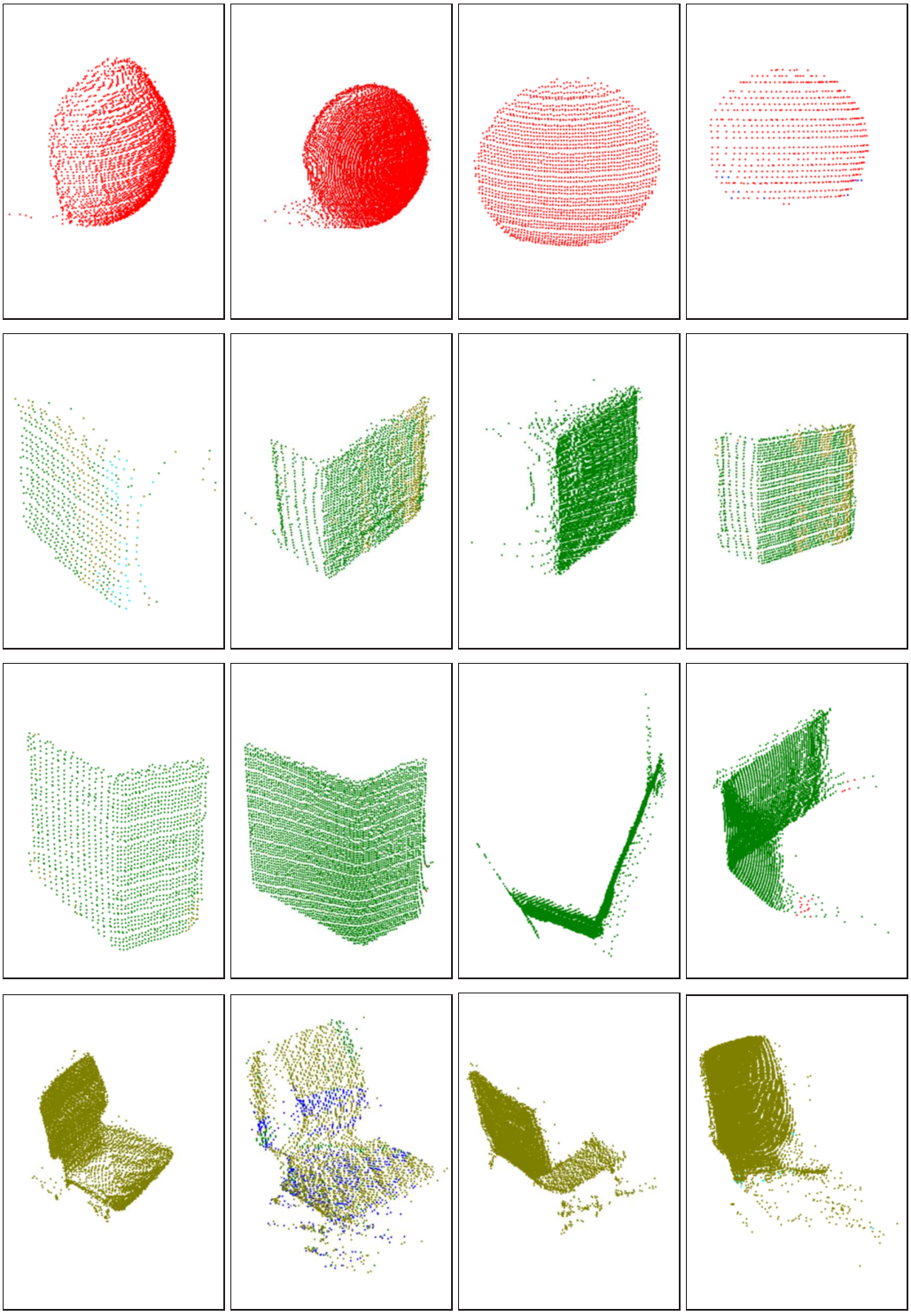


Figure A.2: Scan Segments

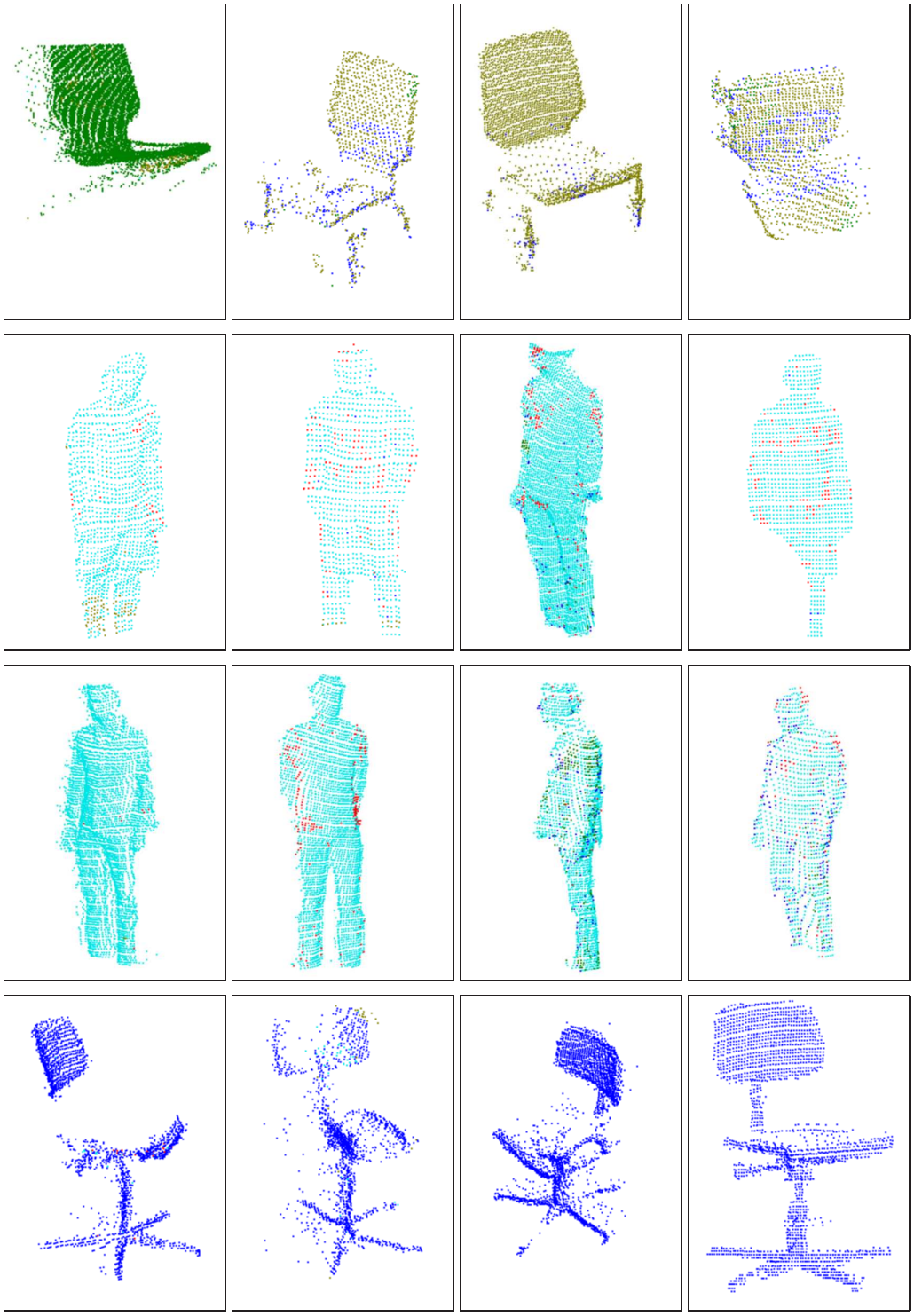


Figure A.3: Scan Segments (continued)

List of Figures

Bibliography

[Anguelov *et al.*, 2005] Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., and Ng, A. *Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data*. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 169–176, Washington, DC, USA, 2005. IEEE Computer Society.

[Bay *et al.*, 2008] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. *Speeded-Up Robust Features (SURF). Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[Bishop, 2006] Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

[Blei *et al.*, 2003] Blei, D. M., Ng, A. Y., Jordan, M. I., and Lafferty, J. *Latent dirichlet allocation*. *Journal of Machine Learning Research*, 3:2003, 2003.

[Bosch *et al.*, 2006] Bosch, A., Zisserman, A., and Munoz, X. *Scene classification via pLSA*. In *In Proc. ECCV*, pages 517–530, 2006.

[Bosch *et al.*, 2007] Bosch, A., Zisserman, A., and Munoz, X. *Representing shape with a spatial pyramid kernel*. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408. ACM Press New York, NY, USA, 2007.

[Brown, 1981] Brown, C. M. *Some mathematical and representational aspects of solid modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:444–453, 1981.

[Bustos *et al.*, 2005] Bustos, B., Keim, D. A., Saupe, D., Schreck, T., and Vranić, D. V. *Feature-based similarity search in 3D object databases. ACM Comput. Surv.*, 37(4):345–387, 2005.

[Campbell and Flynn, 2001] Campbell, R. J., and Flynn, P. J. *A survey of free-form object representation and recognition techniques. Comput. Vis. Image Underst.*, 81(2):166–210, 2001.

[Chen *et al.*, 2003] Chen, D., Tian, X., Shen, Y., and Ouhyoung, M. *On Visual Similarity Based 3D Model Retrieval*. In *Computer Graphics Forum*, volume 22, pages 223–232. Blackwell Synergy, 2003.

[Csurka *et al.*, 2004] Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. *Visual categorization with bags of keypoints*. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[Dalal and Triggs, 2005] Dalal, N., and Triggs, B. *Histograms of Oriented Gradients for Human Detection*. volume 1, pages 886–893, 2005.

[Fehr and Burkhardt, 2007] Fehr, J., and Burkhardt, H. *Harmonic Shape Histograms for 3D Shape Classification and Retrieval*. In *IAPR Workshop on Machine Vision Applications (MVA2007)*, Tokyo, Japan, 2007.

[Ferrari *et al.*, 2008] Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. *Groups of Adjacent Contour Segments for Object Detection. IEEE Transactions On Pattern Analysis And Machine Intelligence*, pages 36–51, 2008.

[Fritz and Schiele, 2008] Fritz, M., and Schiele, B. *Decomposition, discovery and detection of visual categories using topic models*. In *CVPR08*, pages 1–8, 2008.

[Frome *et al.*, 2004] Frome, A., Huber, D., Kolluri, R., Bulow, T., and Malik, J. *Recognizing Objects in Range Data Using Regional Point Descriptors*. pages Vol III: 224–237, 2004.

[Girolami and Kabán, 2003] Girolami, M., and Kabán, A. *On an equivalence between PLSI and LDA*. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 433–434. ACM Press New York, NY, USA, 2003.

[Griffiths and Steyvers, 2004] Griffiths, T. L., and Steyvers, M. *Finding scientific topics. Proc Natl Acad Sci U S A*, 101 Suppl 1:5228–5235, 2004.

[Griffiths, 2004] Griffiths, T. *Gibbs Sampling in the Generative Model of Latent Dirichlet Allocation*, 2004.

[Hetzel *et al.*, 2001] Hetzel, G., Leibe, B., Levi, P., and Schiele, B. *3D object recognition from range images using local feature histograms*. In *Proceedings of CVPR 2001*, pages 394–399, 2001.

[Hofmann, 1999] Hofmann, T. *Probabilistic latent semantic indexing*. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM Press New York, NY, USA, 1999.

[Hoover *et al.*, 1996] Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A., and Fisher, R. B. *An Experimental Comparison of Range Image Segmentation Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996.

[Johnson and Hebert, 1996] Johnson, A., and Hebert, M. *Recognizing Objects by Matching Oriented Points*. Technical Report CMU-RI-TR-96-04, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1996.

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

[Johnson and Hebert, 1999] Johnson, A. E., and Hebert, M. *Using Spin Images for Efficient Object Recognition in Cluttered 3D scenes*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:433–449, 1999.

[Johnson, 1997] Johnson, A. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1997.

[Lampert *et al.*, 2008] Lampert, C. H., Blaschko, M. B., and Hofmann, T. *Beyond sliding windows: Object localization by efficient subwindow search*. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[Lowe, 1999] Lowe, D. *Object recognition from local scale-invariant features*. In *International Conference on Computer Vision*, volume 2, pages 1150–1157. Kerkyra, Greece, 1999.

[Mikolajczyk and Schmid, 2005] Mikolajczyk, K., and Schmid, C. *A performance evaluation of local descriptors*. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[Phan, 2007] Phan, X.-H. *GibbsLDA++*, 2007. [Online; accessed 04-August-2008].

[Philbin *et al.*, 2008] Philbin, J., Sivic, J., and Zisserman, A. *Geometric LDA: A Generative Model for Particular Object Discovery*. In *Proceedings of the British Machine Vision Conference*, 2008.

[Rowley *et al.*, 1996] Rowley, H., Baluja, S., and Kanade, T. *Human Face Detection in Visual Scenes*. *Advances In Neural Information Processing Systems*, pages 875–881, 1996.

[Ruhnke, 2008] Ruhnke, M. *Unüberwachtes Lernen von 3D Modellen für nicht stationäre Objekte auf volumetrischen Daten*. Diplomarbeit, University of Freiburg, 2008.

[Ruiz-Correa *et al.*, 2003] Ruiz-Correa, S., Shapiro, L. G., and Meila, M. *A New Paradigm for Recognizing 3-D Object Shapes from Range Data*. *Computer Vision, IEEE International Conference on*, 2:1126, 2003.

[Schroll, 2008] Schroll, P. *Segmentierung und Lagebestimmung von 3D Objekten in Laser Range Scans*. Diplomarbeit, University of Freiburg, 2008.

[Shilane *et al.*, 2004] Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T. *The Princeton Shape Benchmark*. In *Shape Modeling Applications, 2004. Proceedings*, pages 167–178, 2004.

[Siggelkow *et al.*, 2001] Siggelkow, S., Schael, M., and Burkhardt, H. *SIMBA - Search IMages By Appearance*. In Radig, B., and Florczyk, S., editors, *Pattern Recognition, DAGM, LNCS 2191*, pages 9–16, München, Germany, 2001.

[Steder *et al.*, 2009] Steder, B., Grisetti, G., Loock, M. V., and Burgard, W. *Robust On-line Model-based Object Detection from Range Images*. In *Proc. of the International Conference on Robotics and Automation(ICRA)*, 2009. Under review.

[Stein and Medioni, 1992] Stein, F., and Medioni, G. *Structural Indexing: Efficient 3-D Object Recognition*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–145, 1992.

[Streicher, 2008] Streicher, A. *3D Shape Retrieval mit lokalen Merkmalen*. Diplomarbeit, University of Freiburg, 2008.

[Triebel *et al.*, 2006] Triebel, R., Kersting, K., and Burgard, W. *Robust 3D Scan Point Classification Using Associative Markov Networks*. 2006.

[Triebel *et al.*, 2007] Triebel, R., Schmidt, R., Mozos, O. M., and Burgard, W. *Instace-based AMN Classification for Improved Object Recognition in 2D and 3D Laser Range Data*. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2225–2230, Hyderabad, India, 2007.

[Wang and Grimson, 2007] Wang, X., and Grimson, E. *Spatial Latent Dirichlet Allocation*. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

[Wikipedia, 2008] Wikipedia. *Plate notation — Wikipedia, The Free Encyclopedia*, 2008. [Online; accessed 24-October-2008].

[Zhang *et al.*, 2007] Zhang, J., Marszalek, M., Lazebnik, S., Schmid, and C. *Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study*. *International Journal of Computer Vision*, 73(2):213–238, 2007.