3D Shape Models for Object Categorization and Pose Estimation

Dissertation

zur Erlangung des akademischen Grades
Doktor der technischen Wissenschaften
an der
Technischen Universität Graz

vorgelegt von

Dipl.-Ing. Kerstin Pötsch

Begutachter:
Ao. Univ.-Prof. Dipl.-Ing. Dr. Axel Pinz
Prof. dr. sc. Siniša Šegvić

September, 2011

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Kerstin Pötsch

Acknowledgement

This PhD thesis is the result of research at the vision-based measurement group of the Institute of Electrical Measurement and Measurement Signal Processing at the Graz University of Technology. From June 2007 to April 2011 my PhD thesis was founded by the Austrian Science Fund (FWF) under the doctoral program Confluence of Vision and Graphics W1209.

First of all, I would like to thank my supervisor, Prof. Axel Pinz, for the opportunity to join the vision-based measurement group as a PhD student in the doctoral program, the possibility to realize this thesis, and his support throughout this work. Furthermore, I would like to thank all members of the Institute of Electrical Measurement and Measurement Signal Processing and my colleagues during this time.

Also, I would like to thank Prof. Siniša Šegvić from Faculty of Electrical Engineering and Computing of University of Zagreb who agreed to be the second examiner of this thesis and who gave some useful and valuable final remarks.

I would thank my family for their love and support during this period and always on the way which I chose. Finally, I want to thank my partner Hubert for his love, his patience, his critical questions and long discussions and especially for being part of my life.

Thank you all!

Graz, September 2011 Kerstin Pötsch

Abstract

Category and pose hypotheses are crucial for certain computer vision applications such as 'active object categorization', where the active movement of a camera can be based on these hypotheses. Motivated by the success of 2D shape models for category hypotheses this thesis investigates methods for view-independent object categorization and pose estimation based on 3D shape models. With this 3D approach we can resolve problems of sensitiveness to view/pose changes inherent in 2D models. This thesis addresses three main aspects:

- (How) can we model 3D shape for specific objects?
- (How) can we learn one single, pose-invariant 3D category model based on shape information?
- (How) can we use such a 3D category model for pose estimation in 2D images?

First, I propose a stereo reconstruction framework for geometric 3D shape modeling by 3D contour fragments (1D manifolds in 3D) - the resulting 3D shape model is called a '3D contour cloud'. In the second part, I describe how to use a set of such '3D contour clouds' in a probabilistic framework based on Gaussian Mixture Models for learning a 3D category model. The third major part of this thesis describes the application of such 3D category models for pose estimation in 2D images. Several experiments on the GRAZ-STEREO-BASE-XX dataset (which was also developed as a part of this thesis) as well as on well-known datasets such as the ETH-80 dataset demonstrate that 3D Gaussian Mixture Category Models based on 3D contour fragments are suitable category representations for object categorization and pose estimation.

Kurzfassung

Kategorie- und Posenhypothesen spielen eine wichtige Rolle in verschiedensten Anwendungen der Bildverarbeitung. Eines der prominentesten Beispiele hierbei ist die 'aktive Objektkategorisierung' bei welcher die Kamera aktiv um das Objekt bewegt wird, um eine Kategorie- und Posenhypothese anhand einer weiteren Ansicht zu verifizieren und zu verbessern. Meine Dissertation beschäftigt sich mit Objektkategorisierung und Posenbestimmung basierend auf 3D Formmodellen. Motiviert wird dies durch den Erfolg von 2D formbasierten Modellen, die jedoch anfällig auf Ansicht- bzw. Posenveränderung reagieren. Diese Dissertation behandelt im Besonderen folgende drei Fragen:

- (Wie) können wir die 3D Form von spezifischen Objekten beschreiben?
- (Wie) können wir ein formbasiertes 3D Kategoriemodell lernen?
- (Wie) können wir dieses zur Posenbestimmung in 2D Bildern verwenden?

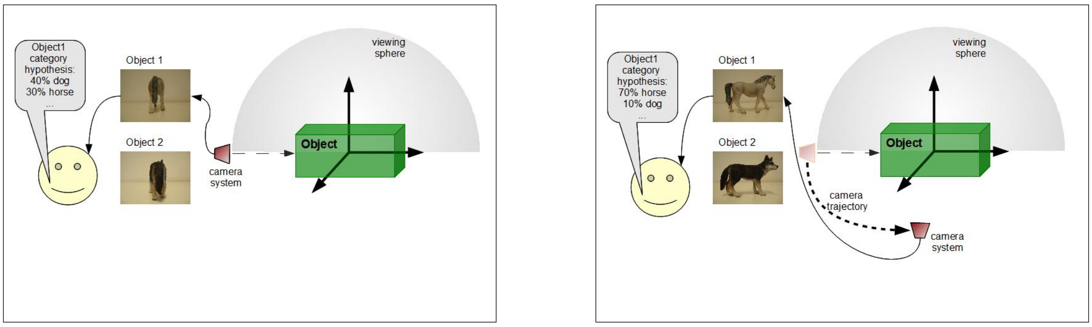
Zu Beginn präsentiere ich einen Rekonstruktionsalgorithmus zur geometrischen 3D Modellierung von Objekten durch 3D Konturfragmenten, eine so genannte '3D contour cloud' (3D Konturwolke). Diese Konturwolken für spezifische Objekte bilden die Basis für das 3D Kategoriemodell, bei dem jedes 3D Konturfragment durch ein Gaussian Mixture Modell repräsentiert und ein probabilistischer Ansatz zum Lernen eines solchen 3D Kategoriemodells angewendet wird. Im Weiteren beschreibe ich einen probabilistischen Ansatz zur Posenbestimmung basierend auf diesen 3D Kategoriemodellen. Durch die experimentelle Auswertung auf unserer eigenen GRAZ-STEREO-BASE-XX Datenbank, die im Zuge dieser Arbeit entstanden ist, und auf Standard-Bilddatenbanken wie etwa der ETH-80 Datenbank, wird die Möglichkeit von Kategorisierung und Posenbestimmung basierend auf 3D Formmodellen demonstriert.

Contents

1 Introduction

Assume an active object categorization system, which takes one image to compute a category and pose hypothesis from one object and which is able to take another view of the object to verify its hypothesis (see Figure 1.1). For such a categorization system it is crucial to do object categorization independent from the view/pose of the object. The way to realize object categorization is to learn for each category a model and to match this model against the input image. Because there is an endless number of different poses from which you can see an object, you have to model all possible aspects for an object, especially for a shape-based categorization. Most of the current object categorization systems are based on 2D and just learn one aspect per model. What would be the best way to realize that: to have one 2D model per aspect and learn all aspects separately or to have one single, pose-invariant 3D representation of a category?

This thesis aims to provide some answers to questions like the above by modeling 3D shape of categories instead of modeling each 2D aspect separately. In Section 1.1, I start with a motivation for the importance of 3D model based systems. Then, in Section 1.2, I describe the problem statement of my thesis - object categorization based on 3D models - my way to compute a category and pose hypothesis of objects, as well as my contributions.



(a)

(b)

Figure 1.1: Active object categorization system: (a) A camera system captures an image of an object and a categorization system computes a category hypothesis for that object (e.g. 40% dog, 30% horse). (b) The camera system moves around the object to a view that is particularly useful to improve the hypothesis (e.g. to 70% horse, 10% dog)). The figure shows images for two objects in order to demonstrate that some views, e.g. as shown in (b), are typically more useful to identify the object category than others, e.g. as shown in (a).

1.1 Motivation

Object categorization - in particular view/pose independent object categorization - is important for many computer vision applications such as active object categorization, active surveillance or robotics (e.g. active object manipulation).

Object categorization (generic object recognition) is the process of detecting objects in images (detection and localization) and assigning categories (as car, bike, horse, cow, human ...) to those objects. Object categorization often includes pose estimation, which is the task of identifying the pose of an object with respect to a camera (e.g. car front, car rear, car side). In contrast to generic object categorization, specific object recognition is the process of recognizing one specific object (e.g. my car, my bike). In this thesis, I solely address generic object recognition (I will use the term *'object categorization'* in this thesis) with a focus on pose-invariant 3D shape models for object categorization and pose estimation.

Most of the current research in object categorization is based on 2D models. Here, we distinguish between appearance-based and shape-based categorization methods (an overview of object categorization can be found in [Pin05]). On the one hand, there are appearance-based methods, which focus on describing an object category by a codebook of appearance patches (interest point + region). In the simplest way such a model has no geometric information as e.g. Bag-of-Words model [CDF$^+$04]. Other models as e.g. the constellation model [FPZ03] combine appearance information with geometric information as spatial relations. On the other hand, shape based methods have been successfully used in object categorization. Such methods describe the shape of an object category by 2D contour fragments of silhouette and inner contour fragments in combination with spatial relations (see Section 2.1 for a state-of-the-art overview). However, all these 2D based object categorization systems are view-dependent in that sense that they are sensitive to view/pose changes. Consequently, one model per aspect has to be learned independently to achieve robustness to view and pose changes. Even if more 2D views are combined in an object categorization system, they can not cover the whole 3D nature of an object. Beside challenges such as intra-class variability, inter-class difference, background clutter, illumination, occlusion, rotation, translation, scale and deformation, robustness to viewpoint and pose changes is among the main difficulty in object categorization (see Figure 1.2 for various views of the object category ‘car’). Therefore, pose-invariant representations of objects and object categories are attractive for vision tasks. In the last years multi-view and 3D models for object categorization attained acceptance in computer vision (see Section 2.2).

If the object categorization system is based on one single, pose-invariant 3D model per category we can overcome problems of categorization systems based on 2D models per aspects[1] and objects may be categorized from arbitrary viewpoints. Moreover, if the underlying category is modeled as 3D, there is no restriction about the dimension of the input data. The categorization can be done in 3D on a 3D model or in 2D on a new input image. Another important advantage is that relations between views are known, which is beneficial for systems such as e.g.

[1] In this thesis, the term ’aspect’ is used for a significant visual view in a geometrical meaning, similar as in an aspect graph.

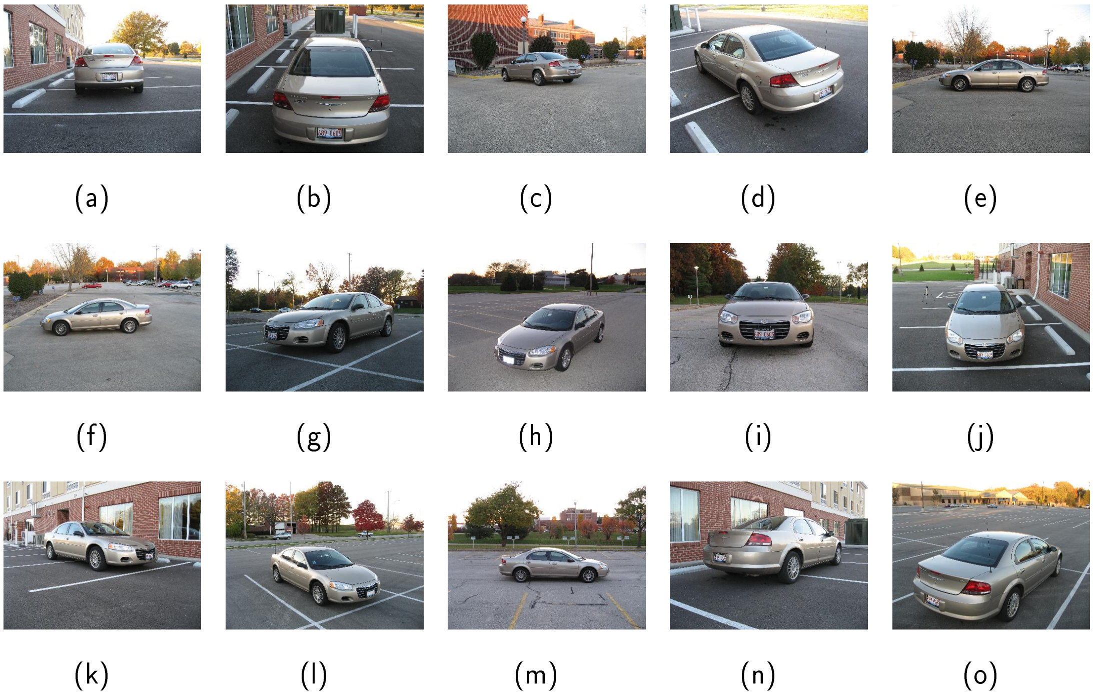


Figure 1.2: A variety of different viewpoints of one car of the 3D object category dataset by Savarese and Fei Fei [SFF07].

active object categorization, which gain more and more importance in computer vision (e.g. [JCC09, RP11]). In such systems, category and pose hypotheses are useful. On the one hand, a category or pose hypothesis can be verified by moving around the object and taking another view. On the other hand, by knowing the 3D nature of the object, the system knows how to move around the object to obtain a good view for the hypothesized category and pose. This is not only useful in active object categorization systems, but also in multiple camera systems, where the camera system is static, but objects move around the cameras and have to be detected and tracked over a scene. Also, when thinking about scene understanding - including Structure-from-Motion of static scenes (e.g. [SSP08]), Multi-body Structure-and-motion (e.g. [HP10]) (3D trajectories and motion pattern of moving objects) and interaction between objects - 3D models of objects and object categories are also of major importance.

Summing up, we can group current object categorization systems in several classes. Figure 1.3 gives an overview about different possibilities. 2D input images or synthetic 3D models e.g. CAD models are the basis for a categorization system. Either, a category model is learned directly from 2D input images or there is a reconstruction step before. The category model itself - 2D or 3D - is either learned from 2D images or 3D models or by a combination of 2D appearance and 3D geometry. For the detection there are several approaches: 3D category model - 3D model, 3D(2D) category model - 2D image, and 2D category model - 2D image. In this thesis, I follow the way emphasized in green: reconstructing 3D models from 2D image sequences, learning a 3D category model from them, categorizing new 3D models, and estimating the pose of objects in 2D images based on a 3D category model.

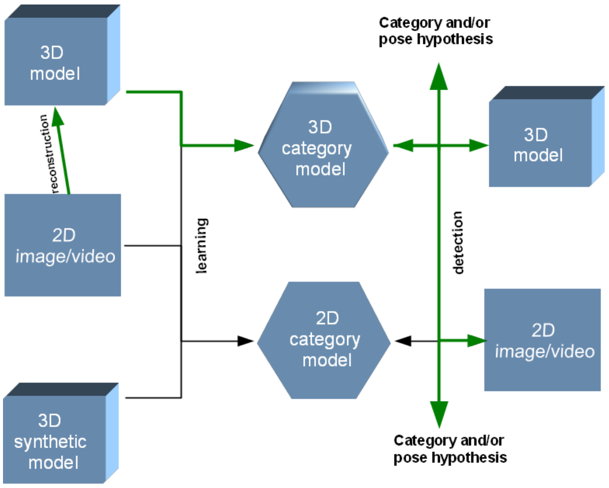


Figure 1.3: Overview of several ways to realize a categorization system. In my thesis, I follow the path emphasized in green.

Motivated by the idea to use 3D models for object categorization and based on the success of 2D category models based on shape information (see Section 2.1) I face the question: *Can we extend 2D shape models per aspect to one single, pose-invariant 3D model per category?*

1.2 Problem and Contribution

In Section 1.1, I motivate the use of 3D models for object categorization instead of modeling each 2D aspect separately, without going into detail on how I build such a 3D category model. This section starts with a short problem statement and continues with the main idea, my contributions as well as benefits of my research.

1.2.1 Problem Statement

My thesis investigates the problem of 3D object categorization based on the question: Can we extend the idea of 2D shape-based category models per aspect towards one single, pose-invariant 3D shape model per category in order to be invariant to pose and view changes? Specifically, the thesis addresses the following three main aspects:

- *(How) can we model 3D shape for specific objects?*
- *(How) can we learn one single, pose-invariant 3D category model based on shape information?*
- *(How) can we use such a 3D category model for pose estimation in 2D images?*

1.2.2 My Contributions

As mentioned before, my intention is to build a 3D category model based on shape information. I will motivate this decision. Most of the existing object categorization systems based on 3D models are appearance-based (see Section 2.2). However, categorization systems based on 2D shape models are very successful (see Section 2.1). In these 2D shape models, 2D contour fragments of silhouette and inner contour fragments (texture and shape information) and their spatial relations are learned. These systems are based on the fact that humans are able to categorize objects just on the basis of simple line drawings or collections of contour or line segments. In contrast to appearance-based systems, which are built on interest points and patches on them, shape is always visible and does not depend on texture information e.g. a white cup can be categorized as cup on the basis of its silhouette, whereas no interest points can be found on it. Sure it can be argued that in some cases, texture information is very important to distinguish between objects, e.g. horses and zebras, and I agree that appearance information is important. Nevertheless, shape information should not be neglected and its importance for object categorization should be analyzed. My thesis investigates the possibility to categorize 3D objects on the basis of 3D shape information and discusses benefits of the approach as well as open questions, problems, and possible solutions.

In Chapter 4, I describe how to model 3D shape of specific objects by 3D contour fragments reconstructed from stereo image sequences. A reader may expect that when reconstructing 2D contour fragments in 3D, the resulting model should consists of 3D surface patches, but this is not the only option. Indeed, our 3D shape model - we call it '3D contour cloud' - consists of a set of 3D curves - 1D embedded in 3D - of silhouette and inner contour fragments. In this thesis I will use the term '*3D contour fragment*' for a 1D manifold in 3D. The reason for this decision is that I want to model 3D shape by features, which can be used later on 2D images (pose hypothesis). Contours can be observed in both, 2D and 3D whereas there are no surfaces in 2D images. I present a reconstruction framework based on stereo image sequences, which combines a stereo reconstruction based on 2D contour fragments and motion estimation on interest points.

In Chapter 5, I present the 3D object categorization system based on a probabilistic framework. I model 3D contour fragments by Gaussian Mixture Models (GMMs). The representation by GMMs is very robust. On the one hand, outliers and noise can be handled. On the other hand, the 3D geometry of 3D contour fragments can be maintained. In my model, I learn partitions (discriminative subsets) of mixture components by introducing a similarity measure based on intra-class variability. I show the possibility to distinguish between categories with small inter-class difference and large intra-class variability and demonstrate translation and scale invariance of the model.

In Chapter 6, I introduce a new pose estimation algorithm, which makes use of the probabilistic 3D category model. By computing 2D aspect models of the 3D category model and by generating Gaussian Mixture Models for 2D contour fragments in the input image, a similar principle and similarity measure as for the categorization in 3D can be used. I demonstrate pose estimation for a large number of different poses on two well-known datasets: ETH-80 [LS03] and 3D object category dataset [SFF07]. In contrast to many existing pose estimation algorithms, the learning itself is not based on the same datasets.

For the task of object reconstruction and categorization I also developed a new dataset - GRAZ-STEREO-BASE-xx - of stereo image sequences. It is used for reconstruction of 3D shape models, for learning and categorization based on 3D shape models and for pose estimation based on 3D category models.

In summary, our results demonstrate that 3D shape information can be used for categorization of 3D objects and for 3D pose estimation of objects in 2D images.

During the research for this thesis the following papers were published. All of these papers are related to the topic of this thesis:

[PP11a] Kerstin Pötsch and Axel Pinz. 3D Object Categorization with Probabilistic Contour Models - Gaussian Mixture Models for 3D Shape Representation. In *Proc. VISAPP*, 2011.

[PP11b] Kerstin Pötsch and Axel Pinz. 3D Geometric Shape Modeling by '3D Contour Cloud' Reconstruction from Stereo Videos. In *Proc. CVWW*, 2011.

[PP11c] Kerstin Pötsch and Axel Pinz. 3D Object Category Pose from 2D Images using Probabilistic 3D Contour Models. In *Proc. ÖAGM*, 2011.

2
State-of-the-Art

Object categorization based on 3D models gains more and more importance in computer vision. The underlying literature is extensive. Most of the research in 3D model based object categorization goes towards appearance-based models, whereas the present work is based on shape information. Therefore, this chapter is grouped in two different sections - Section 2.1 gives an overview of 2D shape-based categorization systems, whose success motivates the idea to build a 3D shape model based on 3D contour fragments. Afterwards, in Section 2.2, existing categorization and pose estimation systems based on multi-view and 3D models are discussed and differences to this thesis are pointed out.

2.1 2D Shape Models for Object Categorization

Object categorization systems based on 2D shape models are quite successful in categorizing objects. Such systems are based on the idea that humans can detect objects on the basis of simple line drawings or collections of contour or line segments. Psychological experiments like [DWW04] have shown that contour information is sufficient to recognize objects. Lowe and Binford [LB83] show the importance of perceptual organization for recognition. Figure 2.1 shows their well known bicycle example. In [CSD$^+$09] Cole et al. investigate in a study how well line drawings (i.e. contour information) depict 3D shape. They point out that good line drawings depict shape nearly as well as shaded images for many objects.

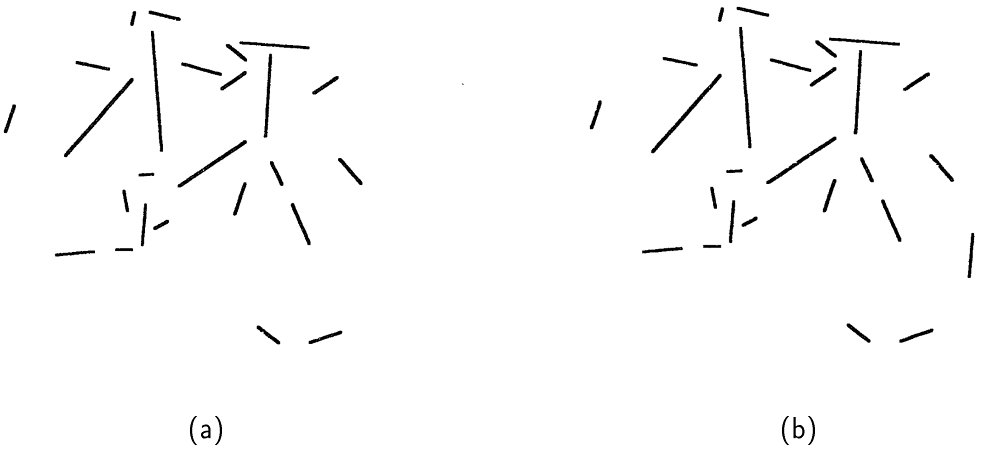
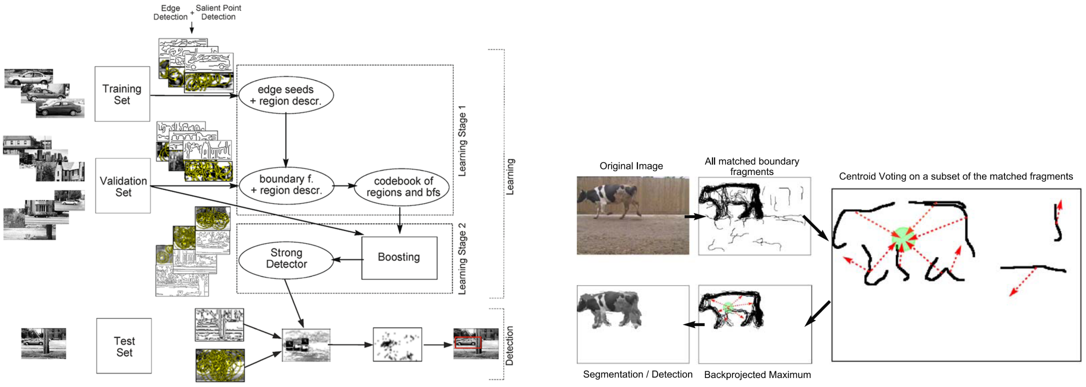


Figure 2.1: Bicycle recognition example of Lowe and Binford [LB83]: (a) Line drawing, which is hard to recognize. (b) One segment is added whereby the line drawing is easier to recognize.

Several approaches for object categorization using 2D shape models have been presented, e.g. in [OPZ06, OPZ08, SBC08, LHS07, FFJ$^+$08]. In 2006, Opelt et al. [OPZ06, OPZ08] proposed the Boundary Fragment Model (BFM), where they use 2D contour fragments of inner and outer (silhouette) contours, so called 'boundary fragments', and geometric information about the object's centroid in an Implicit Shape Model [LS04] way. The category model is learned in two steps (see Figure 2.2(a)) from training images and validation images. First, a discriminant codebook of 2D contour fragments together with their object centroid votes is learned from training images. Second, a strong classifier is built from weak hypotheses using Adaboost on the validation set. A weak hypothesis consists of constellations of size k of 2D contour fragments of the codebook (typically $k = 2$). The BFM is a supervised method as it requires 2D bounding boxes around the objects during training and the validation sets need to contain labels and centroids of the objects. The detection of an object in a new input image using a voting space is demonstrated in Figure 2.2(b). In [SBC05, SBC08] Shotton et al. present a similar categorization system using a 2D shape model based on contour fragments. However, they use a different Boosting technique, where the centroid prediction and the weak detectors differ slightly.



(a) (b)

Figure 2.2: Overview of the Boundary Fragment Model [OPZ06]. (a) 2 learning steps of a BFM. (b) Detection of objects in a new input image .

A completely different approach based on pairwise relations between contour fragments is proposed by Leordeanu et al. in [LHS07]. They build a structural representation of objects influenced by ideas of Hummel [Hum00], who pointed out the benefits of structural descriptions compared to view-based representations. Furthermore, he pointed out that humans make other decisions with regard to similarity than view-based recognition systems. This fact depends on the observation that humans use categorical spatial relations among object parts for measuring similarity. Consequently, view-based methods have limitations in object recognition compared to structural representations. Leordeanu et al. show in their 2D based categorization method that pairwise relations between contour fragments are sufficient to represent the shape of the object. They sample contour points on 2D contour fragments and model geometric pairwise relations between these points. Thereby, they build a clique where each sampled contour fragment point is related with each other point. The relation is described by a vector of seven elements $e = (\theta_i, \theta_j, \sigma_{ij}, \sigma_{ji}, \alpha_{ij}, \beta_{ij}, d_{ij})$ (see Figure 2.3). Let (i, j) be a contour fragment pair. Then d_{ij} describes the distance between i and j, β_{ij} the angle between their normals, θ_i and θ_j the angles between the normals and the x-axis, σ_{ij} and σ_{ji} the angles between the normals and the distance and α_{ij} the angle between the x-axis and the distance. During learning, a spectral correspondence algorithm is used [LH05].

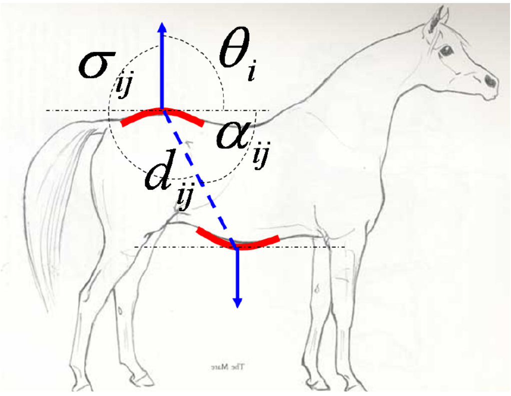


Figure 2.3: Illustration ([LH05]) of the geometric relations (distances and angles) between contour points used in the 2D shape model.

In [FFJ$^+$08] Ferrari et al. propose an object categorization system based on adjacent contour segments, which they call k adjacent segments, short kAS. Such kAS are groups of connected, more or less straight lines, thus connectedness is an important property. They use a scale and translation invariant descriptor and kAS are matched to an image using a contour segmentation network. To detect object classes, a codebook of kAS features is learned with a clique-partitioning clustering and a sliding window approach for detection.

Summing up, all the mentioned 2D shape models for object categorization are based on 2D contour information in combination with spatial relations. Spatial relations between fragments are described in different ways by using the object's centroid, pairwise geometric relations or connectedness information.

2.2 Multi-view and 3D Models for Object Categorization

In the last years, object recognition and categorization systems based on multi-view and 3D models gain more and more importance in computer vision research e.g. [RLSP06b, RLSP06a, TFL$^+$06, SFF07, SFF08, SSSFF09, SSFFS09, LSS08, LS10, SGS10, ANB09, YKS07].

3D modeling has successfully been used in some specific object recognition methods e.g. [RLSP06b, RLSP06a, DPP09]. To name just a few examples, Rothganger et al. [RLSP06b, RLSP06a] have proposed a specific 3D object representation, where they model the spatial relationships between surface patches. Detry et al. [DPP09] have presented a probabilistic framework for 3D object modeling based on spatial relations between 3D contour features in a hierarchical model. Based on this 3D representation they are able to recognize the modeled object in an unknown scene as well as to estimate its pose in 3D.

With respect to object categorization, several systems have been suggested, e.g. in [SFF07, SFF08, SSSFF09, SSFFS09, LSS08, LS10, SGS10]. Most of them have in common that they are appearance-based. However, they are realized in completely different ways. They can be divided in 3D categorization methods with 3D pose estimation e.g. [SFF07, SFF08, LS10, SSFFS09] and without 3D pose estimation e.g. [TFL$^+$06, YKS07] or they can be divided on the basis of the multi-view and 3D information. I prefer a grouping based on how geometric multi-view or 3D information is obtained: multi-view and 3D information can be obtained from images e.g. [TFL$^+$06, SFF07, SFF08, SSSFF09, SSFFS09, ANB09, YKS07] or from synthetic 3D models e.g. [LSS08, LS10, SGS10].

In the first group of object categorization system, the multi-view or 3D information is obtained from images (including pose annotations) [HRW07, KSP07, CKLP07, TFL$^+$06, SFF07, SFF08, SSSFF09, SSFFS09, ANB09]. In [TFL$^+$06] Thomas et al. propose a multi-view object detection system, where they combine the Implicit Shape Model of Leibe and Schiele [LS04] and the multi-view specific object recognition system proposed by Ferrari et al. [FTG04]. In their model, single-view codebook entries are linked together among multiple views with so-called activation links. These activation links are then used in recognition to transfer votes between views.

Extensions of this work are presented by Savarese and Fei Fei in [SFF07, SFF08]. For their 3D model based object categorization system they combine 2D appearance information and 3D geometric information avoiding 3D reconstruction. 2D appearance information is represented by so called 'canonical parts', which are regions containing a set of image patches seen in several instances of an object category. So, one of these parts represents more or less one particular view. Two canonical parts are geometrically linked together when they are visible at the same time e.g. when one sees the front of a car but also parts of the side view. The linkage is described by a homography containing an affine transformation and a translation vector. Thereby, a connected graph of canonical parts is built during the learning stage. When a new input image is categorized, first features are extracted and canonical part candidates are built. Then, these candidates are matched to the category model using a global optimization. For their experiments they build a new 3D object category dataset (see Section 3.1.1). In [SFF08] Savarese and Fei Fei extend their model using view synthesis such that they are able to also recognize poses, which have not been seen during the learning stage. For this they use a view morphing technique to generate canonical parts for novel views on category level. The categorization and pose estimation is done using a two-step algorithm. First, a new image is tested on the category model and the best model views are computed. Second, view synthesis is computed for a canonical view of the list with its nearest canonical poses on the viewing sphere.

A multi-view approach is proposed by Yan et al. in [YKS07]. They reconstruct a 3D model from multiple 2D images, so called model views, for one specific object, using an approach based on homography. Then, 2D features of additional training images of the category are computed using SIFT and attached to the 3D model. Thus, they build a 3D feature model by combining multiple views of one specific object and arbitrary views of several additional objects, so-called supplemental views [KYS07].

In [SSSFF09] Sun et al. propose a generative probabilistic framework for object categorization based on a multi-view part-based model. Their model consists of parts, which are linked together using geometric constraints. The part-based representation is obtained by computing a part type assignment, a viewpoint assignment, as well as a patch appearance distribution and a patch location distribution for each feature patch. Thereby, they model appearance information and location information over several views. For consistency under nearby viewpoints, they also include epipolar constraints. Based on this part-based generative category model, Su et al. propose a categorization system including view synthesis in [SSFFS09]. Their categorization system can categorize objects and estimate their pose even when the view was not seen during training. For the initial model they build a dense multi-view representation of the viewing sphere without any labeled pose information. They build a triangle mesh where each new view can be generated by using a homography and an interpolating morphing parameter. Similar to [SSSFF09] they compute location, appearance and proportion parameters for each part. Instead of epipolar constraints, they compute geometric constraints within one triangle and across several triangles using affine transformations. This 3D model is then updated with a set of unsorted, unlabeled training images using an incremental learning approach.

In [ANB09] Arie-Nachimson and Basri propose a 3D category model for rigid objects by building a 3D Implicit Shape Model combining 2D appearance information and 3D location information. The voting procedure is based on that of a 2D Implicit Shape Model but modified such that a transformation parameter and a visibility parameter are included. Their 3D reconstruction approach is based on a factorization method. To construct a 3D class model they match training images to the initial model by comparing them to those images, which were used for the 3D model. The category model is particularly constructed for the task of pose estimation of cars.

In the second group of object categorization systems, synthetic 3D models as e.g. computer aided design (CAD) models are used to obtain geometric 3D information [LSS08, LS10, SGS10]. This geometric 3D information is then used in combination with 2D categorization systems.

In [LSS08] Liebelt et al. propose a multi-view object categorization method based on 3D feature maps consisting of 2D appearance and 3D position information built from CAD models. Views are rendered from these 3D CAD models over the upper hemisphere and features are extracted from rendered views of these CAD models. For background features an additional dataset of real 2D images is used. The category model is learned using a two-class support vector machine.

In [LS10] Liebelt and Schmid present a method where they combine a 2D appearance-based part model with 3D geometry from CAD models. They build a 2D part detector per view by dividing the bounding box in a regular grid and learn a spatial pyramid detector. For the 3D geometry, one or more CAD models are rendered and the rendered images are divided in the same grid as in the 2D learning. For each part a Gaussian Mixture Model of its 3D point cloud representation is learned. In the detection process, they combine a 2D detection based on the 2D model and a pose estimation based on the 3D Gaussian Mixture Models.

The approach by Stark et al. [SGS10] is solely based on 3D CAD models. They use non-photorealistic rendering and build a part based approach of thirteen parts (e.g. wheels, doors, rear window of a car), representing them by edges. Furthermore, they use a probabilistic model influenced by the idea of constellation models for the spatial layout of these parts.

Summing up, most of the mentioned state-of-the-art research on 3D model based object categorization use appearance-based methods. The research varies from combining 2D models with 3D synthetic models to reconstructing 3D models from images and attaching 2D patches to them or to combine 2D models with

3D information. All these approaches use some kind of patches. So far, shape information played a minor role.

In contrast to this, my aim is object categorization based on a 3D shape model. Our approach can be assigned to the first group. Consequently, all information for our 3D shape model is computed from stereo image sequences - no synthetic models are used. Thereby, we do not avoid the reconstruction as e.g. in [SFF07, SFF08], but we reconstruct a model as in [YKS07] without attaching additional 2D information. In contrast to [ANB09], our 3D model is reconstructed fully automatically. We reconstruct one 3D model per training object and learn a category model in 3D. This 3D category model can be used for both, categorization of 3D models, and pose estimation in still 2D input images.

3
Datasets

For the evaluation of their own experiments and the comparison to other state-of-the-art work, image and video datasets constitute an important tool for researchers in computer vision. Datasets for the tasks of recognition and categorization are manifold. However, for the task of multi-view categorization, where objects of different categories are captured from different poses, the number of datasets is limited. Especially, up to now there exists no dataset for categorization which consists of a set of specific objects of several categories suitable for 3D reconstruction. Therefore, I have built a new dataset - GRAZ-STEREO-BASE-xx - of stereo image sequences for the task of reconstruction and categorization. The dataset includes stereo image sequences, calibration data, 2D contour fragments as well as estimated camera poses for objects of five categories captured under different conditions. Our GRAZ-STEREO-BASE-xx dataset, which consists of the GRAZ-STEREO-BASE-EYE, GRAZ-STEREO-BASE-30, and GRAZ-STEREO-BASE-EYE-TURNTABLE dataset, is described in Section 3.2. Additionally, the chapter provides an overview of existing datasets for the task of categorization from multiple viewpoints in Section 3.1. Both our new dataset and the previously available datasets are used in the subsequent chapters for comparison and evaluation of the object categorization method and the pose estimation approach.

3.1 Existing Multi-view Datasets

There exists a large number of datasets for the task of object categorization e.g. Pascal, CALTECH, GRAZ-01 or GRAZ-02. These datasets differ - among other things - in the type and number of object categories, background clutter, scale, inter-class difference, intra-class variability and viewpoints. However, for the task of multi-view object categorization only a small number of datasets is available, where objects have been captured under several viewpoints and where ground truth information is available (pose, scale, bounding box, etc.). We picked out two of them: ETH-80 [LS03] and 3D object category dataset [SFF07] for our experiments. Therefore, these datasets are described below.

3.1.1 3D Object Category Dataset

Savarese and Fei Fei [SFF07] have built a 3D object category dataset for the task of object categorization based on multi-view and 3D models (see Figure 3.1 for example images). The dataset consists of ten different categories: bicycle, car, cellphone, head, iron, monitor, mouse, shoe, stapler, and toaster and contains around 7000 color images captured under different conditions: eight viewing angles, two or three heights and three scales. In addition, also the ground truth is provided as a binary mask.

3.1.2 ETH-80

For the task of object categorization, Leibe and Schiele have built the ETH-80 dataset [LS03] (see Figure 3.2). It contains 80 objects of eight categories: apples, pears, tomatoes, cows, dogs, horses, cups, and cars. The dataset consists of 41 equally distributed (22.5^o) views over the upper hemisphere of each object. The ground truth of each color image is included as a segmentation mask. Additionally, the dataset contains the objects contour.

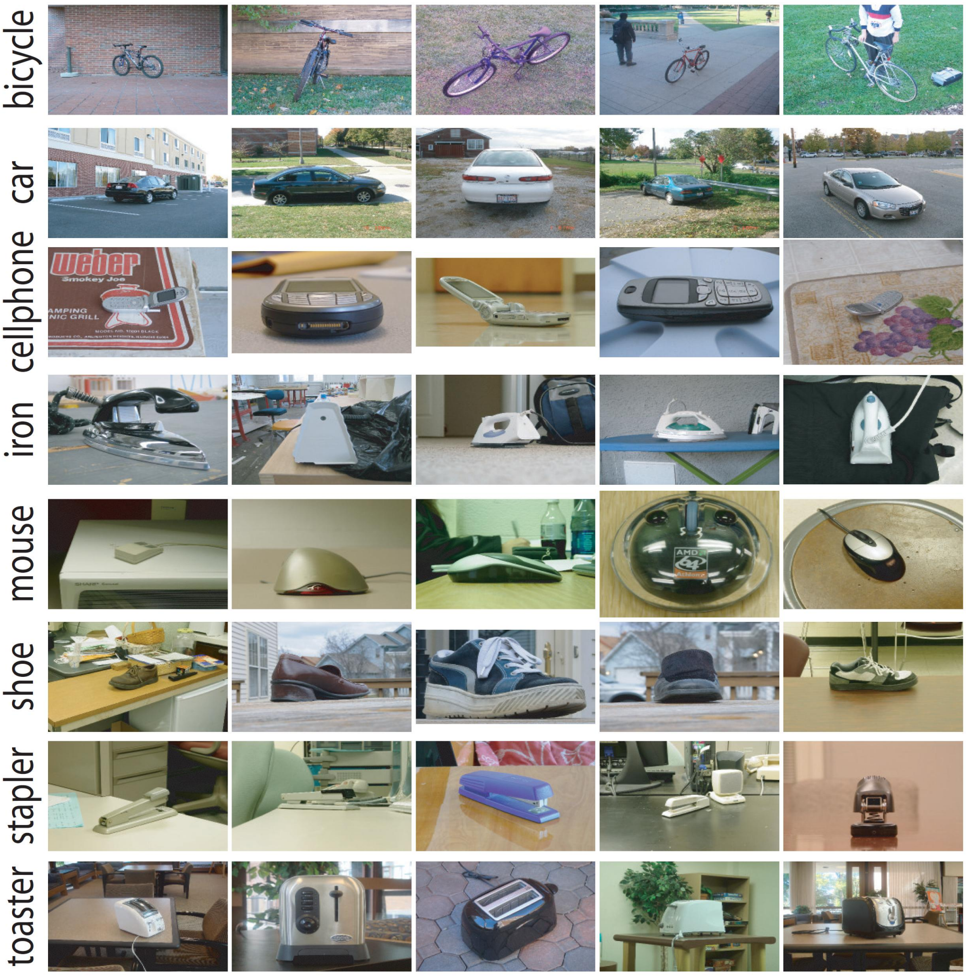


Figure 3.1: Sample images from eight of the ten categories of the 3D object category dataset [SFF07].

3.2 Our GRAZ-STEREO-BASE-xx Dataset

We have built a new dataset of stereo image sequences, which show objects of several categories from different aspects: GRAZ-STEREO-BASE-EYE, GRAZ-STEREO-BASE-30, and GRAZ-STEREO-BASE-EYE-TURNTABLE. The dataset is built using two types of stereo setups, which differ in the length of their baseline as well as in their focal length and their vergence angle. In addition to the stereo image sequences, we also provide camera calibration data, estimated camera poses and 2D contour fragments with subpixel accuracy. The dataset

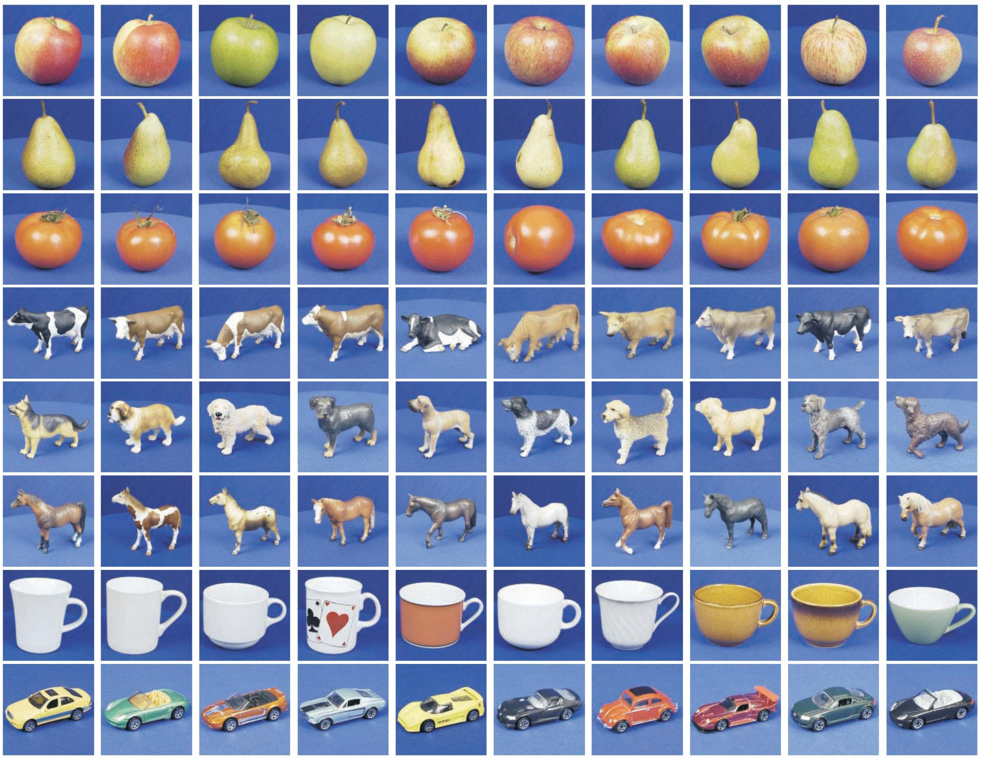


Figure 3.2: Sample image of each object of the ETH-80 dataset [LS03].

can be used for several tasks in computer vision as stereo reconstruction, motion estimation, multi-view object recognition and categorization as well as 3D model based recognition and object categorization.

3.2.1 Framework

Our dataset is built using three different settings and two different types of stereo rigs: STEREO-RIG-EYE and STERE-RIG-30. The dataset GRAZ-STEREO-BASE-EYE contains stereo image sequences of handheld objects captured using the STEREO-RIG-EYE. The GRAZ-STEREO-BASE-30 consists of stereo image sequences in an outdoor environment captured using the STEREO-RIG-30. The third dataset, GRAZ-STEREO-BASE-EYE-TURNTABLE contains stereo image sequences of objects on a turntable captured using the STEREO-RIG-EYE.

Our stereo rigs (STEREO-RIG-EYE (see Figure 3.3(a)) and STEREO-RIG-30 see Figure 3.3(b))) consist of two μEye 1220C cameras and Cosmicar/Pentax

Table 3.1: Stereo rigs parameters

	baseline (cm)	**focal length** (mm)	**vergence angle** (^{o})
STEREO-RIG-EYE	6	12.5	5.5
STEREO-RIG-30	30	6.5	6.5

lenses. The detailed parameters can be found in Table 3.1. The frame rate is 15 Hz. The size of the images is 480x752 px. The field of view is 40.5^o (21.5^o) for the focal length of $6.5mm$ ($12.5mm$). For the calibration of the stereo rig we use the Camera Calibration Toolbox for Matlab [Bou]. To capture the GRAZ-STEREO-BASE-EYE-TURNTABLE dataset we additionally use a turntable (see Figure 3.3(c)).

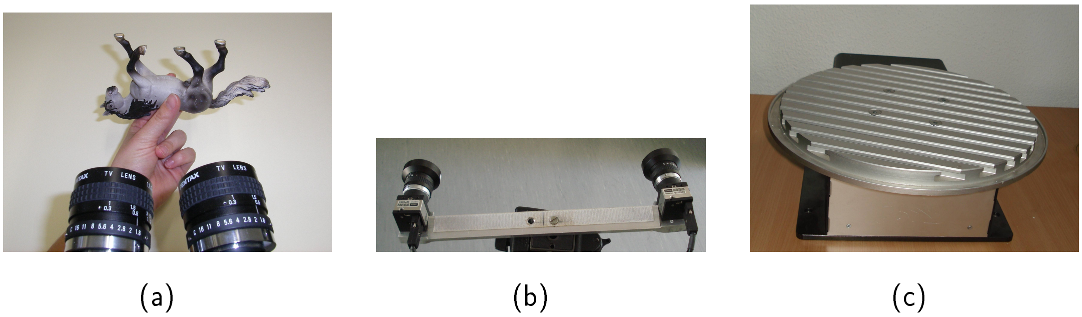

(a) (b) (c)

Figure 3.3: Stereo rigs and turntable, which were used to capture our dataset. (a) STEREO-RIG-EYE (baseline: human eye distance). (b) STEREO-RIG-30. (c) Turntable.

3.2.2 GRAZ-STEREO-BASE-EYE

For the GRAZ-STEREO-BASE-EYE dataset we have captured several stereo image sequences of small toy objects, which are manipulated in front of the STEREO-RIG-EYE in the lab. This dataset contains two categories: horses and cows, with nine horses and five cows. Each of these stereo image sequences typically shows one object, which is presented in a hand-held manner in front of a homogeneous background. In this dataset, we avoid the controlled setting of a turntable. The object is manipulated naturally by hand in front of the stereo rig, so that it is seen pretty much from all sides, showing as many aspects as possible. Figure 3.4 shows an overview of the objects in the dataset.

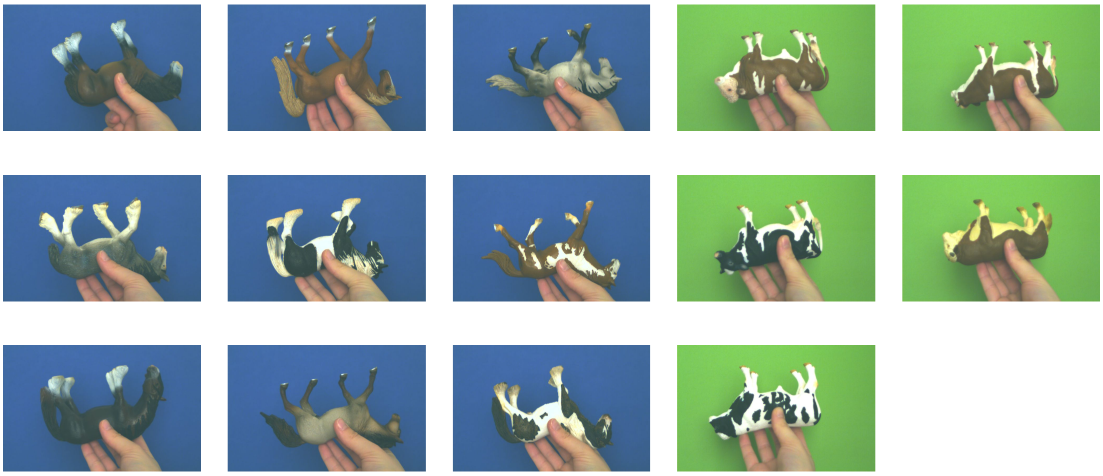


Figure 3.4: Overview of the objects captured for the GRAZ-STEREO-BASE-EYE: 2 categories, 14 objects of the categories horse and cow.

One stereo image sequence typically contains around 500 to 600 RGB stereo image pairs. The image sequences were captured in Bayer format. To improve the color quality of the images, we use the color calibration algorithm by Wolf [Wol03]. For this, we also captured an image of a color target and calibrate the values with regard to these reference values. Figure 3.5 shows example stereo views for one of the horse sequences. In addition to the image sequence, we also provide the calibration data, 2D contour fragments and the absolute camera poses per frame. 2D contour fragments are computed using the Canny edge detector with subpixel accuracy (using linear interpolation) and a linking algorithm based on Peter Kovesi's [Kov]. Afterwards, we apply a clockwise ordering.

In order to reconstruct just contours of the hand-held objects and not contours of the hand we apply an interactive segmentation system based on variational methods (see [UMPB09]), which gives us a precise hand segmentation. Coarse manual labels have to be provided for the initial frames of each stereo image sequence while the remaining frames are processed automatically, providing excellent ground truth for hand segmentation (see Figure 3.6). Features that belong to the hand are subsequently ignored.

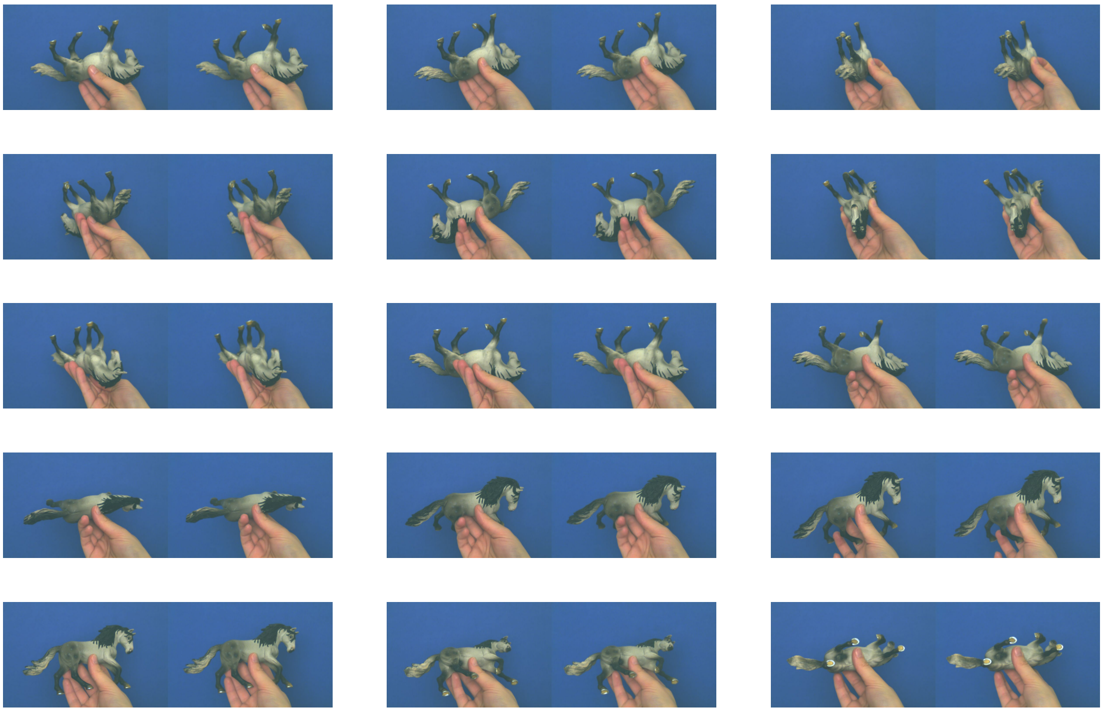


Figure 3.5: Example views from a stereo image sequence of the GRAZ-STEREO-BASE-EYE, where a horse was manipulated by hand in front of the stereo rig.

The motion analysis is based on the approach by Schweighofer et al. [SSP08]. The system is able to reconstruct structure and motion of stationary scenes and it is robust if there are at least 50% of the features in the stationary scene, foreground motion is detected as outliers. In our case, because the majority of the interest points is located on a rigid object, the system assumes that the stereo rig is moving around the object although we manipulate the object in front of the cameras. This leads to an ‘object-centered’ representation as shown in Figure 3.7 for the horse sequence above.

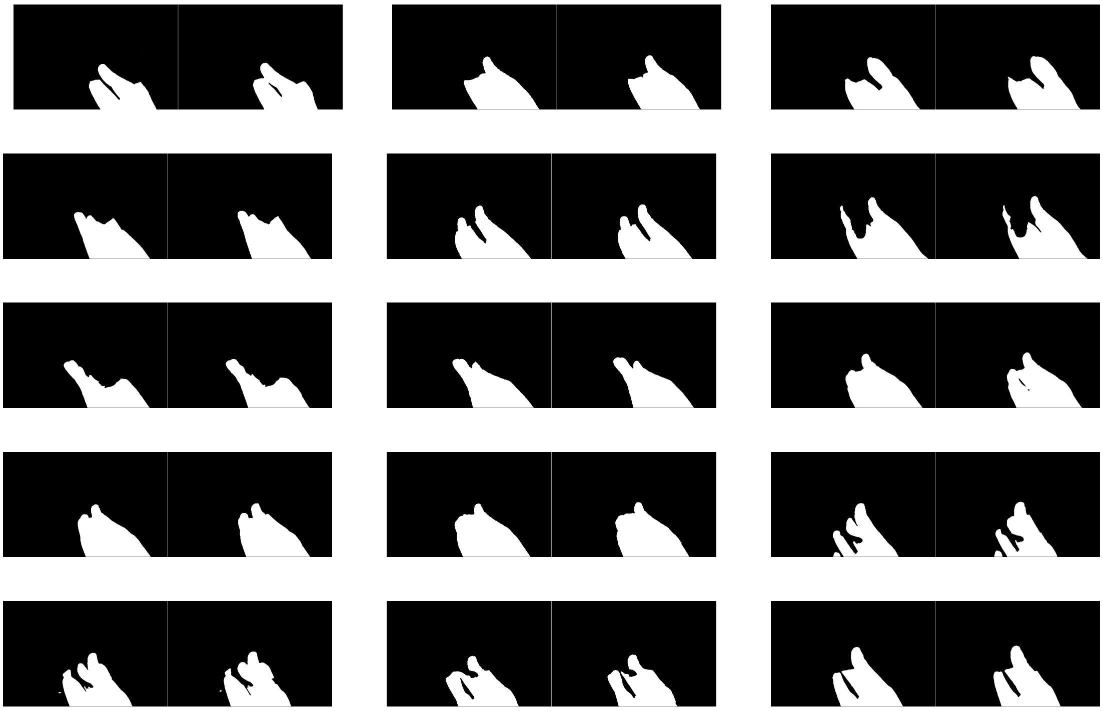


Figure 3.6: Mask of skin color using [UMPB09] for the example views shown in Figure 3.5.

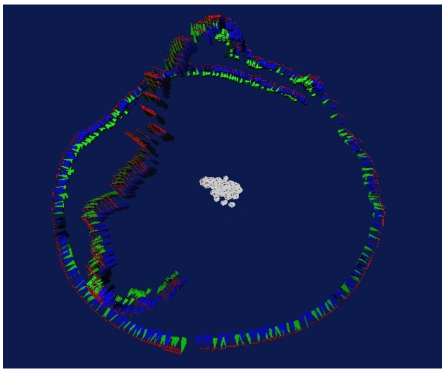


Figure 3.7: Estimated absolute camera poses (R,t) of the stereo rig around the horse using [SSP08] represented in an object-centered coordinate system. Each camera pose is drawn as a stereo rig represented by a green and a blue triangle connected by a red line. In the middle of the trajectory the point cloud of the horse is shown in grey.

3.2.3 GRAZ-STEREO-BASE-30

The GRAZ-STEREO-BASE-30 contains stereo image sequences of five humans who rotate around their vertical axis in front of the STEREO-RIG-30. One stereo image sequence typically contains around 200-300 grey value stereo image pairs, each image has a size of 480x572 px. The stereo image sequences were captured outdoors in front of a white wall. Figure 3.8 shows an overview of the humans in the dataset, Figure 3.9 shows example views for one of the video sequences. In addition to the image sequence, we also provide the calibration data, 2D contour fragments and camera poses. The motion analysis, the computation of 2D contour fragments and the calibration is done in the same way as for the GRAZ-STEREO-BASE-EYE dataset.

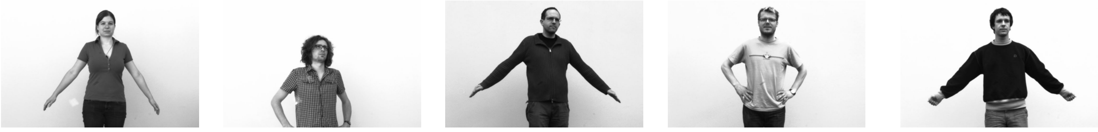


Figure 3.8: Overview of the objects captured for the GRAZ-STEREO-BASE-30: 5 objects of the category human.

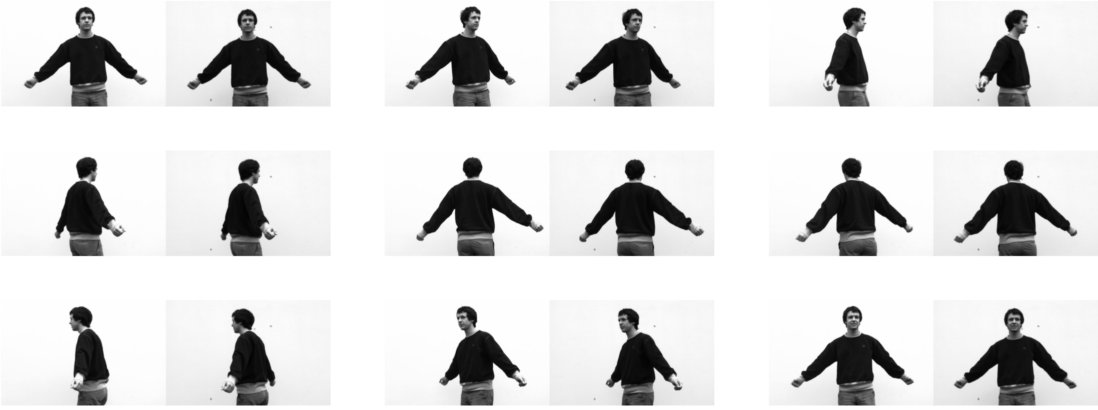


Figure 3.9: Example views from a stereo image sequence of the GRAZ-STEREO-BASE-30, where humans rotate around their vertical axis in front of the stereo rig.

3.2.4 GRAZ-STEREO-BASE-EYE-TURNTABLE

For the GRAZ-STEREO-BASE-EYE-TURNTABLE dataset we have captured several stereo image sequences of small toy objects, which are rotated on a

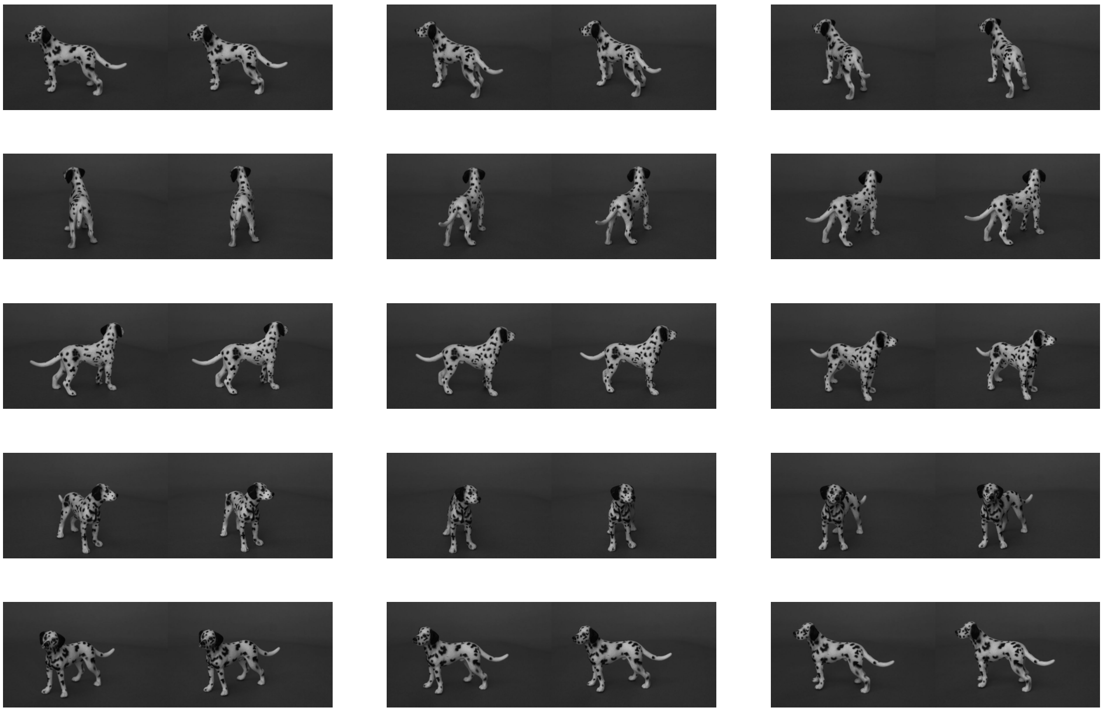


Figure 3.10: Example stereo pairs from a stereo sequence one rotation on the turntable of a dog. Example stereo pairs from an image sequence of the GRAZ-STEREO-BASE-EYE-TURNTABLE, where one rotation on the turntable of a dog is recorded.

turntable in front of the STEREO-RIG-EYE in the lab. This dataset contains 35 objects of four categories: horses, cows, dogs, and cars with ten horses, eight cows, nine dogs, and eight cars (see Figure 3.11). Each stereo image sequence typically shows one object, which is presented on a turntable in front of a homogeneous background. The object is rotated around its vertical axis on the turntable for around 360^o. Figure 3.11 shows an overview of the objects contained in the dataset.

One stereo image sequence typically contains around 600 to 700 grey value stereo image pairs. Figure 3.10 shows example views for one of the dog sequences. In addition to the image sequence, we also provide the calibration data, 2D contour fragments and camera poses. Again, the motion analysis uses the image based approach by Schweighofer et al. [SSP08], no position sensor information from the turntable is used.

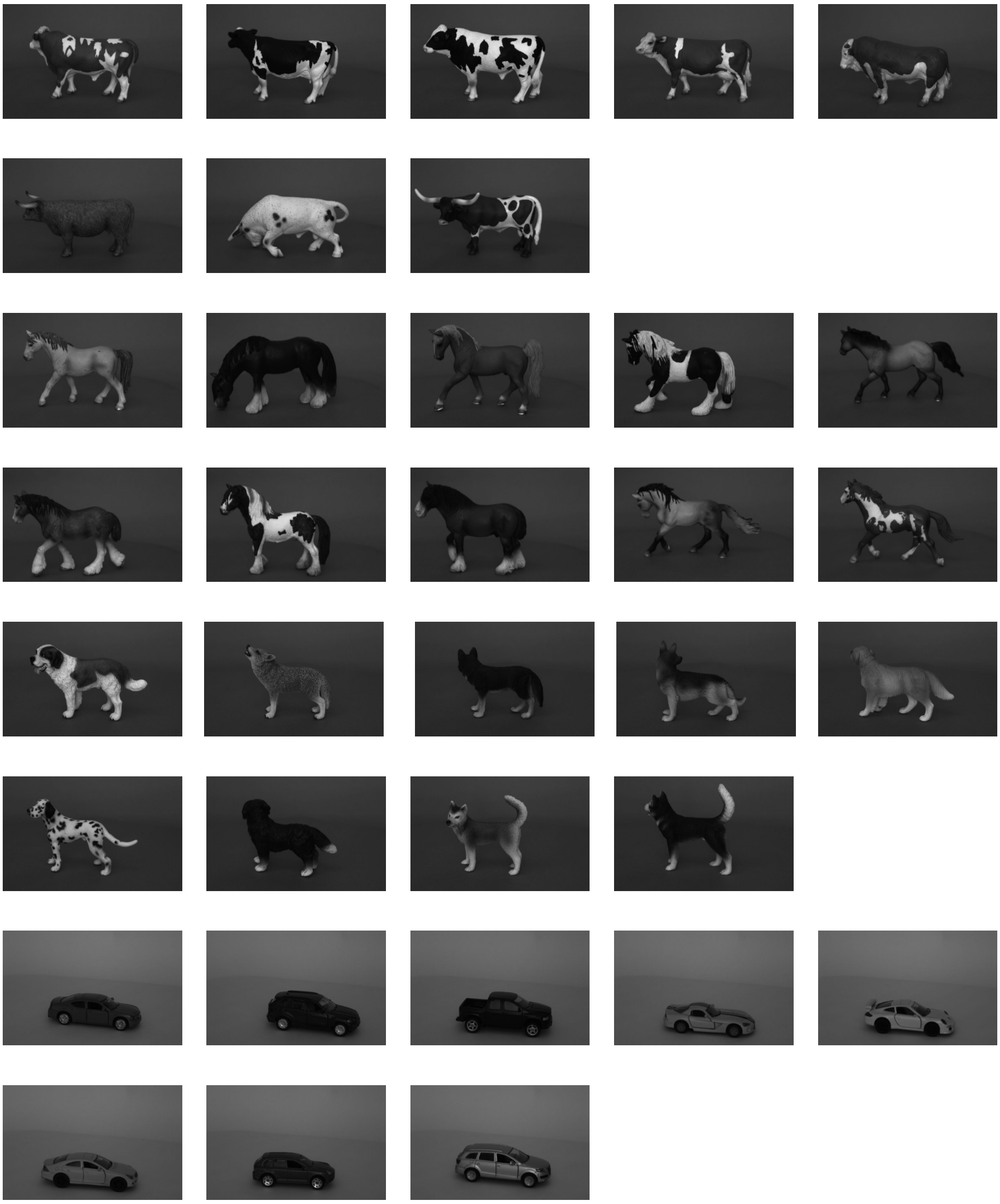


Figure 3.11: Overview of the objects captured for the GRAZ-STEREO-BASE-EYE-TURNTABLE: 4 categories, 35 objects of the categories cow, horse, dog and car.

4
3D Geometric Shape Modeling by 3D Contour Cloud Reconstruction

(How) can we model 3D shape for specific objects? There are several methods in computer vision on how the 3D nature of specific objects can be modeled e.g. by means of 3D point clouds or 3D surface meshes. In this chapter, I present a method on how to model 3D shape for specific objects by so called '3D contour clouds'. On the one hand, a '3D contour cloud' is an extension of 3D point clouds towards 3D contour fragments instead of 3D points. On the other hand, it is an extension of 2D shape models based on silhouette and inner 2D contour fragments towards 3D.

In Section 4.1, I start with an introduction, followed by an overview of state-of-the-art research based on 3D contour fragments in Section 4.2. There, I concentrate on 3D curve representations. In Section 4.3, I introduce the stereo reconstruction framework and describe in detail the '3D contour cloud' reconstruction method from stereo image sequences. I introduce a novel stereo correspondence algorithm based on shape and stereo information and I propose the idea of 3D shape context for outlier reduction. The main part of Section 4.4 comprises reconstruction results on our GRAZ-STEREO-BASE-xx dataset as well as on a standard multi-view stereo dataset.

4.1 Introduction

Modeling shape plays an important role in computer vision and graphics and builds the basis for many applications e.g. in object categorization or 3D shape retrieval. Methods for generating 3D models give us the opportunity to model the 3D nature of objects and can thus provide us with additional information about shape and appearance of the objects.

3D point clouds, generated by laser range scanners, by stereo vision, or by Structure-from-Motion techniques, are probably the most obvious and simplest way to represent 3D shape. Often, these 3D point clouds are converted into triangle meshes or polygonal models. Extensive research has been done to generate, analyze, match, and classify such models. There exist many 3D model databases such as the Princeton Shape Benchmark [SMKF04], the McGill 3D Shape Benchmark [SZM$^+$08] or the ISDB [GSCO07a] for evaluating 3D shape retrieval algorithms.

In contrast to these 3D point-based methods, we aim to build 3D geometric shape models for objects using 3D contour fragments. In this thesis we use the term '3D contour cloud'. A '3D contour cloud' is a set of 3D contour fragments, which describe the 3D shape of an object. In our terminology a 3D contour fragment is a 1D manifold in 3D (curve in 3D), not a surface in 3D. These 3D contour fragments are silhouette contours as well as inner contours, which are reconstructed from stereo image sequences. For the remaining thesis we make the following definitions:

4.1 Definition

A '3D contour cloud' $\mathcal{C}$ is an unorganized set of N 3D contour fragments $\mathcal{F}_l$ in a three dimensional coordinate system:

$$\mathcal{C} = \{\mathcal{F}_l, l = 1...N\}, \tag{4.1}$$

4.2 Definition

A 3D contour fragment $\mathcal{F}_l$ is a 1D manifold in 3D and is defined by a set of 3D points p_i given by their X,Y,Z coordinates in an ordered list:

$$\mathcal{F}_l = \{[p_1, p_2, \ldots p_i \ldots] \mid p_i = (x_i, y_i, z_i) \in \mathbb{R}^3 \text{ and } p_i \text{ is neighbor of } p_{i+1}\} \quad (4.2)$$

4.2 State-of-the-Art

As mentioned before, there exist various 3D shape models based on 3D point clouds, meshes or polygonal models. The methods for describing such 3D shapes vary from shape distributions [GSCO07b, MS09, OMT05, OFCD] to symmetry descriptors [KFR04] or Skeletal Graphs [SSGD03]. I refer to Iyer et al. [IJL+05] and Tangelder and Veltkamp [TV07] for an overview of 3D shape representation methods.

So far, only a few methods exist, which represent 3D shape by 3D contour fragments. The research on 3D contours (1D manifolds embedded in 3D) concentrates on 3D curve reconstruction methods [EG07, PH02, FK10] and 3D contour extraction from existing 3D surface models [DFRS03, DR07, OBS04, PKG03].

With respect to reconstruction methods in this overview I concentrate on 3D contour fragment reconstruction, not on dense 3D models generated by silhouette reconstruction as e.g. presented in [Her04]. One reconstruction method based on the usage of a double stereo rig was presented by [EG07]. The authors describe a method for 3D reconstruction of object curves from different views as well as motion estimation based on these 3D curves. In [PH02], the authors propose a method for Euclidean contour reconstruction including self-calibration, but there the data is not very natural and their algorithm has problems in matching contour fragments. Recently, Fabbri and Kimia [FK10] presented an approach for multi-view stereo reconstruction and calibration of curves. In their paper, they concentrate on the reconstruction of 3D contour fragments without motion analysis and their algorithm is mainly based on so called view-stationary curves e.g. shadows, sharp ridges, reflectance curves. Therefore, it is well applicable for aerial images as they show in their results.

3D contour extraction methods are manifold and differ mainly in the type of the extracted contour/line fragments, e.g. occluding contours, ridges [OBS04], suggestive contours [DFRS03], suggestive highlights and principal highlights [DR07]. For a more detailed literature overview we refer to the mentioned publications.

4.3 3D Contour Cloud Stereo Reconstruction

For the reconstruction of a '3D contour cloud' from stereo image sequences, we use a framework, which combines the stereo reconstruction of 3D contour fragments from single stereo frame pairs and the motion estimation between consecutive frames based on the Structure-and-Motion approach by Schweighofer and Pinz [SP06] (see Section 4.3.1). The main novel idea in our reconstruction framework is the integration of geometric information in the concept of 2D shape context proposed by Belongie et al. [BMP02] (see Section 4.3.2) and the extension of 2D shape context towards a '3D shape context' for outlier reduction (see Section 4.3.3).

4.3.1 Stereo Reconstruction Framework

An overview of our stereo reconstruction framework can be seen in Figure 4.1. Our '3D contour cloud' generation is based on calibrated stereo image sequences of objects. For the task of stereo reconstruction and object categorization we have built the GRAZ-STEREO-BASE-xx dataset (see Section 3.2). For stereo image sequences, where objects are manipulated by hand in front of the stereo rig we start with a preprocessing step where we apply an interactive segmentation system based on variational methods (see [UMPB09]) to mask the hand. Features that belong to the hand are subsequently ignored. To compute 2D contour fragments we apply the Canny edge detection algorithm at subpixel accuracy in the left and the right frame of a stereo frame pair. Then, a linking algorithm [Kov] is used in order to obtain long, connected 2D contour fragments. For the reconstruction of 3D contour fragments we need to find corresponding 2D contour fragments and point correspondences on them. Here, we combine the well known

2D shape context and epipolar information into one single cost matrix $\mathbf{C}_{ij}$. The cost matrix $\mathbf{C}_{ij}$ is a weighted combination of the original 2D shape context cost matrix $\mathbf{CS}_{ij}$ and an epipolar constraint cost matrix $\mathbf{CE}_{ij}$. Based on the contour point correspondences we reconstruct the 3D contour fragments using the 'Object Space Error for General Camera Models' [SP06], which is based on the 'Object Space Error' [LHM00] as a cost function. For the absolute orientation - rotation, translation, scale (R, t, s) - between 3D contour fragment reconstructions of consecutive frames we estimate the motion between 3D reconstructions of points based on [SSP08]. To find corresponding 3D contour fragments in consecutive frames we first use correspondences of 2D fragments over time. For 2D correspondences over time we first reduce the search space by taking into account only those 2D contour fragments in the consecutive frame, which lie in a similar region as the 2D contour fragment in the current frame. Then we match those contour fragments using 2D shape context. Afterwards, we apply 3D shape context for the correspondence of 3D contour fragments and outlier reduction. For this, we compute a cost matrix $\mathbf{CF}_{ij}$ based on the 3D shape context cost between 3D contour fragments.

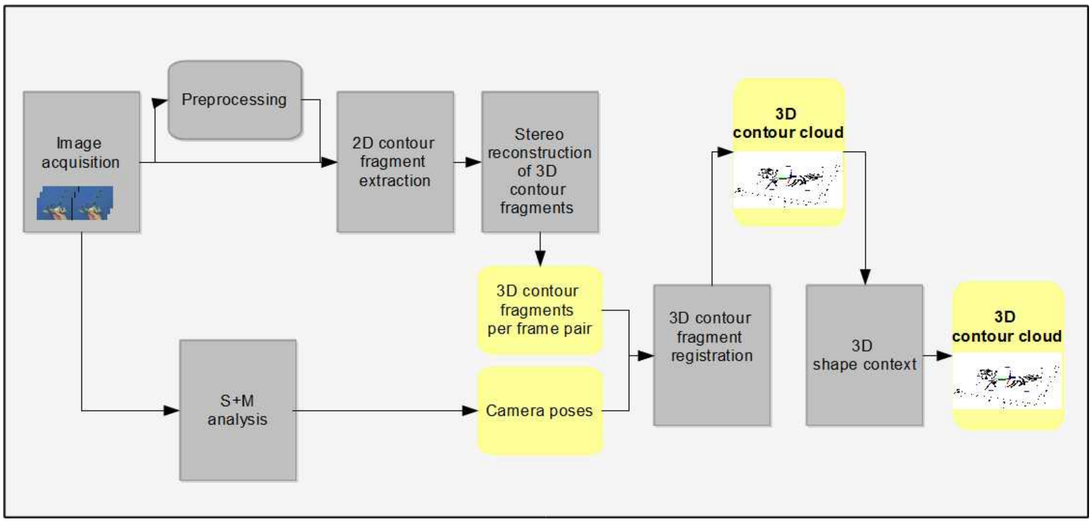


Image acquisition
Preprocessing
2D contour fragment extraction
Stereo reconstruction of 3D contour fragments
3D contour cloud
3D contour fragments per frame pair
S+M analysis
Camera poses
3D contour fragment registration
3D shape context
3D contour cloud

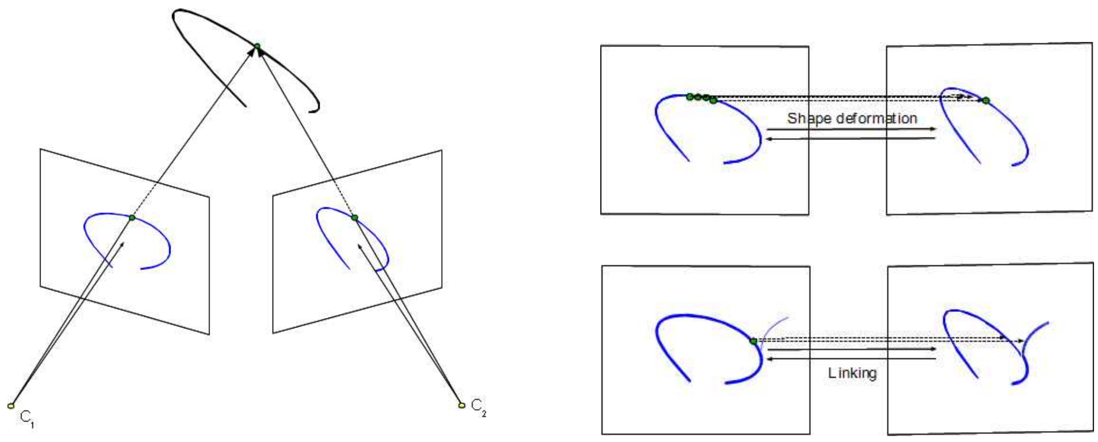
Figure 4.1: Overview of the stereo reconstruction framework.

The determination of long and salient 3D contour fragments is a rather difficult task. One main requirement in reconstructing 3D contour fragments is to find accurate point correspondences on 2D contour fragments of the left and the right stereo frame. Three problems may occur:

- Linking: Different linking of edges to longer contour fragments in different views (stereo frame pairs as well as consecutive frames) influences a matching procedure. 2D contour fragments may not have the same length, same start point, and the same end point (see Figure 4.2(b)).
- Shape deformation: The shape of contours changes significantly when viewing them from different poses. This fact makes it harder to track contours over time, to find corresponding 2D contour fragments in stereo image pairs, and point correspondences on them (see Figure 4.2(a) and Figure 4.2(b)).
- Visual rim: When viewing objects from different poses the visual rim changes. Therefore, the silhouette which is seen in the left and the right stereo frame differs. The same situation occurs due to camera movement between consecutive frames. Consequently, 2D contour fragments, which arise from different positions on the object are matched. Therefore, the main focus is the reconstruction of a qualitative '3D contour cloud' representation for the shape of a category instead of a precise 3D contour reconstruction, which is not possible just based on stereo reconstruction.

A standard stereo correspondence algorithm would compute the intersection between epipolar line and 2D contour fragments and search in a neighborhood of these intersection points for the corresponding point. In contrast to this method, our new approach integrates the well known 2D shape context with epipolar information in one single cost matrix. This is necessary as there are only few intersections along a contour that would directly permit the identification of point correspondences.

Nevertheless, 2D contour fragments with similar shape may produce false correspondences of 2D contour fragments and contour points which result in incorrectly reconstructed 3D contour fragments. For outlier reduction we introduce 3D shape context as an extension of 2D shape context.



(a)

(b)

Figure 4.2: (a) 3D contour fragments seen from different views of a stereo setting. (b) Two main problems occur when matching 2D contour fragments in different views. First, the shape of a contour may change in projection of different views. Therefore, point correspondences are harder to find (top). Second, linking of edges to longer contour fragments may be different when viewing the object from different poses (bottom).

4.3.2 Stereo Reconstruction of 3D Contour Fragments

The stereo correspondence for the reconstruction of 3D contour fragments is done in two steps. First, we find corresponding 2D contour fragments by doing a coarse stereo matching of 2D contour fragments. Then we compute point correspondences on them. 2D shape context is very suitable for 2D contour fragment matching but using 2D shape context in combination with epipolar geometry has further advantages. First, we can limit our search space to a subset of contours by only taking into account those 2D contour fragments, which lie in regions restricted by the epipolar lines. Second, we rely on point correspondences, which are more precise than using just one of the methods.

In order to find corresponding 2D contour fragments, we compute the epipolar line for four example points of a 2D contour fragment in the left frame and find those 2D contour fragments in the right stereo frame pair which lie in a region around one of the four epipolar lines. Shape similarity is tested using 2D shape context. Thus, the search space for each 2D contour fragment is reduced to a small number of possible contour fragments. In order to find the correct point correspondences on matching 2D contour fragments we compute a cost matrix by combining 2D shape context, epipolar geometry and an ordering constraint. We first achieve a clockwise ordering of contour points. Afterwards we generate a new cost matrix by building a weighted combination of the 2D shape context cost matrix and a cost matrix computed from epipolar information.

Let $\mathbf{p}_i$ be a point on a 2D contour fragment in the left stereo image and $\mathbf{q}_j$ be a point on a contour fragment in the right stereo image. Then, the original 2D shape context cost matrix (see [BMP02]) is given by

$$\mathbf{CS}_{ij} := \frac{1}{2}\sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}, \tag{4.3}$$

where $h_i(k)$ and $h_j(k)$ are the normalized histograms with K bins[1] at points $\mathbf{p}_i$ and $\mathbf{q}_j$

$$h_i(k) := \#\left\{\mathbf{g} \neq \mathbf{p}_i : (r, \theta) \in bin(k)\right\}. \tag{4.4}$$

In 2D shape context, each point $\mathbf{p}_i$ on a contour fragment is represented by a histogram, which contains the relative position to all other points on the contour fragment. The cost matrix $\mathbf{CS}$ is simply the χ^2 measure between the point-histograms of two contour fragments. Hence, the cost matrix $\mathbf{CS}$ contains for each point $\mathbf{p}_i$ on one fragment the matching cost to each point $\mathbf{q}_j$ on the second contour fragment. For our application for stereo correspondences we create an additional cost matrix, which contains for each epipolar line $\mathbf{l}_{p_i}$ of one point $\mathbf{p}_i$ of the left contour fragment the distance to each point $\mathbf{q}_j$ on the right contour fragment and vice versa. Consequently,

$$\mathbf{l}_{p_i} = \mathbf{F}\mathbf{p}_i \qquad \text{and} \qquad \mathbf{l}_{q_j} = \mathbf{F}^T\mathbf{q}_j, \tag{4.5}$$

[1] In our experiments $K = 12 * 5$, where 12 is the number of bins for θ and 5 is the number of bins for r (cmp. [BMP02])

where $\mathbf{F}$ is the Fundamental Matrix and $\mathbf{l}_{p_i}$ $(\mathbf{l}_{q_j})$ defines the epipolar lines for points $\mathbf{p}_i$ $(\mathbf{q}_j)$. This leads to the second cost matrix $\mathbf{CE}$, with

$$\mathbf{CE}_{ij} = \frac{(d(\mathbf{l}_{p_i}, \mathbf{q}_j) + d(\mathbf{l}_{q_j}, \mathbf{p}_i))^2}{max((d(\mathbf{l}_{p_i}, \mathbf{q}_j) + d(\mathbf{l}_{q_j}, \mathbf{p}_i))^2)}, \tag{4.6}$$

where d is the Euclidean distance between the epipolar line and the point. The maximum $max((d(\mathbf{l}_{p_i}, \mathbf{q}_j) + d(\mathbf{l}_{q_j}, \mathbf{p}_i))^2)$ is the maximum over all entries of the matrix and denotes a normalization factor. We combine both cost matrices by a weighted sum

$$\mathbf{C}_{ij} = w_1 * \mathbf{CS}_{ij} + w_2 * \mathbf{CE}_{ij}, \tag{4.7}$$

where w_1 and w_2 are weighting factors[2].

For a continuous reconstruction, we additionally use an ordering constraint on the contour fragments, because neighboring points should have neighboring correspondence points. By first achieving a clockwise ordering of the contour points, the corresponding points then are given by a sub-diagonal of the cost matrix or a path through the cost matrix. Based on these contour point correspondences we reconstruct the 3D contour fragments using the 'Object Space Error for General Camera Models' [SSP08].

4.3.3 3D Shape Context

Outliers - falsely reconstructed 3D contour fragments - may always occur due to false stereo correspondences or false matching over time. It is not always possible to detect falsely reconstructed 3D contour fragments solely on the basis of 2D information. Therefore, we introduce a 3D shape representation and matching based on the idea of extending 2D shape context - we call it 3D shape context - to reduce the number of outliers in a '3D contour cloud'. Koertgen et al. [KPNK03] describe a similarity measure between 3D models based on 2D shape context, which is similar to our idea.

[2] Empirically, we found $w_1 = 0.3$ and $w_2 = 0.7$ to be good choices for the weighting factors.

Remember, a '3D contour cloud' $\mathcal{C}$ consists of a number of 3D contour fragments

$$\mathcal{C} = \{\mathcal{F}_l, l = 1...N\}, \tag{4.8}$$

where N is the number of 3D contour fragments in a cloud. Each of these 3D contour fragments has been seen in several frames and tracked over time, so that each 3D contour fragment $\mathcal{F}_l$ consists of a number of reconstructed fragments

$$\mathcal{F}_l = \{f_i^l, i = 1...M\}, \tag{4.9}$$

where M is the number of frames in which $\mathcal{F}_l$ has been seen. We then use 3D shape context to verify those fragments f_i^l which have the most similar shape. 3D contour fragments which do not have a similar shape as the majority of F_l are rejected as outliers. The cost matrix on the fragment $\mathcal{F}_l$ is defined by

$$\mathbf{CF}_{ij} = sc\_cost\_3D(f_i^l, f_j^l) \qquad \forall i, j \in M, \tag{4.10}$$

where $sc\_cost\_3D(f_i^l, f_j^l)$ is the 3D shape context matching cost between fragment f_i^l and f_j^l. By analyzing the median and variance of this cost matrix we identify those tracked 3D contour fragments f_n^l which are not similar to the other fragments of F_l.

The 3D shape context matching cost $sc\_cost\_3D(f_i^l, f_j^l)$ is defined in a similar way as in 2D. In order to handle shape deformations as well as spatial displacements of 3D contour fragments on the object (see Section 4.4.1), our 3D shape context cost depends on two measures $\mathbf{CS3D}_{ab}$ and $\mathbf{CP3D}_{ab}$. Because we are only interested in corresponding 3D fragments and not in corresponding 3D contour points, we do not include an ordering constraint (in contrast to Section 4.3.2).

Let $\mathbf{p_a}$ be a 3D contour point of fragment f_i^l. Similar to 2D shape context we build a multi-dimensional histogram $h1_a$ by computing the relative positions to all other contour points $\mathbf{g}$ on fragment f_i^l in a 3D log-polar space. Hence, we compute the distance r, the azimuth θ, and elevation ϕ. The basis for this computation builds an object-centered coordinate system. Figure 4.3 illustrates the principle.

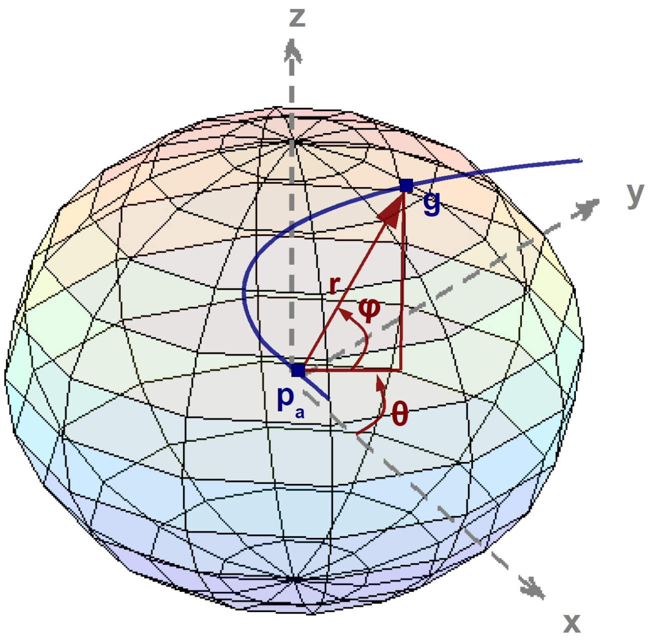


Figure 4.3: 3D shape context description: For each point on a 3D contour fragment (blue curve) the relative position $(r,\ \phi,\ \theta)$ to all other points is computed.

Then, our 3D shape context is defined in the following way

$$h1_a(k) := \#\left\{\mathbf{g} \neq \mathbf{p}_a : (r, \theta, \phi) \in bin(k)\right\}. \tag{4.11}$$

Here, the cost matrix **CS3D** between two fragments is defined in the same manner as in the 2D case using the χ^2 measure

$$\mathbf{CS3D}_{ab} := \frac{1}{2}\sum_{k=1}^{K}\frac{\left[h1_a(k) - h1_b(k)\right]^2}{h1_a(k) + h1_b(k)}, \tag{4.12}$$

for points $\mathbf{p_a}$ and $\mathbf{q_b}$ and K bins[3]. Taking the position on the object into account, we compute a second, very sparse histogram $h2_a$. To obtain the position of the contour points to a reference point, we compute the distance r, the azimuth θ, and elevation ϕ relative to the defined reference point **oc** e.g. object center:

$$h2_a(k) := \#\left\{\mathbf{oc} \neq \mathbf{p}_a : (r, \theta, \phi) \in bin(k)\right\}. \tag{4.13}$$

[3] In our experiments the number of bins $K = 12 * 12 * 5$ where 12 is the number of bins for θ and ϕ, and 5 defines the number of bins for r.

This histogram is sparse in the sense, that for each point p_a we just have one entry - the bin which contains (r, θ, ϕ) for p_a. The cost matrix is given by

$$\mathbf{CP3D}_{ab} := \frac{1}{2}\sum_{k=1}^{K} \frac{[h2_a(k) - h2_b(k)]^2}{h2_a(k) + h2_b(k)}. \tag{4.14}$$

The overall cost matrix is then given by a weighted sum of both cost matrices:

$$\mathbf{C3D}_{ab} = w_1 * \mathbf{CS3D}_{ab} + w_2 * \mathbf{CP3D}_{ab}, \tag{4.15}$$

where w_1 and w_2 are weighting factors[4]. The 3D shape context cost *sc_cost_3D* between two 3D contour fragments f_i^l and f_j^l is given by

$$\begin{aligned} &sc\_cost\_3D(f_i^l, f_j^l) = \\ &max(\frac{1}{A}\sum_{a=1}^{A} \min\nolimits_b \mathbf{C3D}_{ab}, \frac{1}{B}\sum_{i=1}^{B} \min\nolimits_a \mathbf{C3D}_{ab}) \end{aligned} \tag{4.16}$$

where A and B are the numbers of contour points on fragment f_i^l and f_j^l, respectively. Similar to the 2D case [BMP02], the 3D shape context cost *sc_cost_3D* defines the 3D shape context distance between two 3D contour fragments on the basis of their best matching points. 3D contour fragments with a matching cost above a threshold with respect to the majority of all other fragments are detected as outliers. Here, the threshold is defined by computing the median and the variance of the cost matrix.

4.4 Experimental Evaluation

The difficulty in an experimental evaluation is the lack of ground truth for real world data. The effort to produce ground truth as e.g. point correspondences for each 2D contour fragment in each frame of a stereo image sequence is all but impossible and even for humans it is very hard to find exact point correspondences. Therefore, we evaluate the 3D shape context on the basis of synthetic contours, where ground truth is available. In addition to this 3D shape context evaluation, we show reconstruction results in form of '3D contour clouds' for objects of our

[4] We found empirically $w_1 = 0.6$ and $w_2 = 0.4$ to be good choices for the weighting factors

own GRAZ-STEREO-BASE-xx dataset (see Section 3.2) as well as for objects of a standard multi-view dataset [SCD$^+$06].

4.4.1 Evaluation of the 3D Shape Context

In Section 4.3.3 we introduce the concept of 3D shape context for outlier reduction. Now, we evaluate it on the basis of synthetic contour fragments. In our experiments we analyze the following cases:

- Incorrect contour fragment stereo correspondences: Our stereo correspondence algorithm is based on 2D shape information, epipolar geometry and an ordering constraint. Incorrect contour fragment stereo correspondences happen when 2D shape and epipolar geometry of several contour fragments are similar e.g. for two contour fragments of horse legs. Such matching errors lead to a deformation of the 3D shape and to a displacement of 3D contour fragment position on the object (see Experiment 1).

- Incorrect contour fragment correspondences over time: For the registration of 3D contour fragments, 2D contour fragments are tracked over time in the left and the right stereo frame pairs. Similar shape and position may lead to false correspondences in consecutive frames. Thus, 3D contour fragments f_i^l with different 3D shape and position on the object are grouped together to one 3D contour fragment $\mathcal{F}_l$ (see Experiment 1).

- Incorrect point correspondences: Due to the ordering constraint in our stereo correspondence algorithm, it can not happen that crossings occur in our point correspondences (see Figure 4.4(b)). However, shifts may occur i.e. points do not match to their corresponding point but to their neighbors (see Figure 4.4(c)). One or more shifts lead to a shape deformation of the reconstructed 3D contour fragment (see Experiment 2).

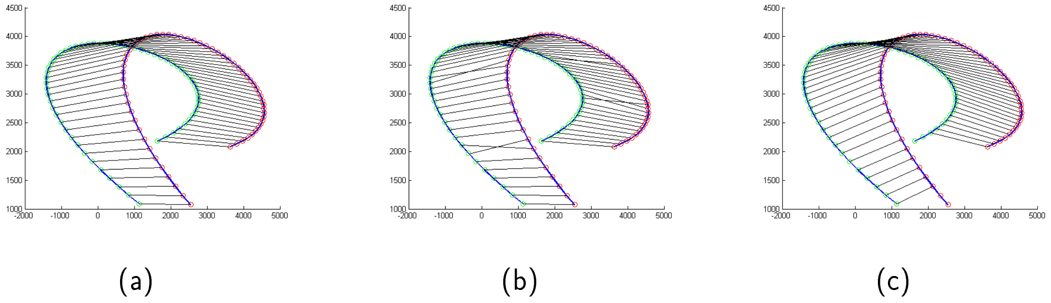


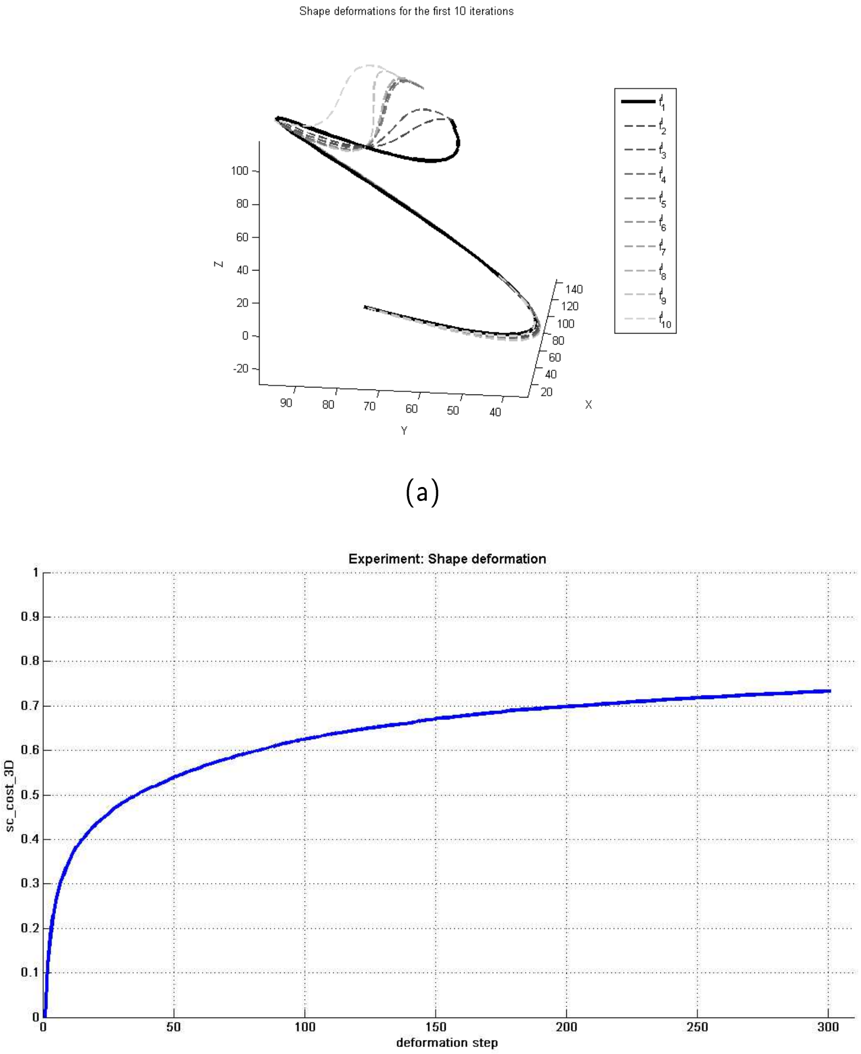
Figure 4.4: (a) Perfect point correspondence between two 2D contour fragments. (b) Crossings - point p_i of first fragment should match to point q_i of second fragment, point p_{i+1} to point q_{i+1}, but point p_i match to q_{i+1} and p_{i+1} to q_i - occur. (c) Shifts of 2 positions - point p_i of first fragment should match to point q_i of second fragment, p_{i+1} to q_{i+1} and so on, but p_i match to q_{i+2}, p_{i+1} match to q_{i+3} and so on - occur.

Experiment 1

In this experiment we show the behavior of the 3D shape context in the case of 3D shape deformation and spatial displacement of 3D contour fragments on the object. As mentioned before, 3D shape deformations and spatial displacements occur as a result of incorrectly determined contour fragment stereo correspondences and incorrect contour fragment correspondences over time.

We use the following testing framework in this experiment: We randomly generate a synthetic 3D contour fragment f_1^l by randomly choosing points in 3D, which build supporting points for interpolating splines. To achieve 3D shape deformation we iteratively add noise to a subset of spline points. To achieve a spatial displacement of the 3D contour fragment we iteratively add noise to all points of the 3D contour fragment. For the evaluation we measure the 3D shape context cost sc_cost_3D between manipulated 3D contour fragments and the original 3D contour fragment. For a statistical evaluation we repeat the experiment for 1000 times and build the average cost.

To evaluate the influence of 3D shape deformation on the 3D shape context cost, we start with a first experiment where we just manipulate the shape (see Figure 4.5(a) for an example). Figure 4.5(b) shows the behavior of the 3D shape context cost sc_cost_3D. As desired, the more 3D shape deformations occur the larger becomes the 3D shape context cost sc_cost_3D.



(a)

(b)

Figure 4.5: (a) Shape deformations for the first 10 iterations for an example synthetic 3D contour fragment; the original 3D contour fragment f_1^l is shown in dark, the deformed contour fragments $f_2^l \dots f_{10}^l$ are shown as dotted lines in gray colors. (b) Average 3D shape context cost sc_cost_3D over 1000 test runs for increasing deformations (300 iterations).

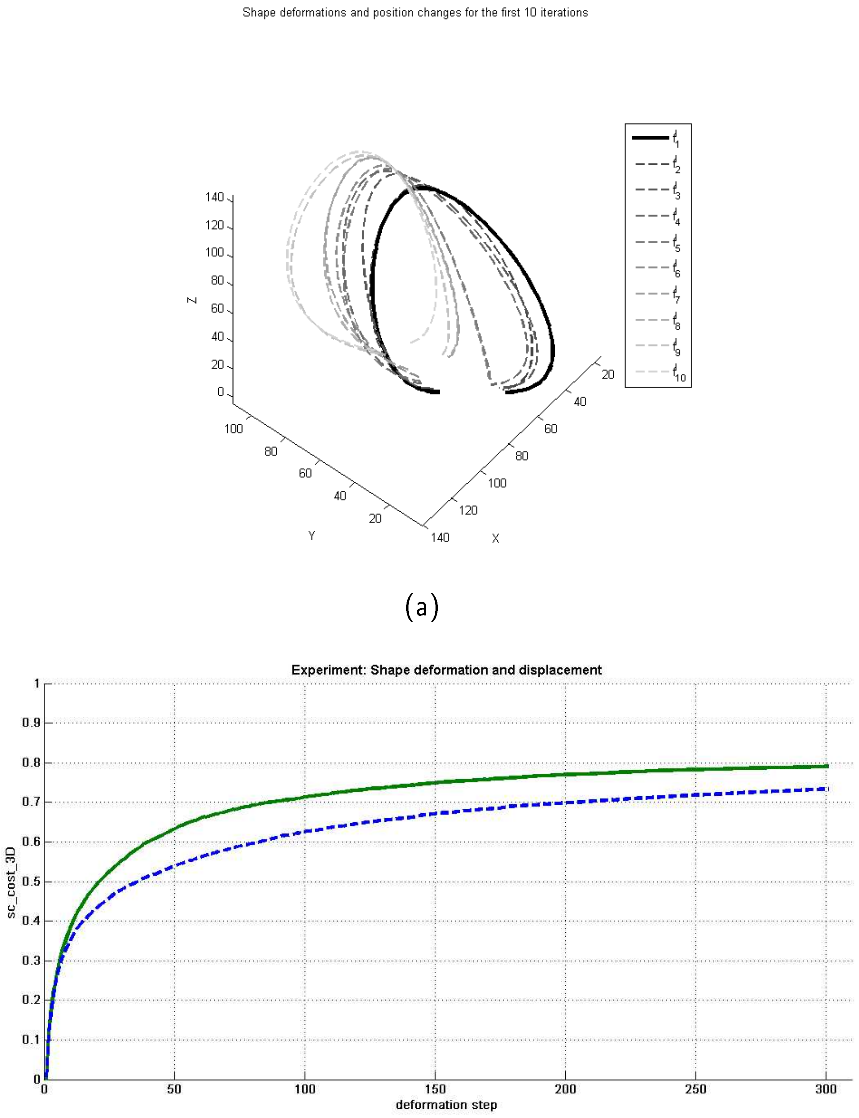
To evaluate the influence of 3D shape deformation and spatial displacements of a 3D contour fragment at the same time we simultaneously deform the 3D shape and displace the 3D contour fragment regarding to the 3D coordinate system (see Figure 4.6(a)). Figure 4.6(b) shows how the 3D shape context cost sc_cost_3D increases when simultaneously simulating 3D shape deformations and spatial displacements.

Experiment 2

In this experiment we show the behavior of the 3D shape context cost in the case of incorrect point stereo correspondences (shifts).

We use the following testing framework in this experiment: We generate a synthetic 3D contour fragment f_1^l by randomly choosing points in 3D, which build supporting points for interpolating splines. In order to simulate shifted point correspondences, we reproject this 3D contour fragment f_1^l to 2D (in a left and a right stereo frame) and compute shifted correspondences (up to fifteen pixels) for this 2D contour fragment pair. Afterwards, we reconstruct the 3D contour fragments $f_2^l \ldots f_{16}^l$ from these correspondences and compute the 3D shape context cost sc_cost_3D between $f_2^l \ldots f_{16}^l$ and the original 3D contour fragment f_1^l. For a statistical evaluation we repeat the experiment 1000 times.

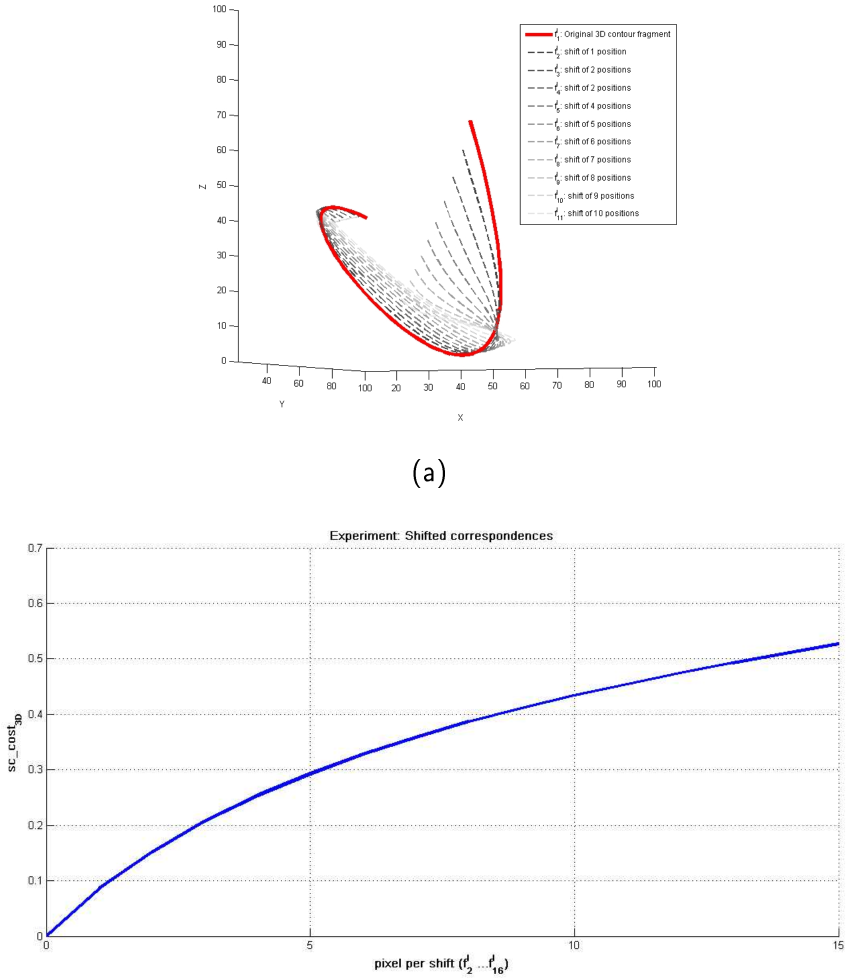
Figure 4.7(a) shows an example of an original 3D contour fragment and those 3D contour fragments which were reconstructed from 2D contour fragments with shifted point correspondences. Figure 4.7(b) shows the average 3D shape context cost sc_cost_3D. We can see how the cost increases when more shifted correspondences occur. Whereas a one pixel shift (f_2^l) results in a 3D shape context cost $sc\_cost\_3D = 0.0855$, it increases to a 3D shape context cost $sc\_cost\_3D = 0.5275$ for a 15 pixel shift (f_{16}^l). As desired, larger shifts, which lead to heavily distorted contour fragments, lead to higher costs.



(a)

(b)

Figure 4.6: (a) Shape deformations and spatial displacements for the first 10 iterations of an example synthetic 3D contour fragment; the original 3D contour fragment f^l_1 is shown in dark, the deformed and displaced 3D contour fragments $f^l_2 \dots f^l_{10}$ are shown as dotted lines in gray colors. (b) Average 3D shape context cost sc_cost_3D for 300 iteration steps (plotted in green). The average is built over 1000 test runs; the curve of the Experiment in Figure 4.5 is shown for comparison (dotted blue line).



(a)

(b)

Figure 4.7: (a) f_1^l: Original contour fragment (red), example shifted correspondences up to the first 10 pixels are simulated in the stereo correspondences: $f_2^l \dots f_{11}^l$. (b) Evaluation of the average 3D shape context cost $shape\_cost\_3D$; the experiment was done 1000 times and the average over the costs are computed.

4.4.2 Evaluation on GRAZ-STEREO-xx Dataset and Multi-view Dataset

We evaluate our '3D contour cloud' reconstruction approach on our own GRAZ-STEREO-xx dataset, which consists of three types of stereo image sequences (see Section 3.2), as well as on a standard multi-view dataset [SCD$^+$06].

3D Contour Clouds of GRAZ-STEREO-BASE-EYE

The GRAZ-STEREO-BASE-EYE consists of stereo image sequences of small hand-held objects, which are manipulated in front of the stereo rig (see Section 3.2.2). First, we demonstrate our '3D contour cloud' reconstruction framework on one horse stereo image sequence from this dataset. The stereo image sequence shows a small toy horse with the dimensions *length* = 175mm, *height* = 95mm, and *width* = 45mm (see Figure 4.8 first and third row).

The estimated camera poses of the stereo rig are shown in Section 3.2.2 in Figure 3.7. We can see that we manipulate the object by first rotating it for approximately 360^o around the vertical axis of the horse and then by about 180^o around the horizontal axis. In our reconstruction framework we compute 3D contour fragments of the object for every fifth frame. Figure 4.8 shows the output for this stereo reconstruction process - 3D contour fragments for single stereo frame pairs. We can see how the 3D shape changes over time when we rotate the object.

Figure 4.9 shows a '3D contour cloud' of the horse for 201 frames without outlier reduction. Figure 4.10 shows the same '3D contour cloud' after outlier reduction using the 3D shape context. Furthermore, only 3D contour fragments are drawn, which were tracked for at least 3 frames. 201 frames correspond to a rotation of the object of approximately 270^o around its vertical axis. We used every fifth frame during the reconstruction. Instead of visualizing one representative contour fragment, we show all registered contours in corresponding color. This explains the width of the visualized contour fragments. We see that a large fraction of outliers has been removed successfully so that the shape of the horse is clearly visible. One can observe that there are no contours reconstructed for the back of the horse. This is caused by the fact that the back is occluded by the hand during the first part of manipulating the horse.

Figure 4.11 shows a reconstructed '3D contour cloud' with an automatically generated 3D bounding box and object-centered coordinate system. We choose the longest dimension of the 3D bounding box as x-axis. For the purpose of

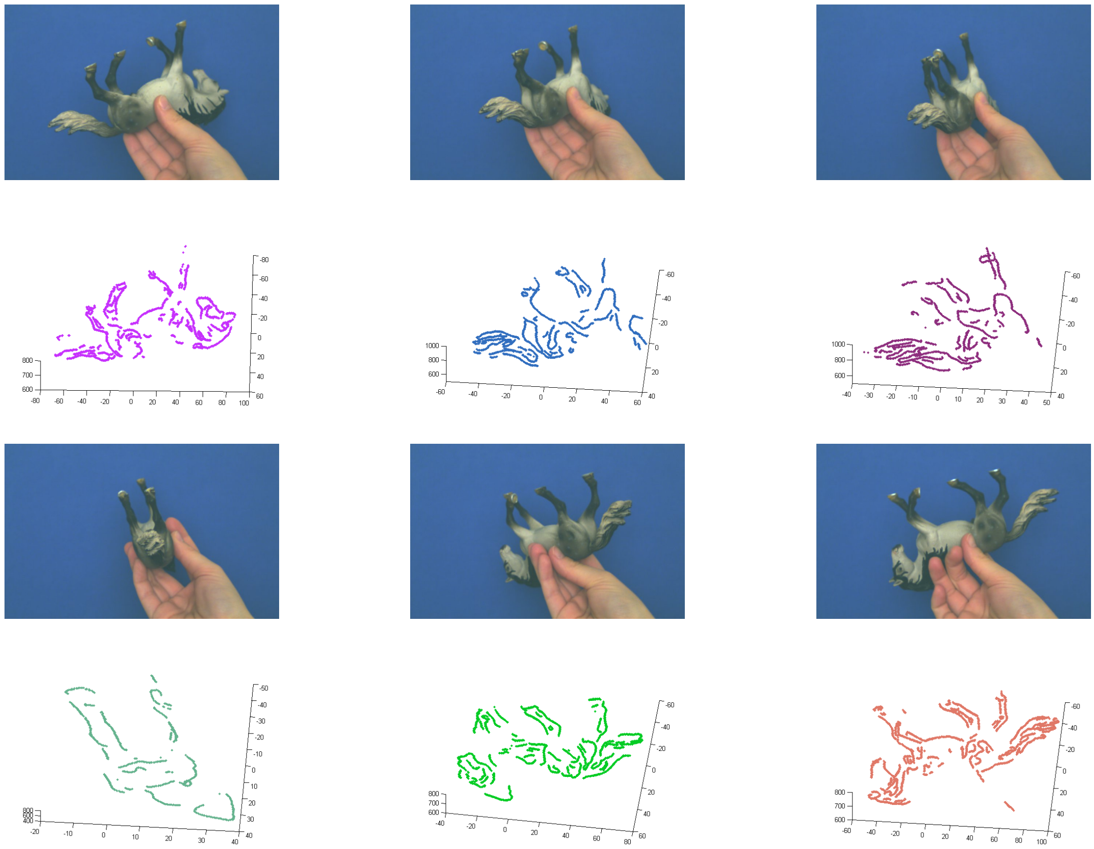


Figure 4.8: 3D contour fragments per stereo frame pair for a small subset of stereo frame pairs over 180^o. For space-saving only the left frames of stereo pairs are shown above their corresponding reconstructions; 3D contour fragments per stereo frame pair are drawn in the same color to demonstrate the shape differences between several camera poses.

visualization, we choose a '3D contour cloud' with a reduced number of 3D contour fragments, which were reconstructed just over a short subsequence of the stereo image sequence.

3D Contour Clouds of GRAZ-STEREO-BASE-30

The GRAZ-STEREO-BASE-30 consists of calibrated stereo image sequences showing humans that rotate around their vertical axis (see Section 3.2.3). Figure 4.12(b) shows the estimated camera poses of the stereo rig around the human seen in Figure 4.12(a) in an object centered coordinate system. We can see that the human

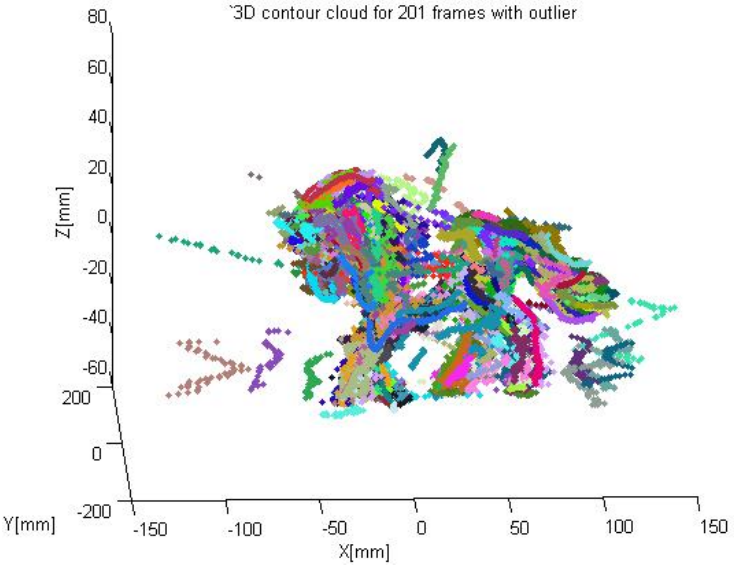


Figure 4.9: '3D contour cloud' of a horse for 201 frames. In this reconstruction, outliers have not been removed and thus all reconstructed 3D contour fragments are drawn.

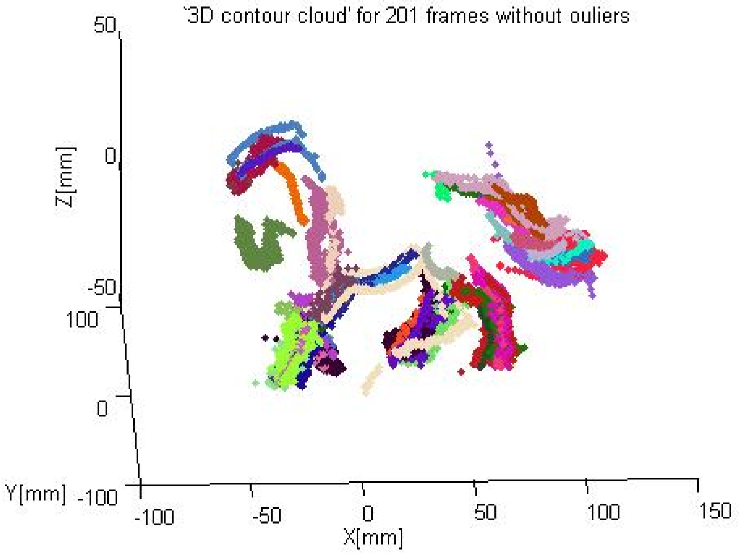


Figure 4.10: '3D contour cloud' of a horse for 201 frames. Outliers have been removed using 3D shape context and the median as threshold. Only those 3D fragments are visible which have been tracked over ≥ 3 frames.

rotates approximately 360^o around his vertical axis. Figure 4.12(c) shows a '3D contour cloud' of the human. We can clearly identify the shape of the human. Again, instead of visualizing one representative 3D contour fragment, we show

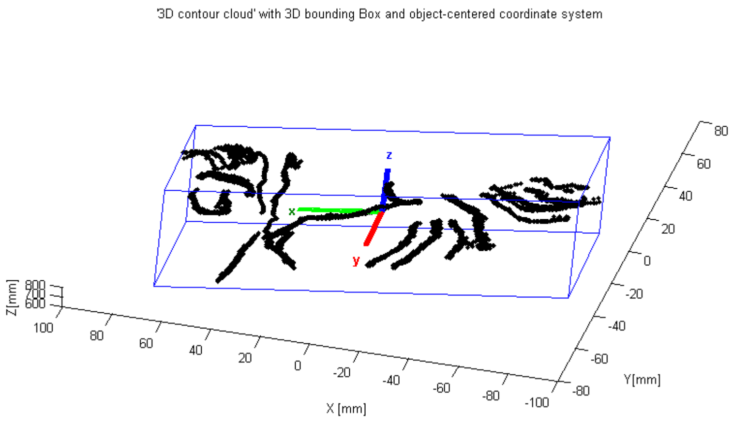


Figure 4.11: '3D contour cloud' with automatically generated 3D bounding box and object-centered coordinate system.

all registered contours in corresponding color, which explains the width of the visualized contour fragments. We can see by the example of the green 3D contour fragment on the arm of the person how the visual rim is tracked over several frames.

Figure 4.13 shows the results for a second human from our GRAZ-STEREO-BASE-30 dataset. Again, the human makes a full turn around his vertical axis (see Figure 4.13(b)). Figure 4.13(c) shows a '3D contour cloud' of the human generated from the entire stereo image sequence without outlier reduction, whereas the outlier reduction using 3D shape context is demonstrated in Figure 4.13(d).

3D Contour Clouds of GRAZ-STEREO-BASE-EYE-TURNTABLE

The GRAZ-STEREO-BASE-EYE-TURNTABLE consists of calibrated stereo image sequences showing toy objects on a turntable (see Section 3.2.4). Figure 4.14(b) shows the estimated camera poses of the stereo rig around a dog (see Figure 4.14(a)) in an object centered coordinate system. We can see a rotation of approximately 360^o around its vertical axis. Figure 4.14(c) shows the corresponding '3D contour cloud'. We can clearly identify the shape of the dog. Again, instead of visualizing one representative contour fragment, we show all registered contours in corresponding color, which explains the width of the visu-

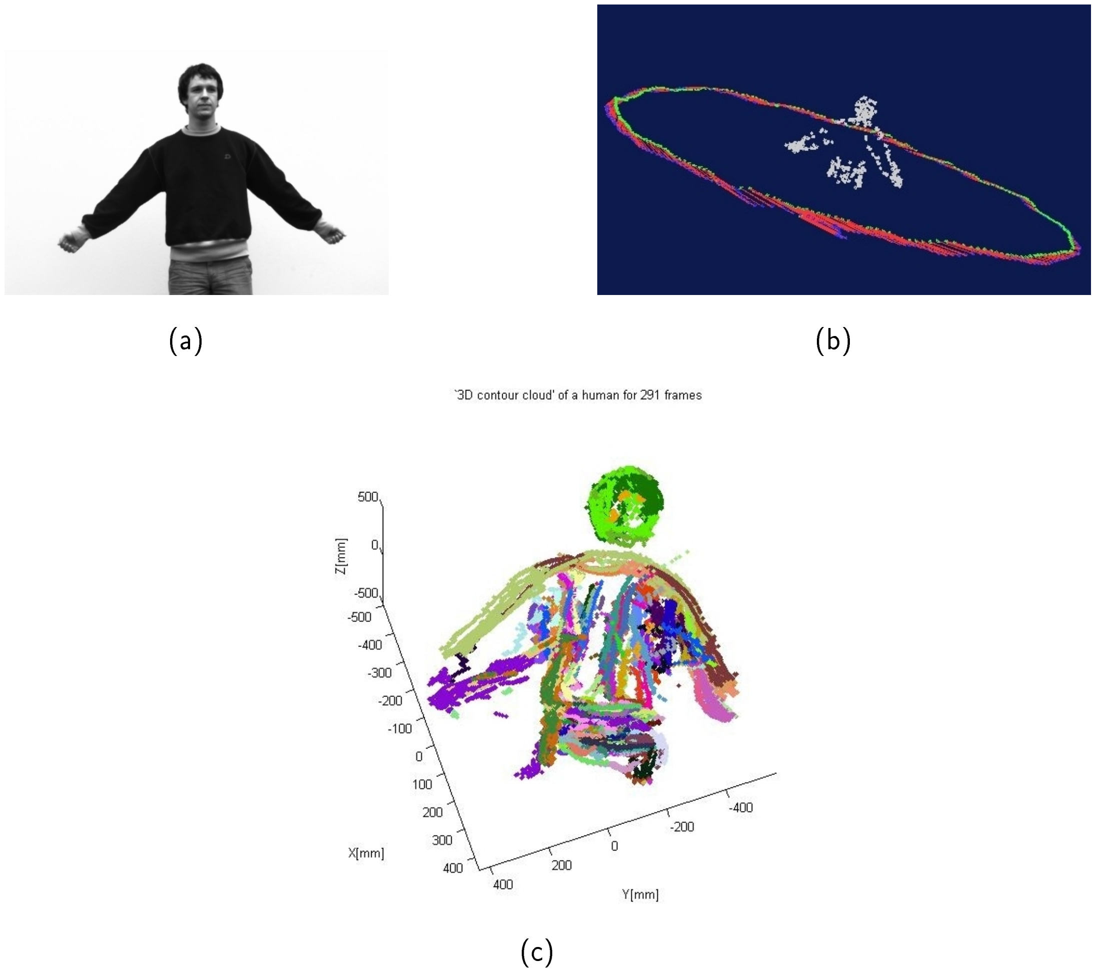
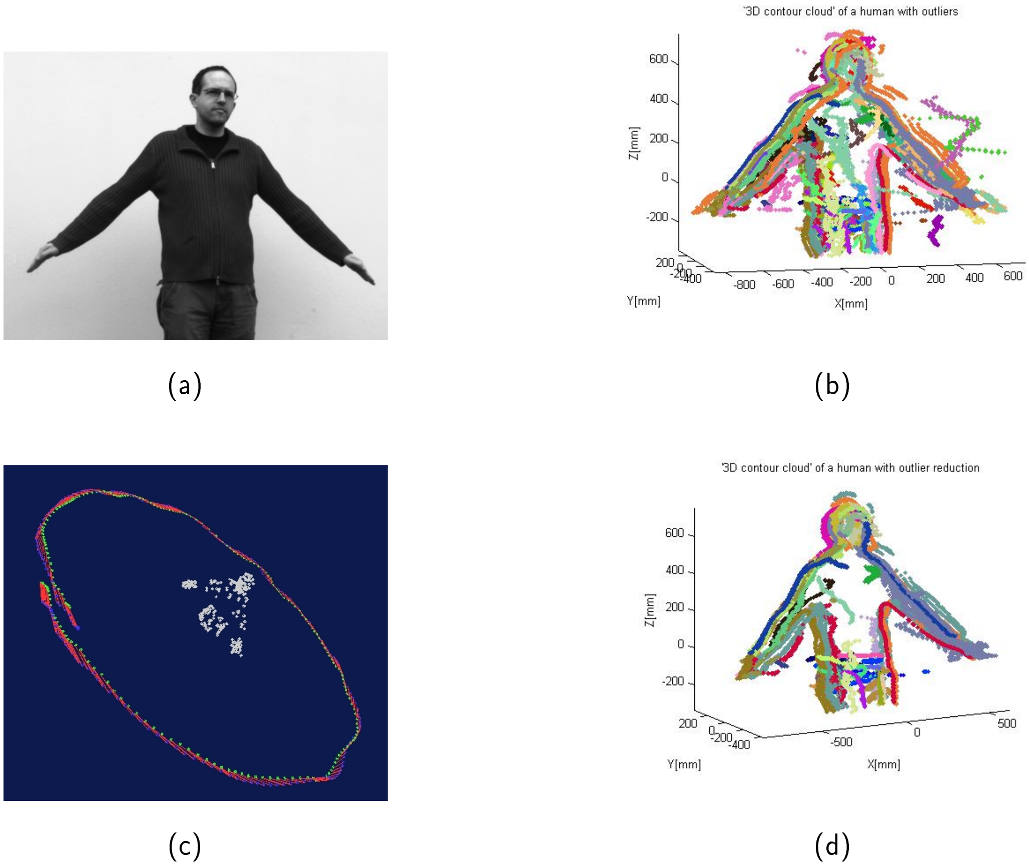


Figure 4.12: '3D contour cloud' (c) and camera trajectory (b) of a human (a) of the GRAZ-STEREO-BASE-30 dataset.

alized contour fragments. Outliers have been removed using 3D shape context and the median as threshold. Only those 3D fragments are visible, which have been tracked over at least three frames.

3D Contour Clouds of the Multi-view stereo dataset

To demonstrate that our method is also applicable to standard datasets, we apply it to the multi-view dataset of [SCD$^+$06], which consists of the Dino-dataset (see Figure 4.15) and the Temple-dataset (see Figure 4.16), including camera



(a) (b)

(c) (d)

Figure 4.13: Stereo reconstruction of a human: (a) Left frame. (b) Estimated camera poses. (c) '3D contour cloud' with outliers. (d) '3D contour cloud' without outliers and 3D contour fragments which are tracked over ≥ 2 frames.

parameters and camera poses. For these datasets, views sampled on a hemisphere or on a ring are available.

Figure 4.17 shows a '3D contour cloud' for a subset of images of the Dino-dataset, and Figure 4.18 for the Temple-dataset. For the visualization we choose a small subsample of images captured from the same aspect (side view of the dino and the temple). Similar to the other datasets, we show all registered contours in corresponding color, which explains the width of the visualized contour fragments. We can see that the plates of the stegosaurus result in many different 3D contour

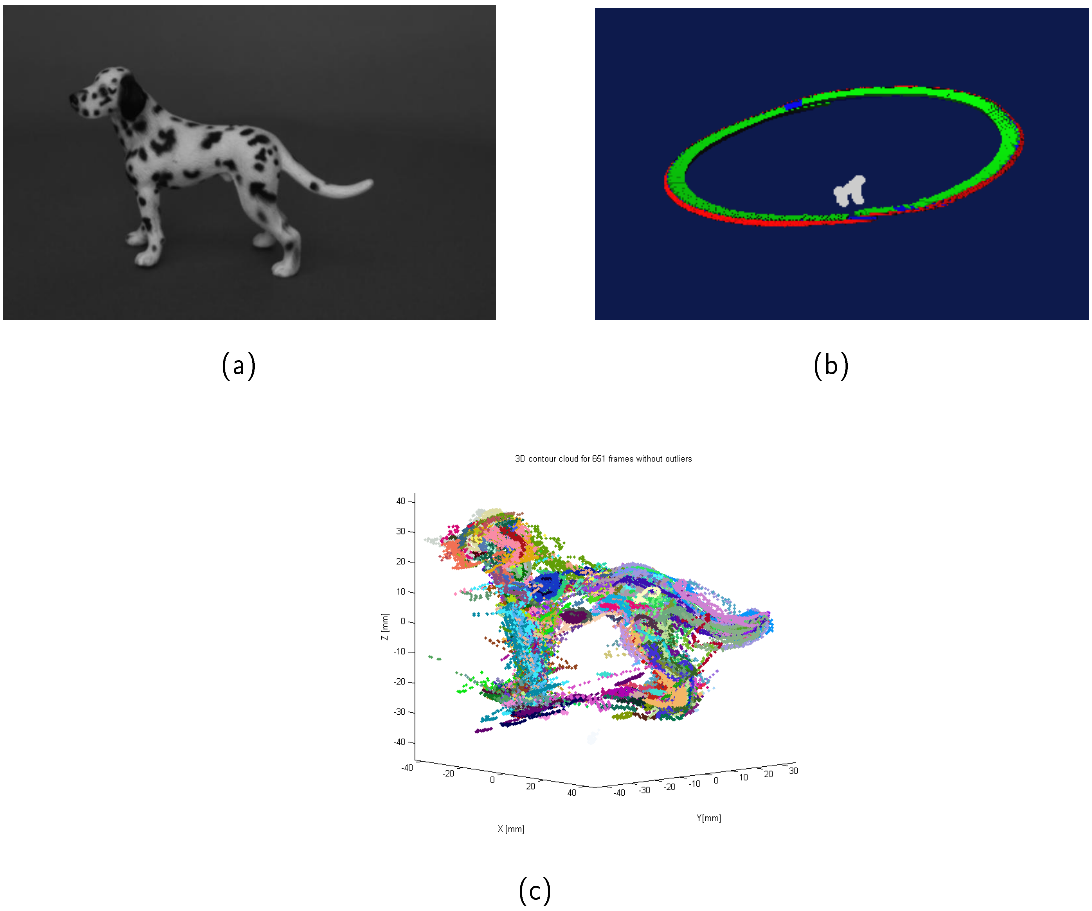


Figure 4.14: (a) Left frame of a dog of the GRAZ-STEREO-BASE-EYE-TURNTABLE dataset. (b) Camera poses of the stereo rig (green and blue triangle connected by a red line) estimated around the dog using [SSP08]. (c) '3D contour cloud' for 651 frames where outliers were reduced.

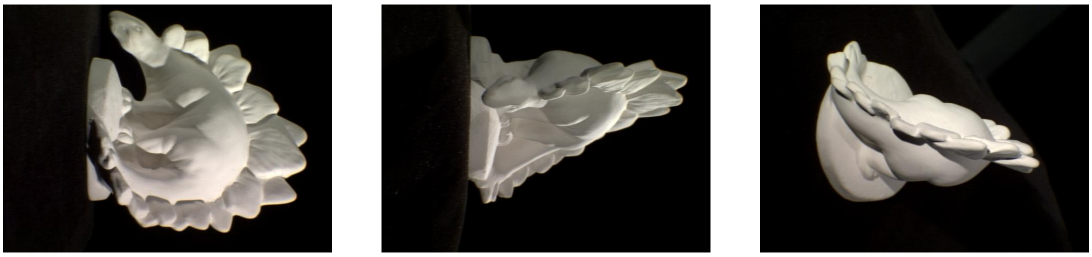


Figure 4.15: Sample images of the Dino-dataset. 363 views are sampled on a hemisphere.

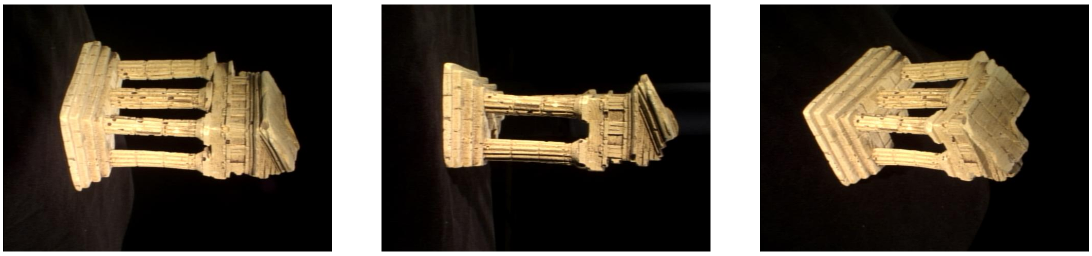


Figure 4.16: Sample images of the Temple-dataset. 312 views are sampled on a hemisphere.

fragments because of different illumination effects and that the striation on the pillars delivers many non-distinguishable inner contour fragments.

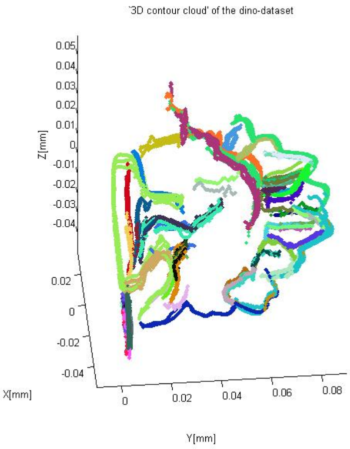


Figure 4.17: '3D contour cloud' of the dino.

4.5 Discussion

At the moment our reconstruction algorithm works on stereo image sequences, where an object is presented to the camera in front of a homogeneous background and either the object or the camera is rigid. This is due to the underlying Structure-and-Motion framework, which was built for the reconstruction of a rigid scene using a moving camera. The system is stable up to 50% outliers, which means that more than half of the tracked feature points have to be stationary on a rigid object; no (major) foreground motion is allowed. In contrast, in the present

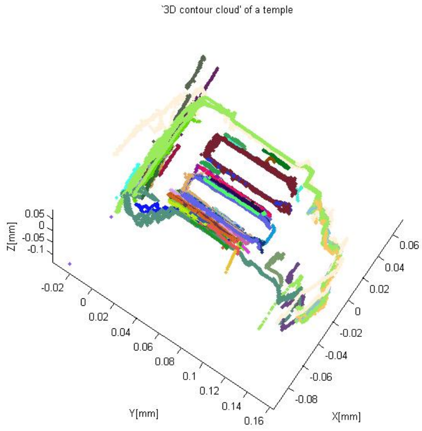


Figure 4.18: '3D contour cloud' of the temple.

application we use stereo image sequences from a stationary camera whereas the foreground object is moved. However, the stereo reconstruction framework is in principle also applicable to more sophisticated stereo image sequences using a multibody structure and motion framework e.g. [HP10] .

We discussed earlier in this chapter problems and difficulties of reconstructing long and salient 3D contour fragments and we have presented a novel method for stereo correspondence and outlier reduction. However, we did not discuss the behavior of the system regarding matching consistency and evolution of shape in time. In order to make sure that the 3D contour cloud contains only those 3D contour fragments which are consistent over time, each 3D contour fragment has to be tracked over a defined number of frames. There are several scenarios: a contour fragment momentarily breaks, a contour fragment constantly changes (as it is the case at the visual rim), or the linking of a contour fragment changes. In these cases, a contour fragment is just tracked as long as the shape similarity (based on a threshold) is granted. Therefore, it may happen, that one or more 3D fragments represent the same contour on the object. We do not discard the individual 3D contour fragments but we keep them in our representation (which explains the 'thickness' of some fragments in our experiments). Thus, also the shape variability of the visual rim is represented. For the task of categorization, it is left to the learning process to select the category specific 3D contour fragments.

4.6 Conclusion

I have presented an automatic reconstruction method that can generate '3D contour clouds' for several objects and various databases. For this, I have developed a stereo correspondence algorithm which integrates 2D shape context information, epipolar information and an ordering constraint into one single cost matrix. Furthermore, the system generates 3D bounding boxes and object-centered coordinate systems. Although, most of the experimental results are obtained on the GRAZ-STEREO-BASE-xx dataset, I show that the approach is also applicable to a standard multi-view stereo dataset. Moreover, an extension of standard 2D shape context towards 3D is presented. This can be useful in various 3D shape matching applications. I have shown the application of 3D shape context for outlier reduction in '3D contour clouds'.

The representation by '3D contour clouds' constitutes a powerful method for geometric modeling of 3D shape of objects. Various computer vision and graphics applications e.g. 3D shape retrieval or matching may benefit from such 3D shape modeling by '3D contour clouds' as presented in this thesis. However, my main interest is in object categorization using pose-invariant 3D shape models. Here, the focus is not a precise reconstruction but rather a qualitatively convincing representation of 3D shape of a category by salient contour fragments. I'm aware that an exact silhouette reconstruction is not possible based on two views where different silhouette contours may be seen on the 'visual rim' of the object. On the other hand, there is the advantage that all aspects of the object's 'visual rim' are integrated into one single 3D model. Especially, active object categorization systems [RP11] can benefit from such 3D category models to get a category and a pose hypothesis. The use of this 3D shape representation for 3D object categorization is subject of the next chapter.

5

3D Gaussian Contour Category Model

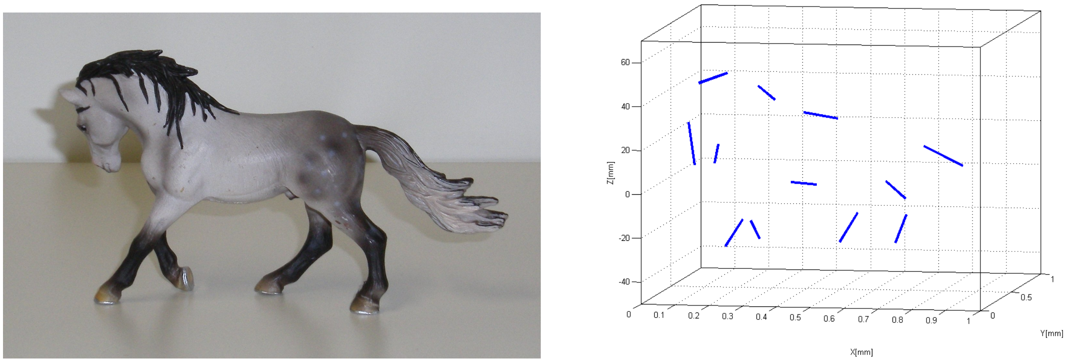
(How) can we learn one single, pose-invariant 3D category model based on shape information? The previous Chapter 4 introduces a model that describes the shape of a *specific* object with 3D contour fragments (1D manifolds embedded in 3D). The idea is now to use several such models to learn one model that describes a whole *category* with all its intra-class variability and inter-class difference to other categories.

In this chapter, I present a probabilistic framework based on Gaussian Mixture Models for 3D contour fragments (see Section 5.3) for learning such a 3D category model. The category model itself consists of partitions of Gaussian Mixture Models, which are selected based on local shape information and 3D geometric information (see Section 5.4). Experiments and results on the GRAZ-STEREO-BASE-xx dataset are discussed. The results demonstrate the capability of the method to categorize 3D objects from categories with small inter-class difference.

5.1 Introduction

2D object categorization based on 2D contour models [OPZ06, SBC08] relies on the fact that contour fragments are sufficient to represent object shape [LB83]. 2D object categorization systems based on shape information try to mimic the human ability to categorize objects just using a few contour fragments. Similarly, we want to extend this concept to 3D. Figure 5.1(b) demonstrates this: When humans look at Figure 5.1(b), they can easily identify the object depicted by this simple collection of straight 3D lines only. For our category model we make use of the following principle:

Let us assume that we have a 3D model of an object such as the horse in Figure 5.1(a). Such a horse consists of several parts, e.g. head, legs, back, tail etc. All these parts are in certain spatial relations to each other. Consequently, if we assume we have found contour fragments that represent the head of a horse, we also assume that there are regions in the vicinity of these fragments which have a high probability to show a leg or a tail. In our model we essentially describe these spatial relations between parts of an object by means of small sets of contour fragments.



(a) (b)

Figure 5.1: (a) Photography of a horse. (b) A collection of 3D line segments that describes the horse.

The probabilistic framework can be summarized as follows: We use probabilistic density functions for 3D shape modeling. The 3D contour fragments - 1D manifolds in 3D - are represented as Gaussian Mixture Models (GMMs). The main idea is to perform a partitioning - selection of discriminative subsets - of the GMM representations of specific objects. During the learning process only such partitions are kept that are commonly found in many GMM representations of objects of this category. We find discriminative subsets of the probability density functions using a random feature selection algorithm and a special distance function based on pairwise spatial relations and similarity of mixture components.

In our framework the 3D contour fragments for specific objects are reconstructed from stereo image sequences (‘3D contour clouds’ for specific objects of a category, ref. to Chapter 4). However, any other method for the generation of 3D contour fragments for specific object may be used. In contrast to 2D based approaches, we build one single 3D model per category instead of one 2D model per significant view.

5.2 State-of-the-Art

Gaussian Mixture Models (GMMs) are often used in computer vision for applications such as pose estimation [LS10], shape modeling, shape matching, shape retrieval [PR09, LViAN10] or point set registration and atlas construction [CT99, WVRE08, JV05]. Most of these methods take the whole Gaussian Mixture Models into account by defining or approximating information theoretic divergences on these mixtures.

Wang et al. [WVRE08] have presented an approach for atlas construction (i.e. computing a single representative of a population of shapes) and a non-rigid registration of shapes. They model each shape using a Gaussian Mixture Model (obtained from a point-set representation of the shape) and minimize the Jenson-Shannon divergence between different shape models of the population. In contrast to other atlas construction methods where the atlas construction requires the non-rigid registration computation beforehand, Wang et al. compute the atlas simultaneously with non-rigid registration.

Jian and Vermuri [JV05] have presented a robust method for the registration of 2D and 3D point sets. Their method is based on the usage of Gaussian Mixture Models for point set representation. The registration is done using a closed-form expression of the L_2 distance between them.

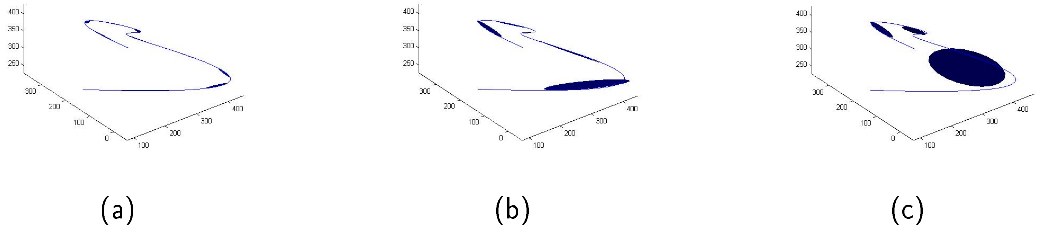
Peter et al. [PR09] represent a unified 2D shape representation and deformation framework based on Gaussian Mixture Models for landmark shape analysis where correspondence is known. They show that the Fisher-Rao metric is a Riemannian metric and can be used to compute a geodesic between shapes which describes an intrinsic deformation. Additionally, and due to lack of a closed form Fisher-Rao metric between GMM they develop a new Riemannian metric - the α-order entropy metric.

Liu et al. [LViAN10] have introduced the Total Bregman divergence for shape retrieval. We compare this approach to our method by applying it on our dataset. For details we refer to Section 5.5.3.

All these methods have in common that they obtain mean shape models solely based on the GMM models of specific objects. Intra-class variability is not explicitly addressed. Our probabilistic framework also uses Gaussian Mixture Models to represent shape. However, in contrast to above methods we assume that information about shape deformation probabilities between objects of the same category is not incorporated in the GMMs. Our similarity measure hypothesis testing framework makes use of local shape information (which is commonly ignored in most of the mentioned approaches by setting the covariance matrix to the identity matrix) and 3D geometric information. Based on a mathematical derivation we emphasize the differences of our approach with respect to the well-known Kullback-Leibler divergence.

5.3 Representation Model: From 3D Contour Clouds to 3D Gaussian Contour Models

Modeling 3D contour fragments by probability density functions has several advantages. The representation of 3D contour fragments by Gaussian Mixture Models (GMMs) is very flexible and robust. In particular, noise, outliers, and deformations can be handled in a simple, natural way and the 3D geometry of the 3D contour fragments can be maintained. The time complexities of generation and matching are closely interrelated to the quality of the 3D geometry description. On the one hand, a higher number of mixture components leads to a more accurate representation of the shape of a contour fragment (see Figure 5.2). On the other hand, the time complexity of the learning stage increases due to the higher number of mixture components.



(a) (b) (c)

Figure 5.2: Gaussian Mixture Models (represented by blue ellipsoids) for a 3D contour with different numbers of mixture components K. (a) $K = 10$. (b) $K = 5$. (c) $K = 3$.

The representation with Gaussian Mixture Models can handle several problems that may occur:

- Noise/Outliers: When working with real data like reconstructed 3D contours from images instead of synthetic 3D models, noise (i.e. slightly displaced reconstructions) always plays an important role. If a few contour points are noisy, a reconstructed 3D contour fragment may no longer be identified as a single connected contour. With the representation by a set of probability density functions contours are always split into smaller parts. Parts that are only caused by noise/outliers are then suppressed during the learning stage.

- Linking: When working with contour fragments, linking of edges to long connected contours always plays a role. Different linking in different models may have a strong influence on a matching algorithm. With the representation as GMMs, mixture components can be flexibly grouped during learning, resulting in partitions (discriminative subset) that are adapted to particular linkings (see Figure 5.3).

- Deformation: By modeling with probability density functions, intra-class variability in form of shape deformations probabilities can be handled with a comparatively low computational effort.

It is worth mentioning that our approach may appear cumbersome: Initially, it seems we throw away the information on how contours are linked during the generation of the GMMs. Later, during the learning stage, we then reconstruct the spatial relation between GMMs. However, the initial reconstruction of long contours helps to provide good GMM representations of the shape. Furthermore, the linking in the learning stage does not solely depend on one specific object model. This is actually the strength of the concept as it can overcome problems like incorrect linking or contour splitting due to noise.

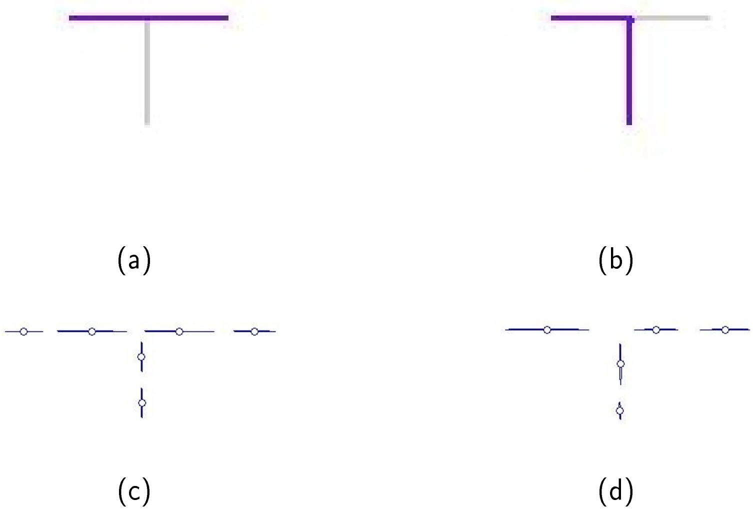


Figure 5.3: 2D example for different linking of the T-shape. (a) & (b) Differently linked edges that form longer contour fragments. (c) & (d) Representation as GMMs.

Our algorithm is applicable to all kind of 3D contour fragments, which describe the shape of an object. As mentioned in Section 4.2, there are many possibilities to generate 3D contour fragments. For the experiments in this thesis, we reconstruct '3D contour clouds' from stereo image sequences of several objects. Exemplary '3D contour clouds' of several objects are shown in Section 4.4.2.

Given a '3D contour cloud' $\mathcal{C}$ consisting of a set of 3D contour fragments $\mathcal{F}_l$

$$\mathcal{C} = \{\mathcal{F}_l, l = 1 \ldots N\}, \tag{5.1}$$

where N is the number of 3D contour fragments in a cloud and $\mathcal{F}_l = p_1, ..., p_n$ is a set of 3D points, we can fit a mixture of multivariate Gaussian distributions to each 3D contour fragment $\mathcal{F}_l$ using the standard Expectation-Maximization (EM) algorithm (see e.g. [XJ96]), so that each 3D contour fragment $\mathcal{F}_l$ is given by

$$\Theta^i_K(x) = \sum_{k=1}^{K} \alpha_k p(x|N(\mu_k, \Sigma_k)) \tag{5.2}$$

with probability

$$p(x|N(\mu_k, \Sigma_k)) = \frac{1}{(2\pi)^{\frac{n}{2}} \|\Sigma_k\|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\right) \tag{5.3}$$

Θ^i_K is the Gaussian Mixture Model for a fragment $\mathcal{F}_l$ and K is the number of mixture components with mean μ_k, variance Σ_k, and weight α_k. In our representation, K is chosen according to the length of the 3D contour fragment (with a maximum of $K = 10$). We call this representation '3D Gaussian Contour Model', where each 3D contour fragment of a '3D contour cloud' is represented by a GMM.

In many of the mentioned approaches it is assumed that the Gaussians are spherical and the covariance matrix is thus set to the identity matrix. In our case, the covariance matrix gives essential information about the orientation of a 3D contour fragment. We assume that each mixture component has the same weight, so that $\alpha_k = 1$. We do not decide on the basis of the weights, whether a mixture component is relevant for a category model or not. We assume that even a mixture component with an originally small weight in the GMM representation can be important for a category model. The decision about this importance is left to the learning stage.

5.4 Category Model: From 3D Gaussian Contour Models to a 3D Gaussian Contour Category Model

The category model - we call it 3D Gaussian Contour Category Model - is built from a set of 3D Gaussian Contour Models of specific objects of a category. We use a random feature selection algorithm to find a discriminant set of features for a category. Each feature is a set of probability densities - we call such a set a partition[1] - which is (weakly) discriminative for a category against another one. The distance measure is given by a similarity measure combining local shape information and 3D geometric information between densities. In the following we describe the test statistic (see Section 5.4.1) and the practical implementation in form of a random feature selection algorithm (see Section 5.4.2).

5.4.1 Test Statistic

Our approach uses a hypothesis test[2], to identify, if a given specific object O belongs to an object category C

$$\begin{aligned} H_0: &\quad O \in C \\ H_1: &\quad O \notin C \\ \\ \text{reject } H_0 \text{ if} &\quad SM(f_O \| g_C) > \gamma \end{aligned} \tag{5.4}$$

where g_C is a learned 3D Gaussian Contour Category Model and f_O is the 3D Gaussian Contour Model of the specific object. The test statistic TS =

[1] In this thesis, the term 'partition' is used for a discriminative subset of mixture components, not a partition in a strictly mathematical meaning.

[2] The statistical hypothesis, which has to be tested is the null hypothesis H_0. The alternative hypothesis is denoted by H_1. The test statistic TS is a statistic on whose value the null hypothesis will be rejected or not. The threshold of rejecting a null hypothesis is given by γ (notation is based on [Ros05]).

$SM(f_O \| g_C)$ is a similarity measure between two GMMs. On the basis of the threshold γ, the null hypothesis is rejected or not.

Most of the existing shape matching methods that use Gaussian Mixture Models are based on the Kullback-Leibler (KL) divergence between GMMs or, similarly, the Jensen-Shannon divergence (e.g. [WVRE08]) as a similarity measure SM. There exists no closed-form for the KL divergence between two Gaussian Mixture Models, but there exists a closed-form Kullback-Leibler divergence between two Gaussians $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$. It is given by

$$KL = \quad \tfrac{1}{2}(\log \tfrac{|\Sigma_2|}{|\Sigma_1|} + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_1 - \mu_2)^T \Sigma_2^{-1}(\mu_1 - \mu_2)) \tag{5.5}$$

Given two Gaussian Mixture Models f and g, where

$$f = \sum_{i=1}^{n} f_i = \sum_{i=1}^{n} \alpha_i p(x|N(\mu_i, \Sigma_i)) \tag{5.6}$$

and

$$g = \sum_{j=1}^{m} g_j = \sum_{j=1}^{m} \beta_j p(x|N(\mu_j, \Sigma_j)), \tag{5.7}$$

the following approximation of the KL-divergence between them has been suggested in [GGG03]:

$$KL(f \| g) \approx \sum_{i=1}^{n} \alpha_i \min_j (KL(f_i \| g_j) + \log \frac{\alpha_i}{\beta_j}) \tag{5.8}$$

With (5.8) we can rewrite the hypothesis test (5.4) as

$$\begin{gathered} \text{reject } H_0 \text{ if} \\ KL(f \| g) \approx \sum_{i=1}^{n} \alpha_i \min_j (KL(f_i \| g_j) + \log \frac{\alpha_i}{\beta_j}) > \gamma. \end{gathered} \tag{5.9}$$

With the simplification that all mixture components have the same weight $\alpha_i = \beta_j = 1$ and with $\gamma_0 = \gamma/n$, we further obtain

$$\begin{gathered} \text{reject } H_0 \text{ if} \\ \sum_{i=1}^{n} \min_j (KL(f_i \| g_j) - \gamma_0) > 0. \end{gathered} \tag{5.10}$$

In this approximation, the term $\min_j(KL(f_i \| g_j) - \gamma_0)$ might be very high when a part of an object is missing as there will be no g_j which is near to f_i. However, for our application we want to permit that certain parts of an object can be missing. Therefore, we first suggest using only discrete values for the similarity measure between two Gaussians:

$$\begin{aligned} &\text{reject } h_0 \text{ if} \\ &\sum_{i=1}^{n} \operatorname{sgn}(\min_j(KL(f_i \| g_j) - \gamma_0)) > -n + 2l \end{aligned} \tag{5.11}$$

where l is the number of Gaussians that are permitted to be missing in the sample. Please note that due to the discretization, the term $\operatorname{sgn}(\min_j(KL(f_i \| g_j) - \gamma_0))$, which is equivalent to $\min_j(\operatorname{sgn}(KL(f_i \| g_j) - \gamma_0))$, can be considered a hypothesis test: It is -1 (keep 'local' hypothesis h_0), if there is at least one Gaussian in the set g that is equal (with respect to a certain significance level) to f_i and $+1$ (reject h_0) otherwise. The 'global' hypothesis H_0 is rejected if the number of Gaussians of f that do not have a match in g is larger than a predefined number l. In our case the Kullback-Leibler divergence $KL(f_i \| g_j)$ is not a particularly useful similarity measure. The probability density function, especially the covariance matrix Σ_i of points, gives no evidence about the shape variability of 3D contours in an object category. It is rather a representation of the reconstruction quality; noise/outliers may have an important influence on the covariance matrix. Furthermore, it also reflects the curvature of the corresponding contour fragment, but this also depends on the number of mixture components which describe the 3D contour fragment. The major information about the orientation lies in the main principal axis which we will use for our similarity measure. Finally, in the Kullback-Leibler divergence a shift of the mean μ_i has more effect on the divergence measure than changes of the covariance matrix Σ_i.

Therefore, we propose a different similarity measure between two Gaussians, which is better suitable for our objective, i.e. the learning of a 3D category shape model. As mentioned above, the covariance matrix represents the orientation of a contour by its principal component and it can handle noise and reconstruction errors by the other two dimensions. Consequently, GMMs are better suitable for shape representations than just an approximation by straight lines.

In the remaining section we use the following notation: (R, T) denotes the global transformation between objects, (R_i, T_i) denotes the local transformation of a single mixture component (μ_i, Σ_i). (μ_j^O, Σ_j^O) and (μ_i^C, Σ_i^C) are mixture components of a specific object O and a model C. Further, we define v_{ab} as the difference vector between a pair of mixture components $((\mu_a^O, \Sigma_a^O), (\mu_b^O, \Sigma_b^O))$ of the object O and v_{xy} as the difference vector between a pair of mixture components $((\mu_x^C, \Sigma_x^C), (\mu_y^C, \Sigma_y^C))$ of the model C.

We can define the following hypothesis to test the similarity of Gaussians:

$$\begin{aligned} H_0 : &\quad (\mu_j^O, \Sigma_j^O) = (\mu_i^C + T_i, R_i \cdot \Sigma_i^C) \\ H_1 : &\quad (\mu_j^O, \Sigma_j^O) \neq (\mu_i^C + T_i, R_i \cdot \Sigma_i^C) \end{aligned} \tag{5.12}$$

$$\text{reject} \qquad H_0 \;\text{ if } TS_1 > \gamma_1 \text{ or } TS_2 < \gamma_2$$

where γ_1 and γ_2 are the thresholds and TS_1 and TS_2 are test statistics. As the GMM's may be subject to translation, we also need to find this translation. In order to achieve this, we define the test statistics (between a mixture component of the sample and a mixture component of the model) on pairs of mixture components. By means of TS_1 and TS_2, a translation between the two models is described. In doing so, the second component of the pairs essentially defines the translation and is used as kind of an anchor point for aligning the two models which are tested on similarity. By testing the angles between anchor point of model and sample we further reduce the space for possible alignments and thus the computational cost.

TS_1 is the test statistic for the pairwise difference of pairs of mixture components

$$TS_1 = \|v_{ab}^O - v_{xy}^C\|_2. \tag{5.13}$$

with

$$v_{ab}^O = \mu_a^O - \mu_b^O \qquad \text{and} \qquad v_{xy}^C = \mu_x^C - \mu_y^C. \tag{5.14}$$

Let e_a^O (e_b^O) be the principal eigenvectors of Σ_a^O (Σ_b^O) and e_x^C (e_y^C) the principal eigenvectors of Σ_x^C (Σ_y^C), then TS_2 is given by the scalar product between the eigenvectors

$$TS_2 = e_i^O \cdot e_j^C \qquad \text{for} \qquad i = a, b \qquad \text{and} \qquad j = x, y. \tag{5.15}$$

With this, we can introduce a discrete similarity measure

$$SM(f_i, g_j) = \begin{cases} -1 & \text{if } TS_1 < \gamma_1 \text{ and } TS_2 > \gamma_2 \\ 1 & \text{otherwise.} \end{cases} \tag{5.16}$$

Consequently, $SM(f_i, g_j) = -1$ means that the two pairs of mixture components have approximately the same relative orientation ($TS_2 > \gamma_2$) and the same relative position ($TS_1 < \gamma_1$). Otherwise, H_0 (see 5.12) will be rejected. With this similarity measure instead of the Kullback-Leiber divergence equation (5.11) is modified to

$$\begin{gathered} \text{reject } H_0 \text{ if} \\ \sum_{i=1}^{n} \min_j (SM(f_i, g_j)) > -n + 2l \end{gathered} \tag{5.17}$$

Invariance Properties

Specific objects of a category may differ by a global rigid transformation (R, T) between their 3D Gaussian Contour Models and a local shape transformation (R_i, T_i) between mixture components. Then the global and local transformation can be described by

$$(\mu_j^O, \Sigma_j^O) = (R \cdot \mu_i^C + T_i + T, R \cdot R_i \cdot \Sigma_i^C). \tag{5.18}$$

To be insensitive to the global translation, we consider pairwise relations between mixture components, which is given by

$$v_{ab}^O = \mu_a^O - \mu_b^O. \tag{5.19}$$

Because of

$$\begin{aligned} v_{xy}^C &= (\mu_x^C + T + T_x) - (\mu_y^C + T + T_y) \\ &= (\mu_x^C + T_x) - (\mu_y^C + T_y), \end{aligned} \tag{5.20}$$

the global rigid translation T can be eliminated. To be insensitive to a global rotation we may also consider pairs of mixture components. We can compute a global rotation R between a pair of the object and the pair of the model by estimating the rotation between their principal eigenvectors. However, for performance reasons we avoid to compute the global rotation for each pair. Instead, we did as preprocessing step a coarse alignment of the rotation of the whole 3D models based on the longest elongation to reduce the matching space to four possibilities.

By normalizing the relative pairwise difference TS_1 by a scaling factor we are moderately scale invariant. Let B^O be the bounding box diagonal of the object model and B^C be the bounding box diagonal of the category model. Then TS_1 can be rewritten as

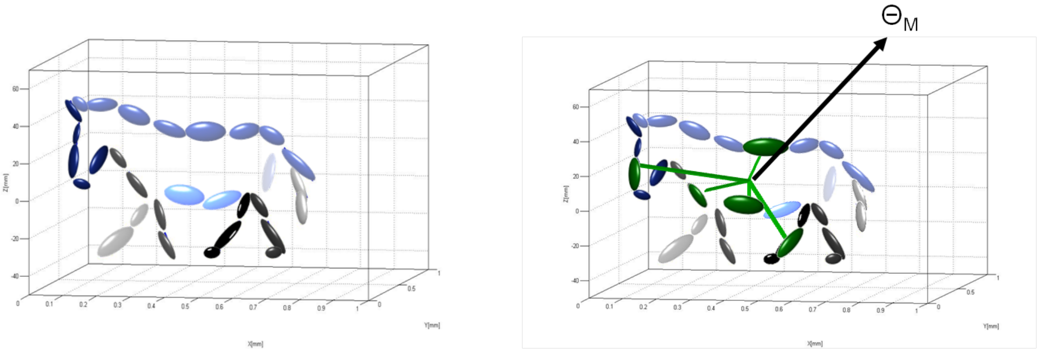
$$TS_1 = \left\| \frac{v_{ab}^O}{B^O} - \frac{v_{xy}^C}{B^C} \right\|_2 . \tag{5.21}$$

Partition model

The presented approach is suitable when we have more or less rigid objects that are deformed only by small translations and rotations of the contours. However, an object may actually be composed of several parts. For example, the head of a horse will be a rather rigid object but can have significant displacement with respect to other parts of the animal. Therefore, we do not consider the GMM of a 3D model in a whole, instead we randomly select discriminative subsets of mixture components which we call partition (see Figure 5.4). Such a discriminative subset lets us handle each part of an object independently or in context to other parts. The partitioning model Θ_M of a subset of probability functions of size M is given by:

$$\Theta_M = \sum_{m=1}^{M} \alpha_m N(\mu_m, \Sigma_m) \qquad \text{with} \qquad \Theta_M \subset \bigcup_{i=1}^{N} \Theta_K^i . \tag{5.22}$$

A 3D Gaussian Contour Category Model consists of sets of partitions which are learned using the method described in Section 5.4.2. Whether a partition belongs to a category or not is decided on the basis of equation (5.17) such that we define a hypothesis test for each partition Θ_M.



(a) (b)

Figure 5.4: Simple illustration of the representation model of a horse including a partition. (a) 3D Gaussian Contour Model where each 3D contour fragment is represented by a GMM Θ^i_K. (b) Selection of one partition Θ_M with $M = 5$. Selected partition is drawn in green.

5.4.2 Learning by Random Feature Selection

In the previous section it was shown how we can test a sample object on a model of a category. In this section we describe how we extract a model of a category from a number of sample objects. In our practical implementation we use a random feature selection algorithm (see Figure 5.5) for the partitioning of probability density functions. Random feature selection is a simple method for reducing the number of features to a discriminative subset of features. In the random feature selection, we randomly select partitions of probability density functions and verify if they are discriminant on training data by testing the hypothesis above for pairs of the partitions. Before starting the random feature selection, the 3D models are aligned on the basis of their bounding box in that way that all animals show in the same direction. For performance issues it is useful to do a preprocessing by considering the position of probability densities on the object. The random selection algorithm stops when a number of iterations or a number of selected partitions is reached.

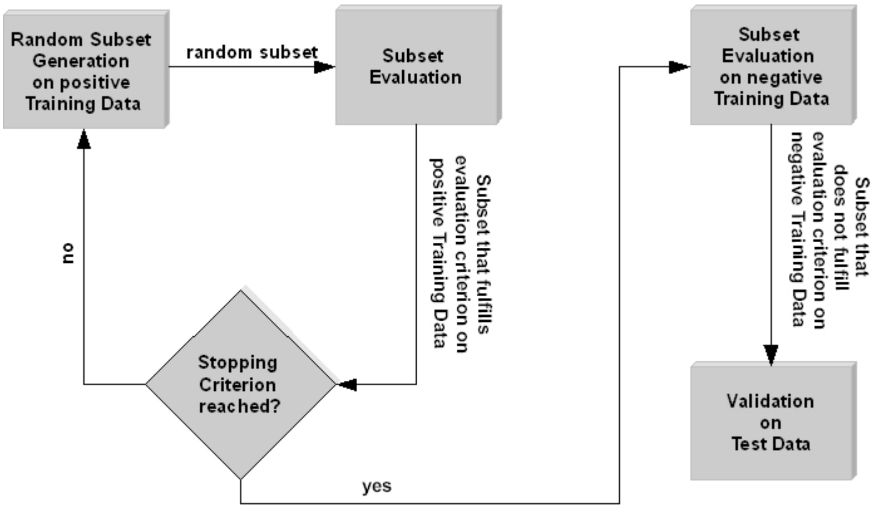

Random Subset Generation on positive Training Data
random subset
Subset Evaluation
Subset that fulfills evaluation criterion on positive Training Data
no
Stopping Criterion reached?
yes
Subset Evaluation on negative Training Data
Subset that does not fulfill evaluation criterion on negative Training Data
Validation on Test Data

Figure 5.5: Random feature selection algorithm with validation.

By the partitioning of probability density functions we can generate additional constraints about their distribution on the object:

- Locality constraint: The selected partitions of probability density functions should be distributed in a local environment on the object. By this constraint we may represent local features on the object, e.g. the leg or the head of an animal.
- Uniformity constraint: The selected partitions of probability densities should be distributed on the object, so that no two density functions should be in a local neighborhood. This constraint yields partitions that are distributed on the object such that each density may represent one part of the object.
- No constraint: The partitions are selected randomly without restrictions on the spatial distribution on the object.

In the random feature selection algorithm we first test if a selected partition of densities is discriminative on the positive training data, afterwards, if it is discriminative against the negative data. Finally, we obtain a subset of partitions of probability density functions.

The classification output $h^q(O)$ of a partition Θ_M^q on a 3D model O is given by

$$h^q(O) = \begin{cases} 0 & \text{if } \sum_{m=1}^{M} \min_i(SM(f_i, g_m)) > -M + 2l \\ & \forall f_i \in O \\ 1 & \text{otherwise} \end{cases} \tag{5.23}$$

where M is the number of mixture components of the subset and q defines the q^{th} subset. We check for all components of the subset $\forall g_m \in \Theta_M^q$ if there is a corresponding component in the model O allowing l missing components. The output of the whole detector $H(O)$ then is

$$H(O) = \frac{1}{Q} \sum_{q=1}^{Q} (h^q(O)). \tag{5.24}$$

5.5 Experimental Evaluation

We evaluate our 3D object categorization system on four categories of our GRAZ-STEREO-BASE-xx dataset. On the one hand, we show that the random feature selection algorithm applied to positive training data can be used to reduce outliers in 3D models. On the other hand, - and this is the main part of this section - we show several experiments and results to achieve several 3D Gaussian Contour Category Models. The difficulties in our dataset are the large intra-class variability and the small inter-class difference between horses and cows, but also dogs. Therefore, we show the possibilities to distinguish between the categories ‘horse’ and ‘cow’, as well as between the categories ‘dog’ and ‘horse’. We also learn a model for the category ‘car’ against ‘horse’, ‘cow’, and ‘dog’, where the inter-class difference is large.

5.5.1 Cross Validation

We use a k-fold cross validation to evaluate our experiments. Given a dataset D, this dataset is split into k subsets of approximately equal size. In our case

$D = \{D_1^p, ..., D_{S_1}^p, D_1^n, ..., D_{S_2}^n\}$ of positive and negative 3D models and $k = S_1 + S_2$ subsets. Consequently, one subset is one 3D model. We train the classifier $S_1 \times S_2$ times. In each iteration $t \in 1, ..., S_1 \times S_2$, we leave out one positive and one negative 3D model. So we train on $D \backslash \{D_i^p, D_j^n\}$ $\forall i = 1, ..., S_1$ and $\forall j = 1, ..., S_2$ and test on D_i^p and D_j^n. In the cross validation we then build the average over the results for the positive and the negative test data.

5.5.2 Outlier Reduction

I mentioned in Section 5.3 that outliers in form of wrongly reconstructed contour points always play an important role in 3D reconstruction. Noise is represented in the probability density function. But there exist also outliers that result from the reconstruction process. The corresponding probability density functions do not actually belong to the shape of the object. By running the random feature selection only on the positive training data, we can observe that noisy parts can be substantially suppressed. Given nine horses, we randomly select 1000 partitions of five mixture components, where in each round of the cross validation a horse is randomly selected from a set of eight horses. Figure 5.6(a) and Figure 5.6(b) show the 3D Gaussian Contour Models of two horses. For visualization, the probability densities are drawn by 3D lines given by their mean and the principal eigenvector. We can see that outliers exist, which result from the '3D contour cloud' generation. In Figure 5.6(c) and Figure 5.6(d), we can see selected partitions of size 5 from the two horses which were discriminant for all other horses. We can see, that most of the discriminative probability densities are located on the head, the back, and the tail of the horses. Fewer densities are located on the legs, because of different arrangements of the legs in the 3D horse models. Outliers have been significantly reduced.

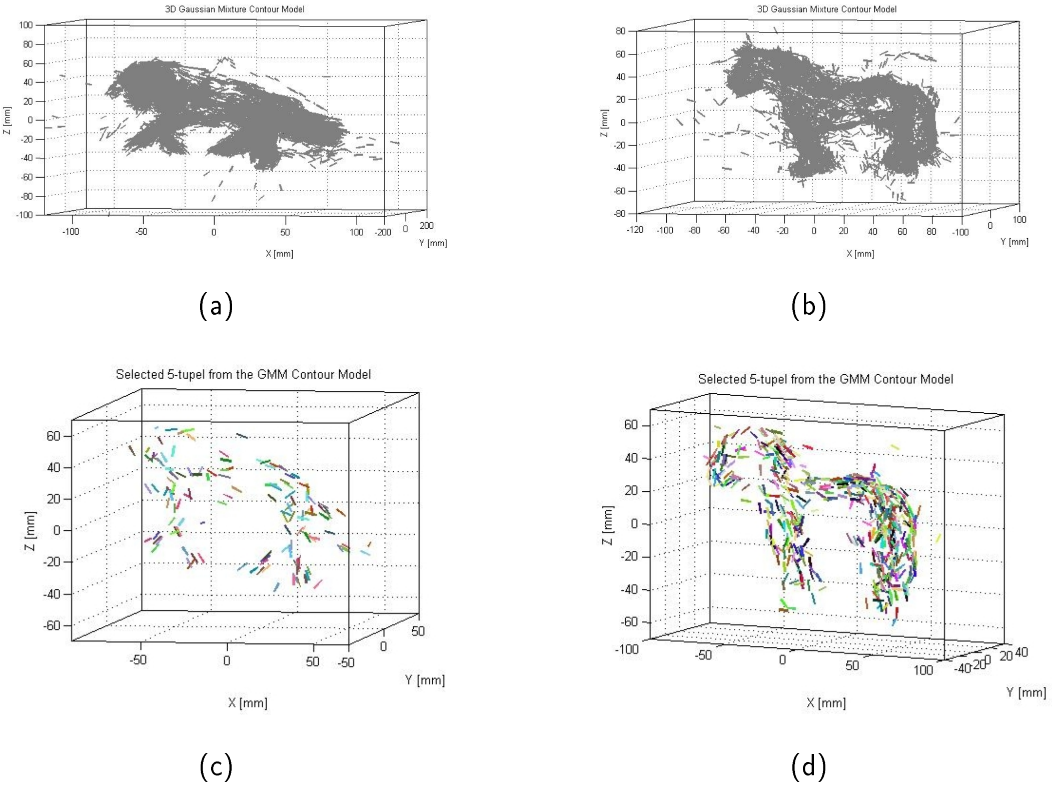


Figure 5.6: (a) & (c) 3D Gaussian Contour Model of a horse and 28 partitions which have been selected. (b) & (d) 3D Gaussian Contour Model of another horse and 128 partitions which have been selected. We see, that the number of outliers is significantly reduced. A partition of size M = 5 is represented by the same color.

5.5.3 3D Object Categorization on the GRAZ-STEREO-BASE-xx

Now, we evaluate our probabilistic 3D object categorization framework for the task of 3D object categorization. Our aim is to learn one category against one or several others. The difficulties are the small inter-class difference between the animals in our GRAZ-STEREO-BASE-xx dataset. We show that we are able to handle this small inter-class difference as well as the large intra-class variability.

'Horse' against 'cow'

The aim of the following experiments is to learn a 3D Gaussian Contour Category model 'horse' against 'cow'. The challenge in this learning experiment is the small inter-class difference between horses and cows. Our object categorization system is validated using the cross validation scheme described in Section 5.5.1. We did several experiments with different kinds of constraints (see Section 5.4.2) on the random selection of partitions. We can summarize the experiments as follows:

- Experiment HORSE 1: We randomly select partitions of size M = 3 or M = 5 of probability density functions from the 3D Gaussian Contour Models of randomly selected horses, where we use the uniformity constraint (see Section 5.4.2). We perform a cross validation experiment where the results of this experiment are summarized in Table 5.1.

- Experiment HORSE 2: We randomly select partitions (M = 5 or M = 7) of probability densities from the 3D Gaussian Contour Models of randomly selected horses, where we use no constraint. The results of this experiment are summarized in Table 5.2.

- Experiment HORSE 3: We randomly select partitions (M = 5) of probability density functions from the 3D Gaussian Contour Models of randomly selected horses, where we use the locality constraint. The results of this experiment are summarized in Table 5.3.

For these experiments we typically choose $\gamma_2 = 0.98$ and $l = 0$. For Experiment HORSE 1 and Experiment HORSE 2 $\gamma_1 = 0.2$, for Experiment HORSE 3 $\gamma_1 = 0.1$. Too small γ_1 and γ_2 do not handle intra-class variability, too large γ_1 and γ_2 do not handle inter-class difference. The result tables (Table 5.1, Table 5.2, Table 5.3) contain seven entries for each experiment. The average result of the cross validation for the positive training set 'horse' and the negative training set 'cow' is shown in the first (μ_{horse}) and the third (μ_{cow}) row. Assuming a normal distribution we can also compute the standard deviations σ_{horse} and σ_{cow}, as well as a classification threshold c_thresh on whose basis we can decide if a 3D test model is a ‘horse’ or a ‘cow’. c_error_{horse} and c_error_{cow} are the classification errors which represent the true positive rate and the false positive rate. Figure 5.7 shows a graphical representation of the results of Experiment HORSE 1 - partition (M = 3). We can see that the two categories are well separable. Figure 5.10 shows the Receiver Operating Characteristic (ROC) curves for all three experiments[3]. As the results show, Experiment HORSE 1 and Experiment HORSE 2 perform better than Experiment HORSE 3 with the local features. In Experiment HORSE 1 and Experiment HORSE 2, the classification errors are $\leq 21\%$. The locality constraint seems to be less useful than a uniform distribution. This result is not very surprising. Local features often have similar shape which is not category specific, e.g. legs of horses and cows. Only combinations with other local features (e.g. leg and head of an animal) could give more discriminance. Moreover, we saw that we have to learn longer for Experiment HORSE 3 for the same number of partitions than for Experiment HORSE 1 or Experiment HORSE 2. This is also true for learning smaller partitions e.g. Experiment HORSE 1 - partition (M = 3).

Figure 5.8 shows an example of a learned 3D Gaussian Contour Category Model ‘horse’ from one training step of Experiment HORSE 1 - partition (M = 5). For this model we randomly choose 1000 partitions, where 135 are found to be discriminative for horses and not for cows. The Gaussian Contour Category Model in Figure 5.8 has 135 partitions from eight horses. All partitions are drawn in one model without special aligning. We can see that discriminant probability density functions are located mainly on the head, the back and the tail, fewer are on the legs which is due to different arrangements of legs on different training models. We can see a similar behavior for Experiment HORSE 2. Figure 5.9 shows a learned 3D Gaussian Contour Category Model ‘horse’ for Experiment HORSE 2 - partition (M = 7). The distribution of the probability density functions is similar to that of Experiment HORSE 1, on head, back, and tail. However, most of the densities are located on the head of the horse.

[3] All ROC curves in this work (Figure 5.10, Figure 5.12, and Figure 5.15) were generated by varying the classification threshold c_thresh.

Table 5.1: Experiment HORSE 1: partition selection based on the uniformity constraint .

	Partition (M = 3)	**Partition (M = 5)**
μ_{horse}	0.7380	0.6781
σ_{horse}	0.1965	0.2328
μ_{cow}	0.3141	0.2481
σ_{cow}	0.2432	0.2191
c_thresh	0.5248	0.4632
c_error_{horse}	0.1390	0.1788
c_error_{cow}	0.1931	0.1635

Table 5.2: Experiment HORSE 2: partition selection based on no constraint .

	Partition (M = 5)	**Partition (M = 7)**
μ_{horse}	0.7066	0.6648
σ_{horse}	0.2225	0.2271
μ_{cow}	0.2836	0.2325
σ_{cow}	0.2555	0.2302
c_thresh	0.4912	0.4485
c_error_{horse}	0.1660	0.1704
c_error_{cow}	0.2082	0.1741

‘Dog’ against ‘horse’

The aim of the following experiments is to learn a 3D Gaussian Contour Category Model ‘dog’ against ‘horse’. Subjectively, the inter-class difference between the category ‘dog’ and the category ‘horse’ is larger than between ‘horse’ and ‘cow’. Again, our object categorization system is validated using the cross validation scheme described in Section 5.5.1. In the Experiment DOG we randomly select partitions of size M = 5 from 3D Gaussian Contour Models of randomly selected dogs based on the uniformity constraint. As parameters, we choose $\gamma_2 = 0.98$ and $l = 1$. The results of this experiment are summarized in Table 5.4. As the results

Table 5.3: Experiment HORSE 3: partition selection based on the locality constraint .

	Partition (M = 5)
μ_{horse}	0.5723
σ_{horse}	0.3304
μ_{cow}	0.2619
σ_{cow}	0.2907
c_thresh	0.4462
c_error_{horse}	0.3514
c_error_{cow}	0.2631

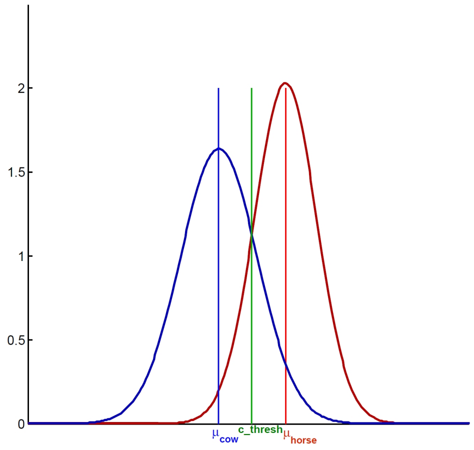


Figure 5.7: Graphical representation of the results of Experiment HORSE 1: Partition (M = 3). The two categories are well separable.

show, we achieve classification errors $\leq 11\%$ for the Experiment DOG. Figure 5.11 shows a graphical separable representation of the results of Experiment DOG - partition (M = 5). We see that dogs and horses are better a than horses and cows. As the features were initially selected such that they are frequently observed for the category 'dogs' they only have random matches on horses. Therefore, it is not surprising that the Gaussian bell curve for the horses is quite flat. Figure 5.12 shows the Receiver Operating Characteristic (ROC) curve for this experiment.

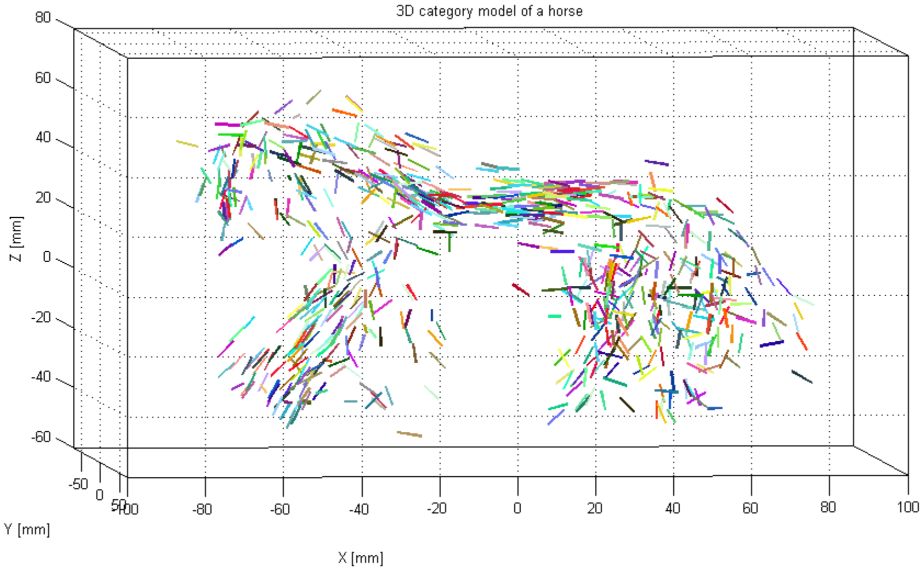


Figure 5.8: 3D Gaussian Contour Category model ‘horse’ with 154 partitions from eight horses of size M = 5 of Experiment HORSE 1. The probability densities are drawn by 3D lines given by their mean and the principal eigenvector.

Figure 5.13 shows an example of a 3D Gaussian Contour Category Model for the category ’dog’ as it has been learned in the training with partitions of size M = 5. For this model we initially generated 50000 partitions, 1720 thereof are found to be discriminative for dogs versus horses. The 3D Gaussian Contour Category Model in Figure 5.13 shows the 1720 partitions which are obtained from eight dog models. A single partition comprises GMMs form one dog model only; no special aligning has been applied. We can see that discriminant probability densities are located mainly on the head, the tail, and the legs, fewer are on the body. Although, the number of discriminant partitions is higher than, e.g. in the 3D Gaussian Contour Category Model ‘horse’ (Figure 5.8), the ’dog’ model seems comparatively sparser. This is because one or more mixture component have been selected for multiple partitions.

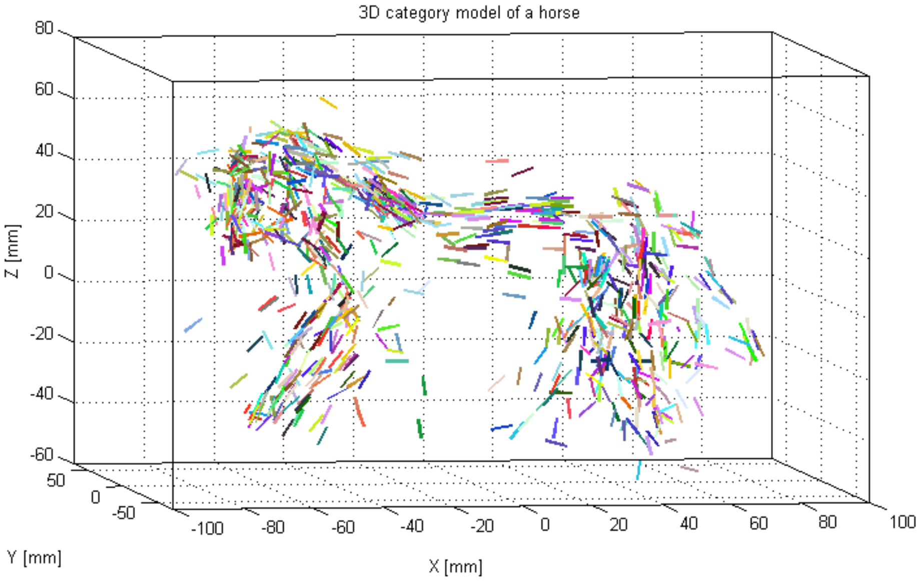


Figure 5.9: 3D Gaussian Contour Category Model 'horse' with 142 partitions (M = 7) from eight horses of Experiment HORSE 2. The probability densities are drawn by 3D lines given by their means and their principal eigenvectors.

'Car' against 'cow', 'horse', and 'dog'

On the one hand, the aim of this experiment is to show how the classification error decreases when the inter-class difference increases. On the other hand, the aim is to demonstrate scale invariance of our approach. Therefore, we perform an experiment where we scale the car models by factors from 1/10 to 50 and the horse models by a factor from 1/10 to 50. The size of the dog models and cow models are unchanged. We run a leave-one-out-test (see Section 5.5.1) where we randomly select partitions of size M = 5 from the 3D Gaussian Contour Models of randomly selected cars without any constraint. The results of this experiment are summarized in Table 5.5.

For the experiment we choose $\gamma_1 = 0.1$, $\gamma_2 = 0.99$, and $l = 0$. The results are summarized in Table 5.5. The average results of the cross validation for the positive training set 'car' and the negative training set (containing the categories 'cow', 'horse' and 'dog') are shown in the first (μ_{car}) and the third (μ_{nocar}) row,

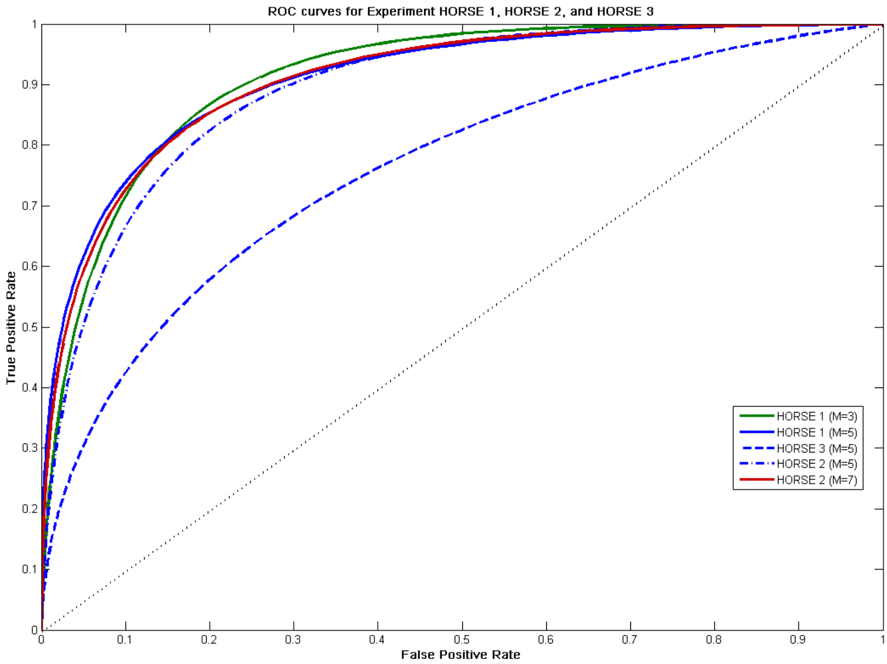


Figure 5.10: ROC curves for Experiment HORSE 1, HORSE2, and HORSE3.

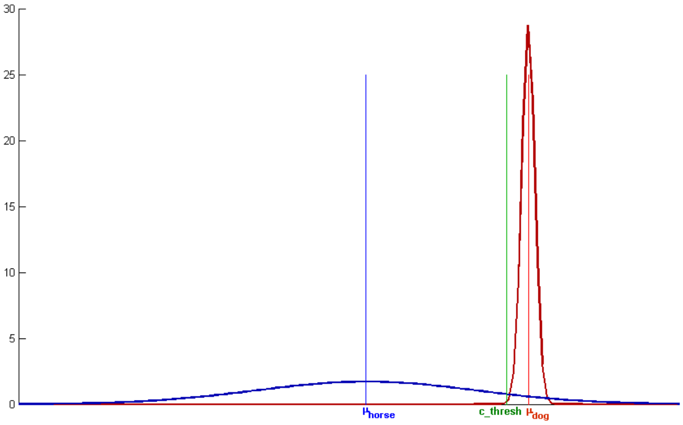


Figure 5.11: Graphical representation of the results of the experiment 'dog' against 'horse': Partition (M = 5) with classification threshold $c\_thresh = 0.9345$. The two categories are well separable with a classification error $c\_error_{dog} = 4.10^{-4}$ and $c\_error_{horse} = 0.1020$.

respectively. Assuming a normal distribution we can also compute the standard deviations σ_{car} and σ_{nocar}, as well as a classification threshold c_thresh

Table 5.4: Experiment DOG: partition selection based on uniformity constraint.

	Partition (M = 5)
μ_{dog} σ_{dog}	0.9807 0.0139
μ_{horse} σ_{horse}	0.6367 0.2335
c_thresh	0.9345
c_error_{dog}	0.0004
c_error_{horse}	0.1020

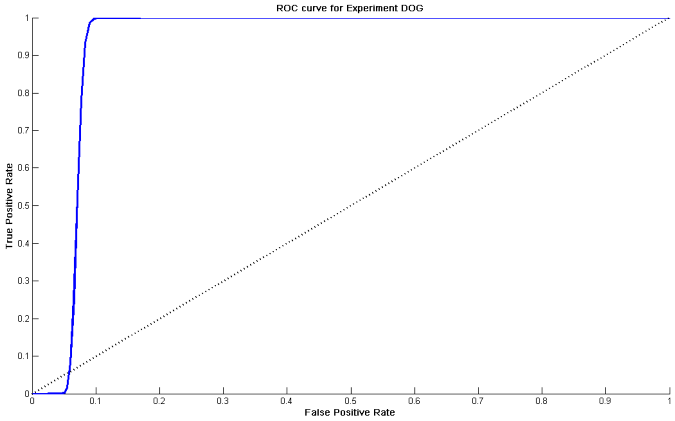


Figure 5.12: ROC curve for Experiment DOG (M=5).

between the categories ‘car’ and ’no car’. The classification errors c_error_{car} and c_error_{nocar} define at which percentage we would obtain an incorrect categorization. Figure 5.14 shows a graphical representation of the results of Experiment CAR, and Figure 5.15 shows the ROC curve for this experiment. We can see that the category ‘car’ is clearly separated from the other three categories with a classification error $\leq 1.2\%$. One reason for the good performance is the larger inter-class difference between cars and the other categories than in the experiments before (‘horse’ against ‘cow’ and ‘dog’ against ‘horse’). The scaling of the cars has no significant influence on the results when using the scale invariant test statistic.

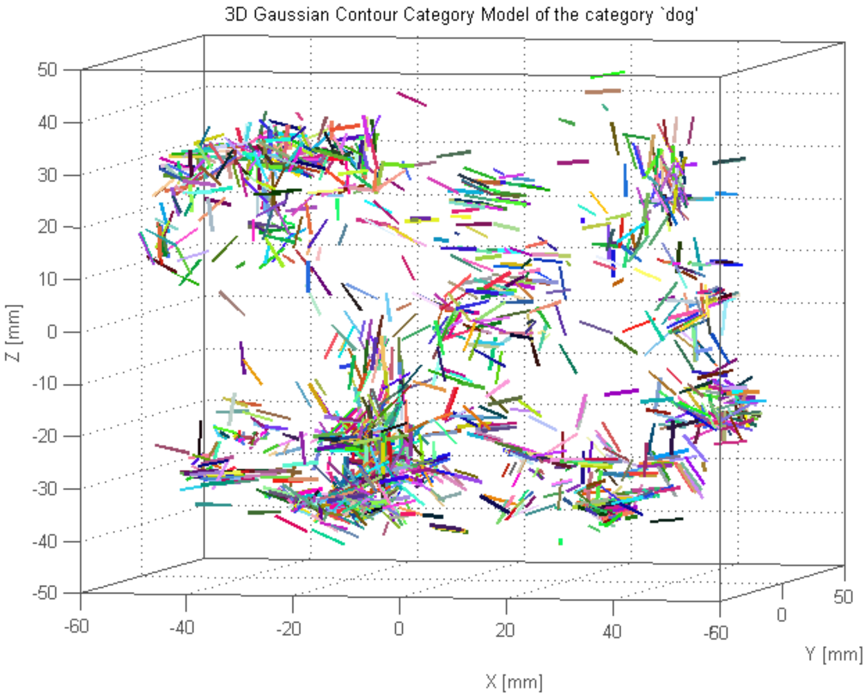


Figure 5.13: 3D Gaussian Contour Category Model 'dog' model with 1720 partitions (M = 5) from eight dogs. The probability densities are drawn by 3D lines given by their means and their principal eigenvectors. We can clearly identify the overlap of several 3D Gaussian Contour Models. Despite this overlap, regions corresponding to legs, heads or tails can still be identified by a human observer.

Figure 5.16 shows an example of a learned 3D Gaussian Contour Category Model 'car' from one training step of Experiment CAR. For this model we randomly choose 50000 partitions, where 1500 are found to be discriminative for the category 'car' and not for the other three categories. All partitions are drawn in one model without special aligning except the car sizes which were different for these experiment. We can see that discriminant probability density functions are located mainly on the car silhouette and the wheels.

Bregman Divergence vs. Our Approach

We motivate our similarity measure by the fact that the Kullback-Leibler and similar divergences are not applicable for our goal - learning a 3D category model

Table 5.5: Experiment CAR: partition selection based on no constraints.

	Partition (M = 5)
μ_{car} σ_{car}	0.7422 0.1620
μ_{nocar} σ_{nocar}	0.0777 0.1267
c_thresh	0.3770
c_error_{car}	0.0121
c_error_{nocar}	0.0091

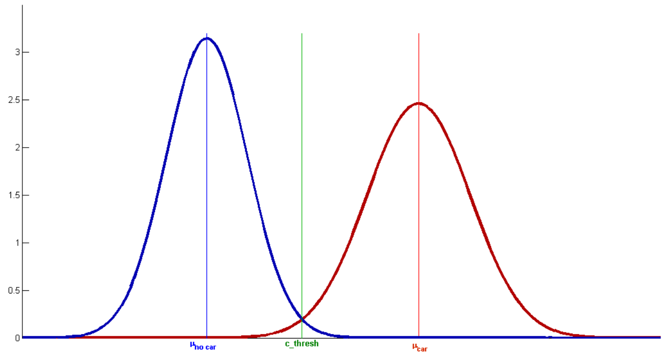


Figure 5.14: Graphical representation of the results of the scale invariance Experiment CAR: partition (M = 5). The classification threshold $c\_thresh = 0.3770$. The two categories are well separable with a classification error $c\_error_{car} = 0.0121$ and $c\_error_{nocar} = 0.0091$.

- for several reasons. The Kullback-Leibler divergence is a special case of a Bregman divergence. Liu et al. [LViAN10] introduce a Total Bregman divergence for shape matching and shape retrieval in an MPEG-7 dataset of 2D shapes. In our experiment we apply this Total Bregman divergence described in [LViAN10] to object categorization and explain the difficulties.

The Bregman divergence is often used to compute cluster centers. Liu et al. have adapted this idea and compute a cluster representative - the so called t-center - of a set of shape models. The t-center is the best model that minimizes the l1-norm Total Bregman divergence between itself and the other shape models.

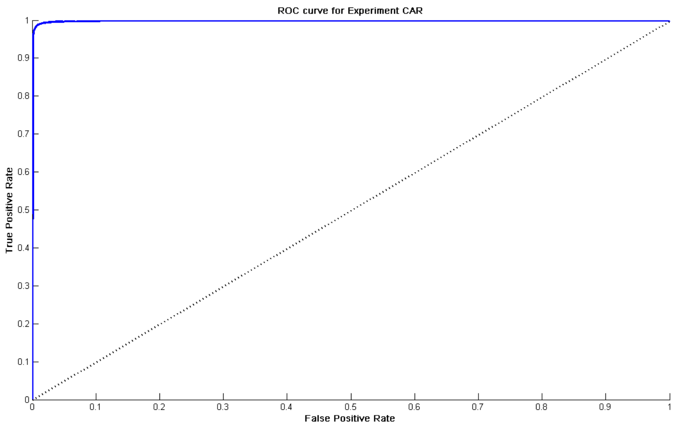


Figure 5.15: ROC curve for Experiment CAR (M=5).

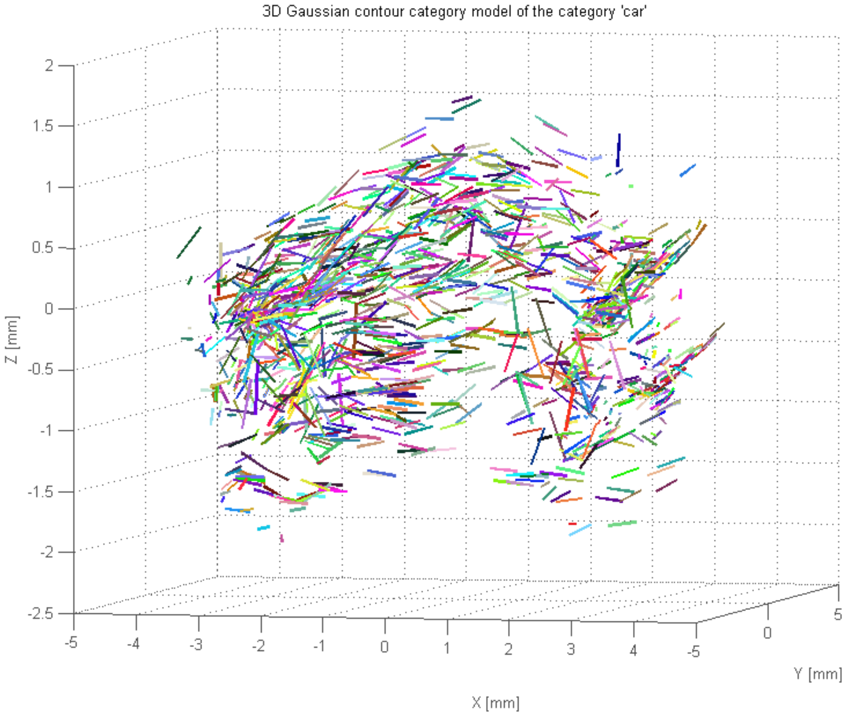


Figure 5.16: 3D Gaussian Contour Category Model 'car' model with 1500 partitions (M = 5) from six cars of Experiment CAR. The probability densities are drawn by 3D lines given by their means and their principal eigenvectors. For illustration we did a coarse alignment of the car sizes which were scaled differently for this experiment.

Moreover, they define the tSL (total square loss divergence) between two GMMs to compute the dissimilarity between them.

In our experiments we did a leave-one-out test on the same dataset which was used for Experiment HORSE 1 (M = 3). In each iteration of the cross validation we leave out one 3D model and compute the t-center of the remaining horses and cows as representative of the class - 3D category model. This t-center we test on the remaining model using the tSL and compute the average recognition rate for all models.

In our experiments we have to deal with two challenges: Time complexity and robustness. The time complexity of computing the t-center and the tSL is $O(n^2)$ where n is the number of mixture components. Assuming a 3D model dataset of nine horses where each horse has 5000-10000 mixture components, we have in each leave-one-out iteration a t-center of 40000 to 80000 mixture components. It is impossible to use the Total Bregman divergence for such large models, as the computational effort would be excessive. Therefore, in order to reduce the computation time we have to reduce the size of the datasets. Therefore, we randomly sample a subset of 500 mixture components from each 3D Gaussian Contour Model and evaluate the algorithm using the cross validation. We iterate the experiments several times. During these experiments we make the observation that the algorithm is very sensitive to the actual choice of mixture components. If there are outliers in at least one of the 3D Gaussian Contour Models, we get no reasonable results because of too high tSL dissimilarities.

Table 5.6 summarizes the comparison of the recognition rate of our approach with the Total Bregman divergence. Due to the sensitivity to noise and outliers we report the best recognition rate that we achieved in series of seven experiments for the Total Bregman divergence.

5.6 Discussion

We performed experiments on our own dataset GRAZ-STEREO-BASE-xx. Because of the size of the dataset, several questions arise:

Table 5.6: Comparison of recognition rates.

	Recognition rate (%)
Our approach	85.9
Total Bregman divergence [LViAN10]	55.6

- How scalable is the presented approach with respect to a large number of categories?
- Is the approach applicable to non-rigid objects?

To answer these questions: In my opinion it is scalable to a large number of objects independent if they are rigid or non-rigid objects (except maybe for some categories like humans with all their very different poses). The number of categories mainly depends on the size of the dataset, which was in my experiments - due to the underlying Structure-and-Motion framework - restricted to a lab environment. If the underlying Structure-and-Motion framework would allow go outside and to identify and model moving objects, e.g. as in a multi-body Structure-and-Motion framework [SSP08], such a dataset of 3D shape models could be generated for a large number of different objects of different categories. The experiments show that it is possible to differ between categories with small inter-class difference although we have a very small dataset of objects.

We use a simple random feature selection algorithm to find a discriminative subset of features for a category. We saw in the results, that non-rigid parts of objects as legs are less often selected for the 3D category model than rigid parts. Therefore, we can make the assumption that our approach currently favors more or less rigid objects. There are several concepts on how to improve this behavior: First, it is known that sophisticated learning algorithms are often capable to partially deal with non-rigid object. Our current learning algorithm is still rather simple. Second, a hierarchical approach might be used, i.e. the relation between partitions due to motion might be exploited: Let us assume we have the category 'horse'. The moving parts of the horse are the head, the tail, and the legs. Each of these moving parts has a restricted movement, e.g. the head can move left, right, up and down. Each movement causes a change in local shape information and changes in spatial relation to other shapes of the object. But there is a relation between this local shape information, e.g. orientation of a mixture component. Thus, spatial and temporal relations to other mixture components could be learned.

5.7 Conclusion

I have presented a new approach for learning a 3D Gaussian Contour Category Model using a probabilistic framework based on Gaussian Mixture Models. Instead of modeling the whole shape by a GMM and computing a divergence between shapes, I represent 3D contour fragments by GMMs and apply a partitioning on them, i.e. our category model is represented by a set of small GMMs. In the similarity measure I combine local shape information and global 3D geometric information. The experiments show that it is possible to build one single, pose-invariant 3D model per category. This is in contrast to building one model per significant view as it is the case in 2D shape based models. The results demonstrate that we can learn one category against another one even when the inter-class difference is small. Furthermore, the experiments show that global partitions, i.e. partitions where the used components are spread over the entire object, are better suitable for a category model than local features.

A further question arises: Can we use such a probabilistic 3D model for pose estimation in 2D images (e.g. [LS10]) or even for categorization in 2D images? This would overcome known problems of standard 2D categorization like sensitivity to pose/view changes. An example pose estimation algorithm for 2D images is shown in the next chapter.

6

3D Object Category Pose from 2D Images using 3D Contour Models

(How) can we use a 3D category model for pose estimation? Pose hypotheses are important for many computer vision applications such as active object categorization. This chapter presents a novel pose estimation algorithm, which computes pose hypotheses of objects in 2D images on the basis of 3D Gaussian Contour Category Models. Chapter 5 already described this probabilistic framework for learning a 3D category model based on 3D shape information. There, we use a similar principle for pose estimation. The algorithm consists of several parts: the representation model for a 2D input image, the 2D aspect models and the voting procedure. Each 2D contour fragment of the image is represented as a Gaussian Mixture Model (see Section 6.3). For the comparison, the algorithm computes probabilistic 2D aspect models from the 3D Gaussian Contour Category Model (see Section 6.4). The voting procedure combines geometric information and local shape information in a novel similarity measure, which we introduce in a hypothesis testing framework (see Section 6.5). Experiments on several poses of the known ETH-80 dataset for the categories 'horse', 'cow', and 'dog' as well as on the 3D object category dataset for the category 'car' demonstrate the applicability of shape information for pose estimation. For our experiments we learn

the 3D Gaussian Contour Category Models for pose estimation just from the toy objects of our GRAZ-STEREO-BASE-xx dataset, whereas the testing is done on the ETH-80 dataset and the 3D object category dataset. Here, our approach differs from many existing pose estimation approaches, which perform learning and testing on the same data by splitting the dataset in a training and a test set.

6.1 Introduction

Objects differ significantly when they are viewed from different poses. Especially shape-based categorization systems [LHS07, OPZ06, SBC08] are sensitive to pose/view changes, because the visual rim changes significantly on the object (see Figure 6.1). We use the following idea for our pose estimation algorithm: When shape information of an object changes when it is viewed in different poses, we can use the shape information to compute a pose hypothesis for an object in a 2D image.

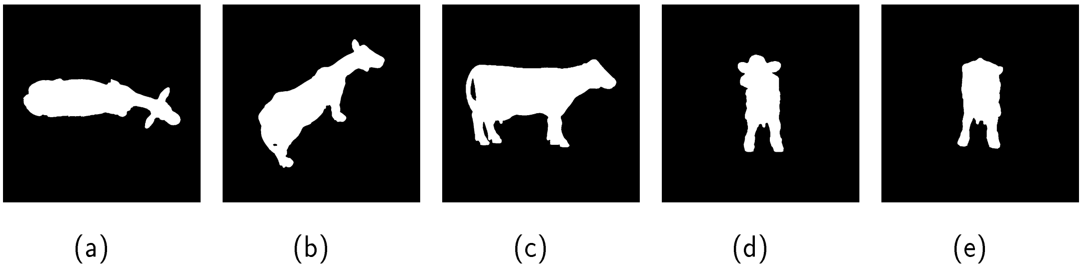


Figure 6.1: Visual rim of an ETH-80 cow seen from different views. (a) azimuth $\alpha = 0^o$ and elevation $\lambda = 0^o$. (b) $\alpha = 35^o$ and $\lambda = 315^o$. (c) $\alpha = 90^o$ and $\lambda = 0^o$. (d) $\alpha = 90^o$ and $\lambda = 90^o$. (e) $\alpha = 90^o$ and $\lambda = 270^o$.

Most of the current pose estimation algorithms are appearance based (see Section 6.2). In contrast, our approach for pose estimation on 2D images is purely *shape-based* in that sense that we use the visual rim of objects and inner contour fragments as shape information. Our algorithm is based on probabilistic 3D category models - 3D Gaussian Contour Category Models (see Chapter 5)- which use a 3D shape representation based on 3D contour fragments (1D manifolds in

3D), so called ‘3D contour clouds’ (see Chapter 4). Our pose estimation algorithm consists of several steps. First, we build probabilistic 2D Gaussian Aspect Models from 3D Gaussian Contour Category Models using an Unscented Transformation, which approximates a probability density function that undergoes a nonlinear transformation using just a small number of weighted points. For this, we can estimate the mean and the variance of our Gaussian Mixture Models (GMMs) in our projected 2D Gaussian Aspect Models. Second, we build a 2D Gaussian Contour Model for each 2D input image by representing each 2D contour fragment by a Gaussian Mixture Model. Third, we establish a voting procedure on the basis of the probabilistic 2D Gaussian Aspect Models to compute a pose hypothesis for an object in a given 2D input image. We combine local shape information and global geometric information in a similarity measure, which we introduce in a hypothesis testing framework.

Our experimental evaluation uses the available ground truth of the horses, cows, and dogs of the well-known ETH-80 dataset (see Section 3.1.2 for an overview) and cars of the 3D object category dataset (see Section 3.1.1 for an overview).

6.2 State-of-the-Art

Most of the current pose estimation algorithms combine object categorization and pose estimation by doing both simultaneously [LS10, SGS10, SFF08, SSSFF09, ÖLF09] or by building a model for a special category for pose estimation [ANB09]. Section 2.2 discusses most of the current research in 3D object categorization and pose estimation. To summarize these methods: In [LS10] Liebelt and Schmid present a multi-view object class detection system where they combine a 2D part-based categorization approach with a 3D geometry-based model built from synthetic data (CAD models) for pose estimation. In [SGS10] Stark et al. also present multi-view object categorization built on 3D CAD Models. In [SFF08] Savarese and Fei Fei develop an approach where they build a 3D part based model from images of different poses for categorization and pose estimation. They also present a synthesis of object categories. In [ANB09] Arie-Nachmison and Basri build a 3D Implicit Shape Model for pose estimation of cars.

Most of the mentioned methods are appearance based. In contrast to them, our method is based on shape information - especially contour information (silhouette and inner contours). Similar to Arie-Nachmison and Basri [ANB09] we build a 3D category model and use this model for pose estimation in 2D images. In contrast to them, our model is generated fully automatically. Whereas most of the current systems split the dataset in training data and test data, our 3D model is completely separated from the dataset, i.e. we do not learn on the same dataset as we use for the evaluation of our pose estimation algorithm.

6.3 2D Aspect Model: From 3D Gaussian Contour Category Model to 2D Gaussian Aspect Models

Our pose estimation algorithm is based on the principle that we do not learn each 2D aspect of a category separately but we learn one pose-invariant 3D model per category and compute 2D aspect models of it afterwards. This has several advantages e.g. we know the relations between views, we have just to learn one model and we are independent of the number of 2D aspect models. We first learn a 3D Gaussian Contour Category Model and compute afterwards 2D aspect models using an Unscented Transformation for the perspective projection - we call them 2D Gaussian Aspect Models. In our practical implementation we choose the camera pose on the basis of azimuth and elevation angles. Thereby, we are able to compute 2D Gaussian Aspect Models from arbitrary poses.

6.3.1 Unscented Transformation

For 3D to 2D projection of a 3D Gaussian Contour Category Model to a 2D Gaussian Aspect Model we use an Unscented Transformation (see [JU96]). An

Unscented Transformation computes the mean and the variance of a random variable, which has been transformed by a transformation f. Let X be an L-dimensional random variable with mean μ_k, and variance Σ_k, so that $y = f(X)$. In our case the transformation f is a perspective projection. For the Unscented Transformation, we first compute $2L+1$ weighted points, so called sigma points, using the following equations:

$$\begin{array}{lll} \mathcal{X}_0 = \mu_X & \tau_0 = \frac{\kappa}{L+\kappa} & i = 0 \\ \mathcal{X}_i = \mu_X + (\sqrt{(L+\kappa)\Sigma_X})_i & \tau_i = \frac{1}{2(L+\kappa)} & i = 1, ..., L \\ \mathcal{X}_i = \mu_X - (\sqrt{(L+\kappa)\Sigma_X})_{i-L} & \tau_i = \frac{1}{2(L+\kappa)} & i = L+1, ..., 2L \end{array} \tag{6.1}$$

where τ_i is the weight of the i^{th} sigma point with $\sum_{i=0}^{2L} \tau_i = 1$ and κ is a scaling factor[1]. The notation $(\sqrt{(L+\kappa)\Sigma_X})_i$ means that we choose the i^{th} row of the matrix for the computation of the sigma point. Now, we compute the mean and the variance of our projected sigma points $\mathcal{Y}_i = f(\mathcal{X}_i)$ by

$$\mu_y = \sum_{i=0}^{2L} \tau_i \mathcal{Y}_i \qquad \text{and} \qquad \Sigma_y = \sum_{i=0}^{2L} \tau_i (\mathcal{Y}_i - \mu_y)(\mathcal{Y}_i - \mu_y)^T. \tag{6.2}$$

6.3.2 Practical Implementation

In our practical implementation we define a viewpoint for the 2D Gaussian Aspect Model by choosing an azimuth and an elevation angle with respect to the object centered coordinate system. Figure 6.2 shows a sample illustration for the upper hemisphere around an object and several camera poses. By the azimuth and elevation angles we are able to compute the camera pose around the object for the perspective projection. For our experiments we use a defined set of poses that correspond to poses available in the test dataset. In an additional experiment, we investigate the effect when poses that are available in the test dataset do not have a direct correspondence in the 2D Gaussian Aspect Models.

[1] In our experiments $L = 3$ and we choose $\kappa = 1$.

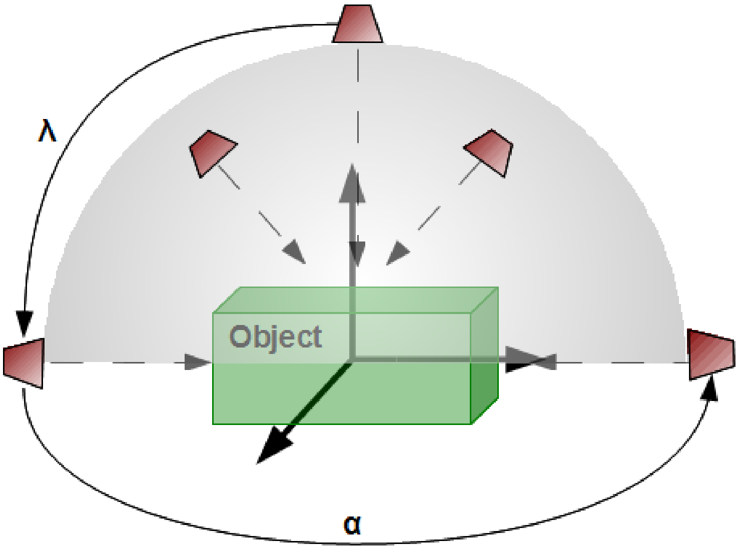


Figure 6.2: 3D to 2D projection of a 3D Gaussian Contour Category Model from several viewpoints that are chosen on a hemisphere around the object.

6.4 Representation Model: From 2D Input Images to 2D Gaussian Contour Models

The representation of 2D contour fragments by Gaussian Mixture Models (GMMs) is very flexible and robust. Noise, outliers, and deformations can be handled, the 2D geometry of the contour shapes can be maintained. The advantages of representation by Gaussian Mixture Models have already been discussed in Section 5.3 for 3D contour fragments. The same apply to 2D contour fragments.

As mentioned in Section 6.3, we compute 2D Gaussian Aspect Models for several poses from a 3D Gaussian Contour Category Model. Such a 2D Gaussian Aspect Model consists of partitions of 2D Gaussian Mixture Models. For our voting procedure we need the same representation for the 2D input image. Therefore, we compute 2D contour fragments of our input image by applying the Canny edge detector and a linking algorithm based on smoothing constraints [LHS07]. Afterwards, we compute for each 2D input image a 2D Gaussian Contour Model by representing each 2D contour fragment by a GMM. Given a 2D input image I consisting of a set of 2D contour fragments $\mathcal{T}_l$

$$I = \{\mathcal{T}_l; l = 1 \dots N\}, \tag{6.3}$$

where N is the number of 2D contour fragments in the image and $\mathcal{T}_l = p_1, ..., p_n$ is a set of 2D edges, we can fit a Gaussian Mixture Model to each 2D contour fragment $\mathcal{T}_l$ using the standard Expectation-Maximization (EM) algorithm (see e.g. [XJ96]). Each 2D contour fragment $\mathcal{T}_l$ is given by

$$\Theta_K^i(x) = \sum_{k=1}^{K} \alpha_k p(x|N(\mu_k, \Sigma_k)) \tag{6.4}$$

with probability

$$p(x|N(\mu_k, \Sigma_k)) = \frac{1}{(2\pi)^{\frac{n}{2}} \|\Sigma_k\|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\right) \tag{6.5}$$

and Θ_K^i is the Gaussian Mixture Model for a fragment $\mathcal{T}_l$ and K is the number of mixture components with mean μ_k, variance Σ_k, and weight α_k. In our representation, K is chosen according to the length of the 2D contour fragment (with a maximum of $K = 10$). Similar to the representation of 3D contour fragment (see Section 5.3), the covariance matrix gives essential information about the orientation of a 3D contour fragment, so we do not simplify it to the identity matrix. We assume that each mixture component has the same weight, so that $\alpha_k = 1$. That means our algorithm does not decide on the basis of the weight, whether a mixture component is relevant for the pose hypothesis or not, because even a mixture component with a small weight may provide essential information about the pose.

6.5 Pose Estimation Algorithm

We establish a voting procedure to compute a pose hypothesis of an object in a given 2D input image. In the voting procedure we compare the 2D Gaussian Contour Model of a 2D input image with 2D Gaussian Aspect Models of several poses. Our pose estimation algorithm requires a similarity measure between mixture components of Gaussian Mixture Models in order to identify the pose of an object in a given 2D input image. The novel similarity measure can be derived as a modification of the well-known Kullback-Leibler (KL) divergence, which is used in many existing shape matching methods as the divergence between two GMMs, but is not direct applicable to our pose estimation algorithm. In contrast to the Kullback-Leibler divergence, we combine local shape information and 2D geometric information in one voting procedure. We present our pose estimation algorithm in a hypothesis framework.

Similar to the test statistic we use for learning a 3D Gaussian Contour Category Model (see Section 5.4.1), we start with the approximation of the KL-divergence between two GMMs [GGG03] by:

$$KL(f\|g) \approx \sum_{i=1}^{n} \alpha_i \min_j (KL(f_i\|g_j) + \log \frac{\alpha_i}{\beta_j}), \tag{6.6}$$

where f and g are two GMMs:

$$f = \sum_{i=1}^{n} f_i = \sum_{i=1}^{n} \alpha_i p(x|N(\mu_i, \Sigma_i)) \tag{6.7}$$

and

$$g = \sum_{j=1}^{m} g_j = \sum_{j=1}^{m} \beta_j p(x|N(\mu_j, \Sigma_j)), \tag{6.8}$$

The KL-divergence between two GMMs uses the closed-form Kullback-Leibler divergence between two Gaussians $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$. It is given by

$$KL = \frac{1}{2}\left(\log \frac{|\Sigma_2|}{|\Sigma_1|} + tr(\Sigma_2^{-1}\Sigma_1) + (\mu_1 - \mu_2)^T \Sigma_2^{-1}(\mu_1 - \mu_2)\right) \tag{6.9}$$

where $|.|$ denotes the determinant.

In the remaining section we use the following notation: $\mathcal{I}$ denotes the 2D Gaussian Contour Model of a given 2D input image containing an object of a given category. $\mathcal{P} = P_1, ..., P_C$ denotes the 2D Gaussian Aspect Models of C poses. $(\mu_j^{P_p}, \Sigma_j^{P_p})$ and $(\mu_i^{\mathcal{I}}, \Sigma_i^{\mathcal{I}})$ are mixture components of 2D Gaussian Aspect Model P_p and a 2D Gaussian Contour Model $\mathcal{I}$. Further, we define $oc^{\mathcal{I}}$ to be the object center of the detected object in the input image $\mathcal{I}$ and we define oc^{P_p} to be the object center of the object in the 2D Gaussian Aspect Model P_p. Let $v_j^{P_p}$

be the relative difference between a mixture component $\mu_j^{P_p}$ and the object center oc^{P_p} of the 2D Gaussian Aspect Model, and $v_i^{\mathcal{I}}$ be the position difference between a mixture component $\mu_i^{\mathcal{I}}$ and the object center $oc^{\mathcal{I}}$ of a 2D Gaussian Contour Model. (R_i, T_i) denotes the local transformation i.e. rotation and translation of a single mixture component (μ_i, Σ_i).

For our pose estimation algorithm we introduce

1. $SM(f_i, g_j)$: Similarity measure between mixture components, where the orientation of a mixture component (given by the principal eigenvector of its variance), and hence the orientation of the contour fragment as well as its position on the object (given by the mean) have the same importance.

2. $SM(f\|g)$: Similarity measure $SM(f\|g)$ between two GMMs. It is defined on the basis of the KL-divergence approximation (6.6), but by using $SM(f_i, g_j)$ we introduce a discretization.

3. $H(\mathcal{P})$: Pose hypothesis $H(\mathcal{P})$. Here, we use $SM(f\|g)$ as a similarity measure between a 2D Gaussian Contour Model $\mathcal{I}$ of a given input image and a 2D Gaussian Aspect Model P_p. In contrast to the 3D Gaussian Contour Category Model we do not use the partitioning but the GMMs as a whole.

Similar to Section 5.4.1 we introduce a similarity measure $SM(f_i, g_j)$ between two Gaussians which takes into account the orientation of the eigenvector and the position of the mixture component on the object. Due to a first alignment of the bounding boxes of a 2D Gaussian Aspect Model and the detected object in the input image, we can neglect the global transformation between the GMMs. We define the following hypothesis test for the similarity between a mixture component of a 2D Gaussian Contour Model and a mixture component of a given 2D Gaussian Aspect Model

$$\begin{aligned} H_0 : &\quad (\mu_i^{\mathcal{I}}, \Sigma_i^{\mathcal{I}}) = (\mu_j^{P_p} + T_j, R_j \cdot \Sigma_j^{P_p}) \\ H_1 : &\quad \text{otherwise} \end{aligned} \tag{6.10}$$

$$\text{reject} \qquad H_0 \;\; \text{if } TS_1 > \gamma_1 \text{ or } TS_2 < \gamma_2$$

where γ_1 and γ_2 are the thresholds and TS_1 and TS_2 are test statistics. TS_1 is the test statistic which defines the position on the object. Let

$$v_i^{\mathcal{I}} = \mu_i^{\mathcal{I}} - oc^{\mathcal{I}} \qquad \text{and} \qquad v_j^{P_p} = \mu_j^{P_p} - oc^{P_p} \tag{6.11}$$

be the vectors between mixture components and object center. Then TS_1 is given by

$$TS_1 = \|v_i^{\mathcal{I}} - v_j^{P_p}\|_2. \tag{6.12}$$

Let e_i^i be the principal eigenvector of Σ_i^I and $e_j^{P_p}$ the principal eigenvector of $\Sigma_j^{P_p}$, then TS_2 is given by the scalar product between these eigenvectors

$$TS_2 = e_i^{\mathcal{I}} \cdot e_j^{P_p}. \tag{6.13}$$

The threshold γ_1 defines the allowed displacement, and γ_2 defines the allowed orientation difference[2]. On the one hand, too low thresholds would not be able to handle shape deformations of different objects of a category. On the other hand, too high thresholds would not be able to handle different poses of an object. Our discrete similarity measure between two Gaussians is then given by

$$SM(f_i, g_j) = \begin{cases} 0 & \text{if } TS_1 < \gamma_1 \text{ and } TS_2 > \gamma_2 \\ 1 & \text{otherwise.} \end{cases} \tag{6.14}$$

To identify which pose hypothesis P_p is the correct one we use a modification of the KL-divergence (6.6). Assuming that all mixture components have the same weight $\alpha_i = 1$ and $\beta_j = 1$ and using our own similarity measure $SM(f_i, g_j)$ instead of the KL-divergence $KL(f_i, g_j)$ between two mixture components we can rewrite (6.6) to

$$S\hat{M}(f\|g) \approx \sum_{i=1}^{n} \min_j (SM(f_i\|g_j)) \tag{6.15}$$

Using this formulation, $S\hat{M}(f\|g)$ defines the number of mixture components in f which have no correct match in g. Similarly,

$$SM(f\|g) \approx \sum_{i=1}^{n} \max_j (1 - SM(f_i\|g_j)) \tag{6.16}$$

[2] In our experiments, typically we choose $\gamma_2 = 0.98$ and we choose γ_1 as 5%-10% of the bounding box size.

defines the number of mixture components in f which have a correct match in g. For all 2D Gaussian Aspect Models in $\mathcal{P}$ the pose hypothesis $H(\mathcal{P})$ is given by

$$H(\mathcal{P}) = \arg\max_{p \in \mathcal{P}} \frac{SM(\mathcal{I} \| P_p)}{\hat{SM}(\mathcal{I} \| P_p)}. \tag{6.17}$$

6.6 Experimental Evaluation

For our experimental evaluation we compute 3D Gaussian Contour Category Models for the categories ‘horse’, ‘cow’, ‘dog’, and ‘car’ on our GRAZ-STEREO-BASE-xx dataset (i.e. it is not trained on ETH-80 or 3D object category dataset). Based on these 3D Gaussian Contour Category Models, we test our pose estimation algorithm on 17 poses of the ETH-80 dataset [LS03] for the categories ‘horse’, ‘cow’, and ‘dog’ and 8 poses for two heights of the 3D object category dataset for the category ‘car’ [SFF08]. For these experiments we use a defined set of poses (17 for ETH-80 and 2×8 for 3D object category dataset) for the 2D Gaussian Aspect Models that correspond to the selected poses of the dataset. To demonstrate that our approach also works for poses that do not have a precise match in the 2D Gaussian Aspect Models, we additionally test 8 intermediate poses of the ETH-80 dataset, which lie between the 17 poses of the 2D Gaussian Aspect Models. In our experiments we assume that the category and location of the object in a given 2D input image is known, leaving us with the task of pose estimation. The complexity of the pose estimation algorithm is linear in the number of poses. The reconstruction of 3D Gaussian Contour Category Models is computationally demanding, but needs to be done only once per category.

6.6.1 Visualization of 2D Gaussian Aspect Models and 2D Gaussian Contour Models

The basis of our pose estimation algorithm are 2D Gaussian Aspect Models derived from a 3D Gaussian Contour Category Model on the one hand and 2D Gaussian Contour Models computed for the 2D input image on the other hand. Figure 6.3 shows an example of such a 3D model and two example 2D Gaussian

Aspect Models for the poses P5 and P13 (cf. Figure 6.5), which were computed using the Unscented Transformation (see Section 6.3.1). In our experiments, we choose several viewpoints over the top hemisphere around the object according to the tested dataset.

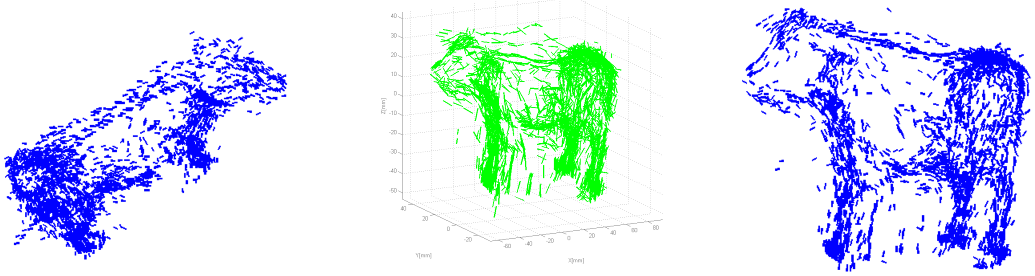


Figure 6.3: Example 3D Gaussian Contour Category Model (mid) and two 2D Gaussian Aspect Models (left and right) of poses P5 and P13 (cf. Figure 6.5). The probability densities are drawn by 2D lines given by their mean and the principal eigenvector of their variance. For purpose of visualization only discriminative mixture components of one training object are shown.

Figure 6.4 shows two example input images and their 2D Gaussian Contour Models for the same poses as in Figure 6.3.

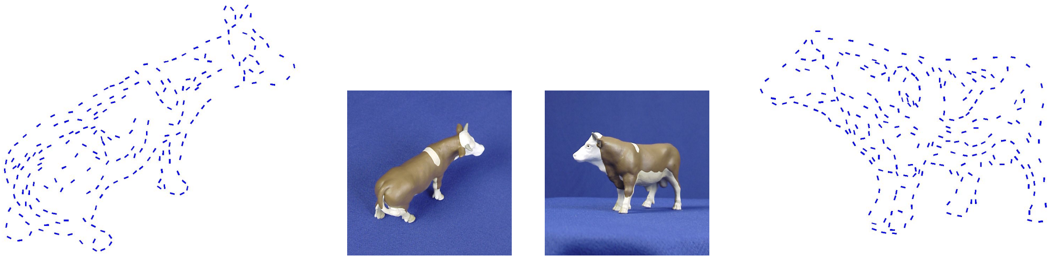


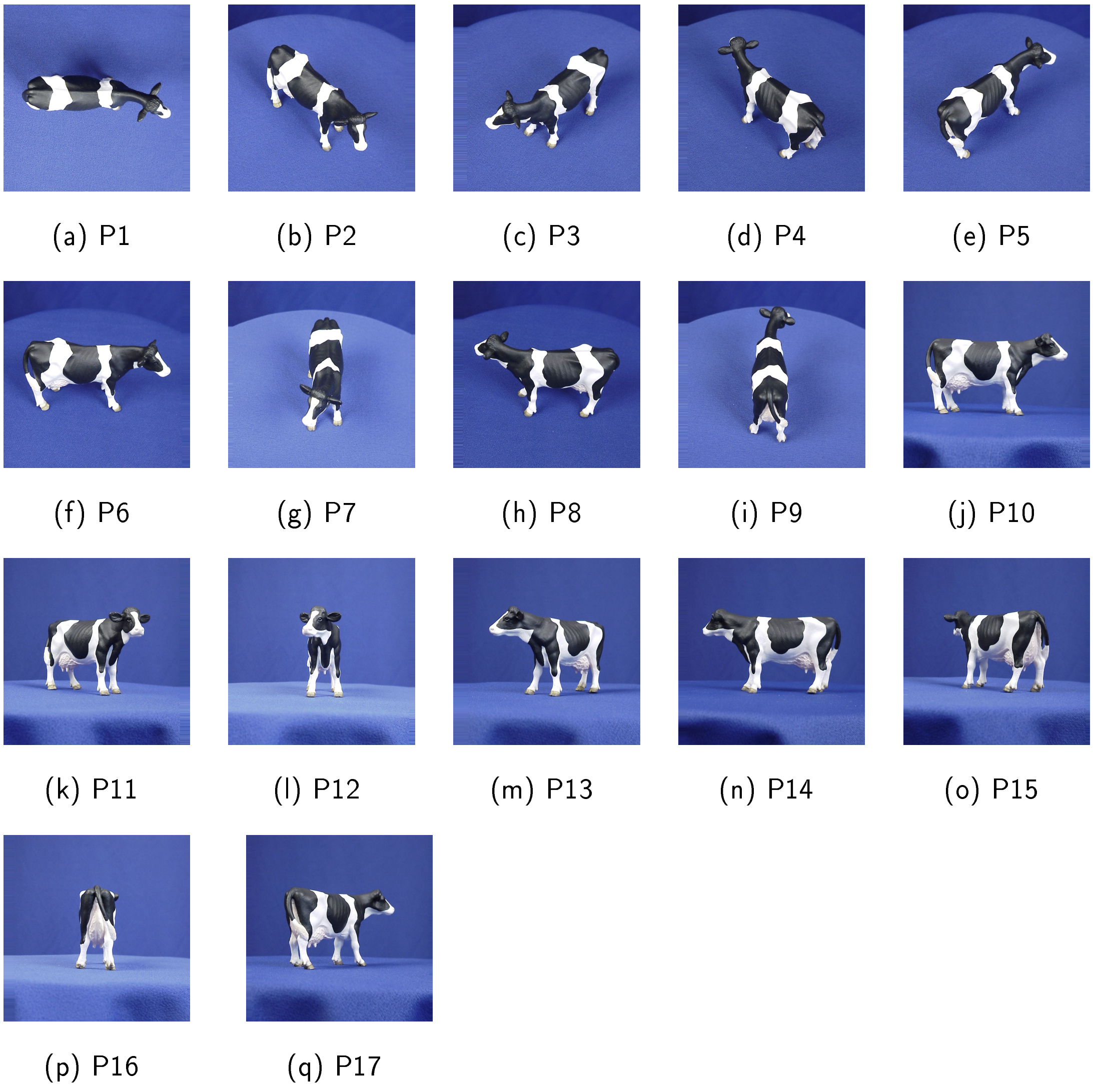
Figure 6.4: Example 2D input images and Gaussian Mixture Models for two example poses P5 and P13 (cf. Figure 6.5). The probability densities are depicted as 2D lines given by the mean and the principal eigenvector of the covariance matrix of the mixture component.

6.6.2 Evaluation on the ETH-80 Dataset

In our experiments on the ETH-80 dataset we estimate the pose of the categories cows, horses, and dogs. We select 17 poses of all ten horses, seven cows (except the lying cow and the cows with head-down), and all ten dogs of the ETH-80 dataset (see Figure 6.5 for a cow example of the selected 17 poses). We evaluate our 3D pose estimation algorithm by computing the similarity measure between the 2D Gaussian Contour Models and 2D Gaussian Aspect Models for the selected categories and compare these pose hypotheses with the ground truth given by the ETH-80 dataset.

Figure 6.6 shows the confusion matrix of the pose estimation experiments on the ETH-80 cows. We achieve an average accuracy of 68% of the pose estimation. We see that problems occur with neighboring views (e.g. P14 votes not only for P14, but also for P8; P10 votes not only for P10, but also for P6 and P11) and pose ambiguity (e.g. P17 votes for P17 and P11). On the one hand, Figure 6.5 shows that neighboring views may look similar and so shape features have a similar appearance of position and orientation on the object. On the other hand, we see that pose ambiguity plays an important role. When working with 2D images we lose one important information: depth. In several poses e.g. front or back view of shape models (see Figure 6.1(d) and Figure 6.1(e) for an example) it is hard to differ solely on the basis of their shape. Figure 6.8(a) demonstrates the shape similarity on the basis of superimposing 2D Gaussian Mixture Models for P11 and P17. Apart from these two points, many poses can be estimated correctly, six poses achieve 100%.

Figure 6.7 shows the confusion matrix of the pose estimation experiments on the ETH-80 horses. We achieve an average accuracy of 64% of the pose estimation. We see a similar behavior as in the cow experiment for similar views (neighboring views, pose ambiguity). Figure 6.8(b) demonstrates the shape similarity on the basis of superimposing 2D Gaussian Mixture Models for P12 and P16. Here, it is even for humans hard to distinguish between these two poses. Figure 6.9 shows example pose estimation results on the ETH-80 dataset.



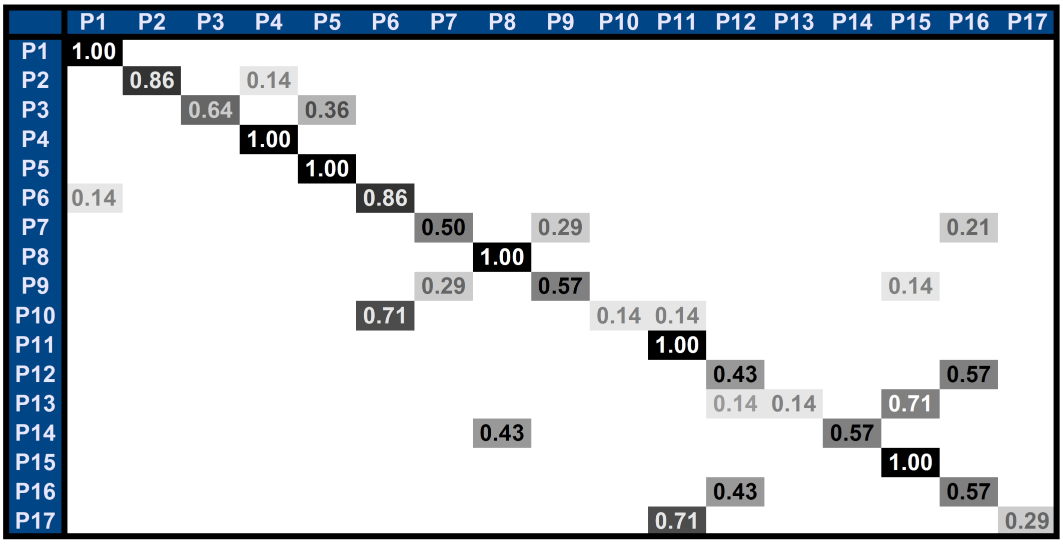
(a) P1 (b) P2 (c) P3 (d) P4 (e) P5

(f) P6 (g) P7 (h) P8 (i) P9 (j) P10

(k) P11 (l) P12 (m) P13 (n) P14 (o) P15

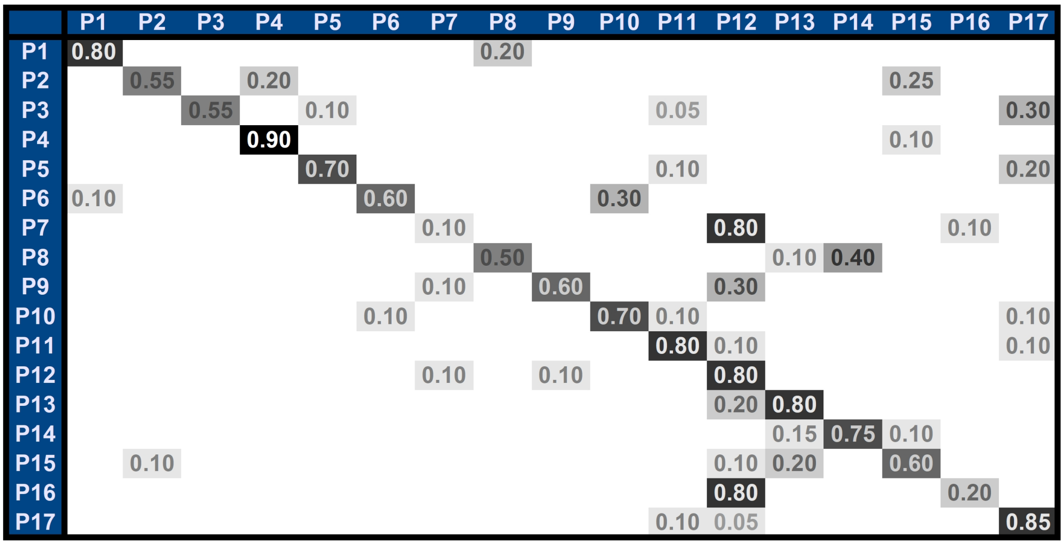
(p) P16 (q) P17

Figure 6.5: 17 poses of a cow chosen from the ETH-80 (azimuth and elevation denoted on the basis of their file names). (a) P1: azimuth $\alpha = 0^o$ and elevation $\lambda = 0^o$. (b) P2: $\alpha = 35^o$ and $\lambda = 45^o$. (c) P3: $\alpha = 35^o$ and $\lambda = 135^o$. (d) P4: $\alpha = 35^o$ and $\lambda = 225^o$. (e) P5: $\alpha = 35^o$ and $\lambda = 315^o$. (f) P6: $\alpha = 45^o$ and $\lambda = 0^o$. (g) P7: $\alpha = 45^o$ and $\lambda = 90^o$. (h) P8: $\alpha = 45^o$ and $\lambda = 180^o$. (i) P9: $\alpha = 45^o$ and $\lambda = 270^o$. (j) P10: $\alpha = 90^o$ and $\lambda = 0^o$. (k) P11: $\alpha = 90^o$ and $\lambda = 45^o$. (l) P12: $\alpha = 90^o$ and $\lambda = 90^o$. (m) P13: $\alpha = 90^o$ and $\lambda = 135^o$. (n) P14: $\alpha = 90^o$ and $\lambda = 180^o$. (o) P15: $\alpha = 90^o$ and $\lambda = 225^o$. (p) P16: $\alpha = 90^o$ and $\lambda = 270^o$. (q) P17: $\alpha = 90^o$ and $\lambda = 315^o$.



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17
P1	1.00																
P2		0.86		0.14													
P3			0.64		0.36												
P4				1.00													
P5					1.00												
P6	0.14					0.86											
P7							0.50		0.29							0.21	
P8								1.00									
P9							0.29		0.57						0.14		
P10						0.71				0.14	0.14						
P11											1.00						
P12												0.43				0.57	
P13												0.14	0.14		0.71		
P14								0.43						0.57			
P15															1.00		
P16												0.43				0.57	
P17											0.71						0.29

Figure 6.6: Confusion matrix for pose estimation for seven cows of the ETH-80 cows (average accuracy 68%).



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17
P1	0.80							0.20									
P2		0.55		0.20											0.25		
P3			0.55		0.10						0.05						0.30
P4				0.90											0.10		
P5					0.70						0.10						0.20
P6	0.10					0.60				0.30							
P7							0.10					0.80				0.10	
P8								0.50					0.10	0.40			
P9							0.10		0.60			0.30					
P10						0.10				0.70	0.10						0.10
P11											0.80	0.10					0.10
P12							0.10		0.10			0.80					
P13												0.20	0.80				
P14													0.15	0.75	0.10		
P15		0.10										0.10	0.20		0.60		
P16												0.80				0.20	
P17											0.10	0.05					0.85

Figure 6.7: Confusion matrix for pose estimation on the ETH-80 horse dataset for all ten horses (average accuracy 64%).

Figure 6.10 shows the confusion matrix of the pose estimation experiments on the ETH-80 dogs. We achieve an average accuracy of 59% for the pose estimation. Similar to the horse and cow results there are problems with neighboring views

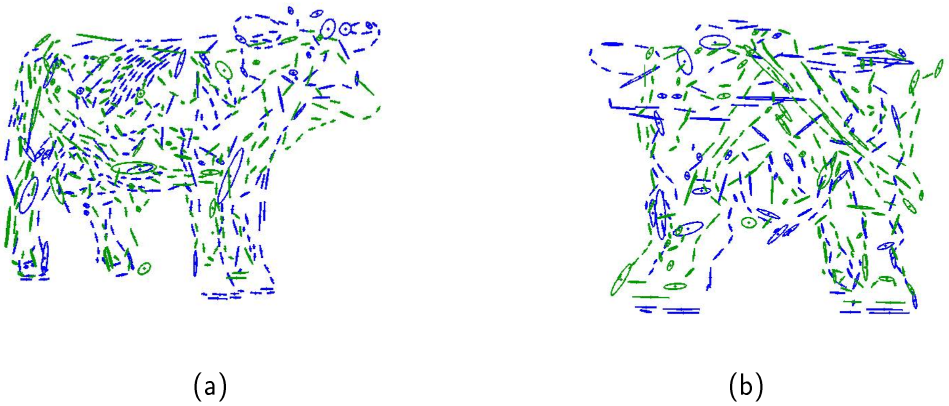


Figure 6.8: Illustration of 2D Gaussian Mixture Model similarity. (a) Two poses P11 (blue) and P17 (green) of two example ETH-80 cows (right). (b) Two poses P12 (blue) and P16 (green) of two example ETH-80 horses (left).

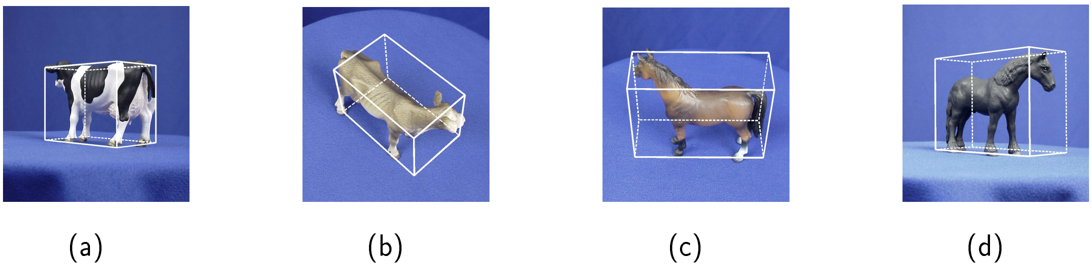
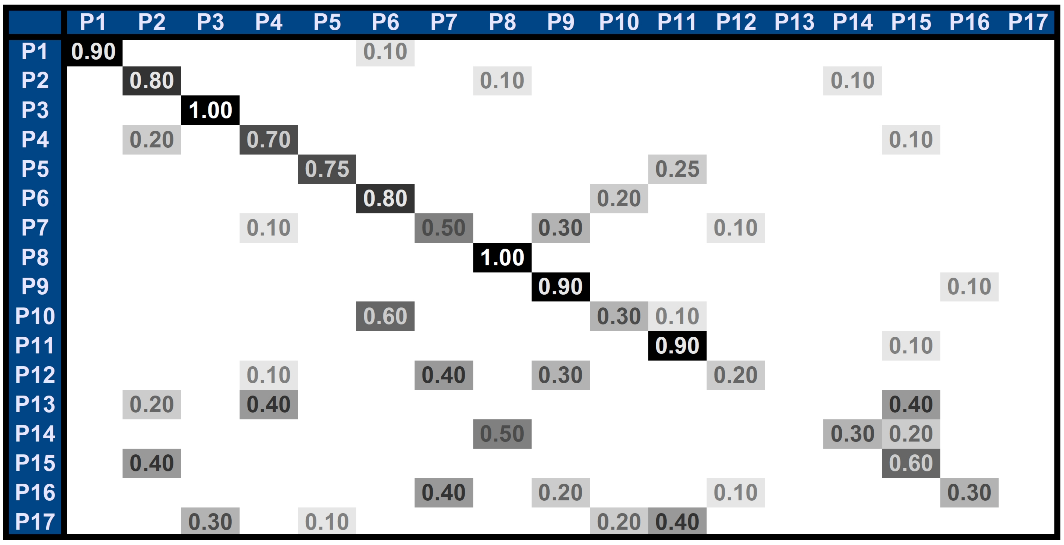


Figure 6.9: Example pose estimation results. (a) P15 of an ETH-80 cow. (b) P2 of an ETH-80 cow. (c) P8 of an ETH-80 horse. (d) P11 of an ETH-80 horse.

and pose ambiguity. Apart from these two points, many poses can be estimated correctly, two poses achieve 100%, three poses achieve 90%.

Evaluation on non-matching poses of the ETH-80 dataset

What happen if the tested view do not directly correspond to a projected 2D Gaussian Aspect Model? To test the robustness of our approach to that situation we select 8 additional poses of seven cows of the ETH-80 dataset which lie between P11 and P17 (see Figure 6.12(a)). Figure 6.11 shows the selected 8 poses for one cow of the ETH-80 dataset. We test these 8 poses on the 2D Gaussian Aspect



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17
P1	0.90					0.10											
P2		0.80						0.10						0.10			
P3			1.00														
P4		0.20		0.70											0.10		
P5					0.75						0.25						
P6						0.80				0.20							
P7				0.10			0.50		0.30			0.10					
P8								1.00									
P9									0.90							0.10	
P10						0.60				0.30	0.10						
P11											0.90				0.10		
P12				0.10			0.40		0.30			0.20					
P13		0.20		0.40											0.40		
P14								0.50						0.30	0.20		
P15		0.40													0.60		
P16							0.40		0.20			0.10				0.30	
P17			0.30		0.10					0.20	0.40						

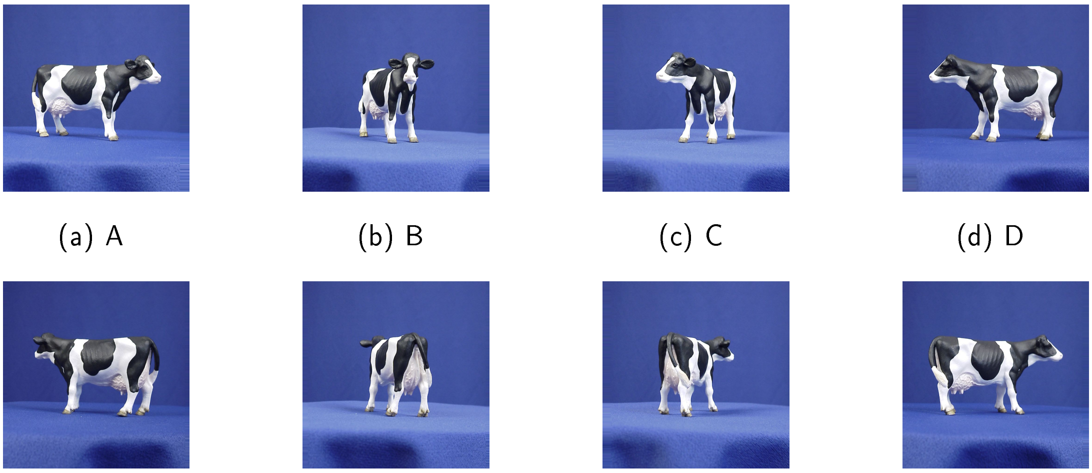
Figure 6.10: Confusion matrix for pose estimation on the ETH-80 dog dataset for all ten dogs (average accuracy 59%).

Models for P11 to P17 of the Experiment in the last section. We would expect that the view vote for those poses which are most similar to its own pose.

Figure 6.12(b) shows the confusion matrix for this experiment. As desired, the selected poses vote for those poses which are most similar. As before (see Section 6.6.2), we observe difficulties due to pose ambiguity for poses C and G. However, we demonstrate that also poses which are not in the 2D Gaussian Aspect Models obtain good pose estimation results.

6.6.3 Evaluation on the 3D Object Category Dataset

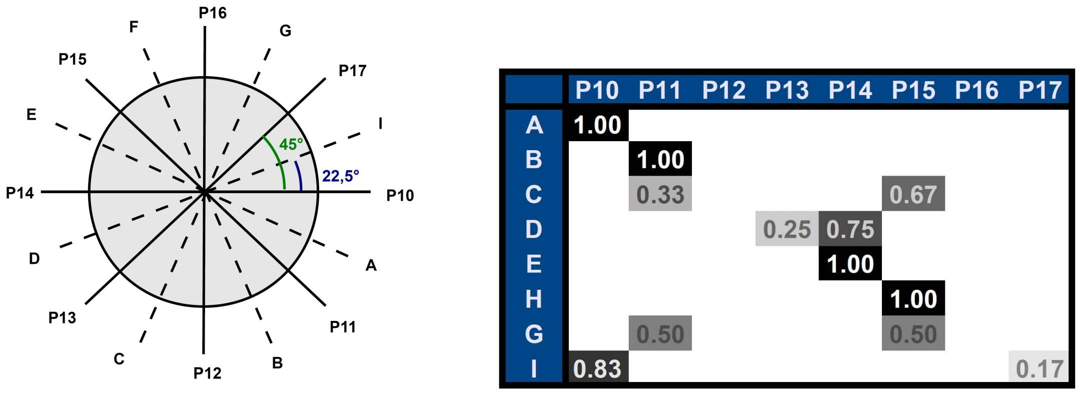
In our experiments on the 3D object category dataset we estimate the pose of the category car. We select 8 poses of ten objects, two heights (H1 and H2) and two scales (S1 and S2) (see Figure 6.13 for the illustration of H1, H2, S1, S2). For our experiments we split them into two sets depending on their azimuth angle. Figure 6.14 gives an overview of these poses for one car (out of ten) and one scale. We evaluate our 3D pose estimation algorithm by computing the similarity measure between the 2D Gaussian Contour Models and 2D Gaussian



(a) A (b) B (c) C (d) D

(e) E (f) F (g) G (h) I

Figure 6.11: 8 poses of a cow chosen from the ETH-80 which lies between that in Figure 6.5 (azimuth and elevation denoted on the basis of their file names). (a) A: azimuth $\alpha = 90^o$ and elevation $\lambda = 22^o$. (b) B: $\alpha = 90^o$ and $\lambda = 68^o$. (c) C: $\alpha = 90^o$ and $\lambda = 112^o$. (d) D: $\alpha = 90^o$ and $\lambda = 158^o$. (e) E: $\alpha = 90^o$ and $\lambda = 202^o$. (f) F: $\alpha = 90^o$ and $\lambda = 248^o$. (g) G: $\alpha = 90^o$ and $\lambda = 292^o$. (h) I: $\alpha = 90^o$ and $\lambda = 338^o$.

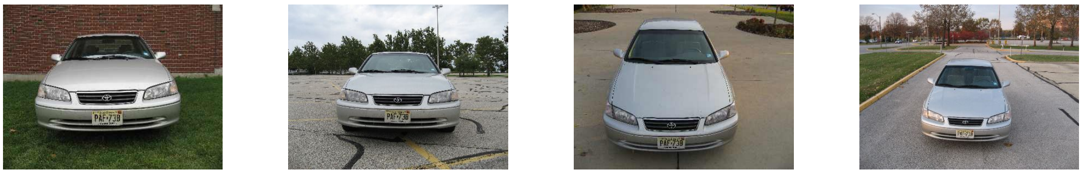


	P10	P11	P12	P13	P14	P15	P16	P17
A	1.00							
B		1.00						
C		0.33				0.67		
D				0.25	0.75			
E					1.00			
H						1.00		
G		0.50				0.50		
I	0.83							0.17

(a) (b)

Figure 6.12: (a) Illustration of the pose distribution for the 8 additional poses which lies between P10 and P17 a distance of 22.5^o. (b) Confusion matrix for pose estimation on 8 additional poses of seven ETH-80 cows .

Aspect Models for the selected category and compare these pose hypotheses with the ground truth given by the dataset.



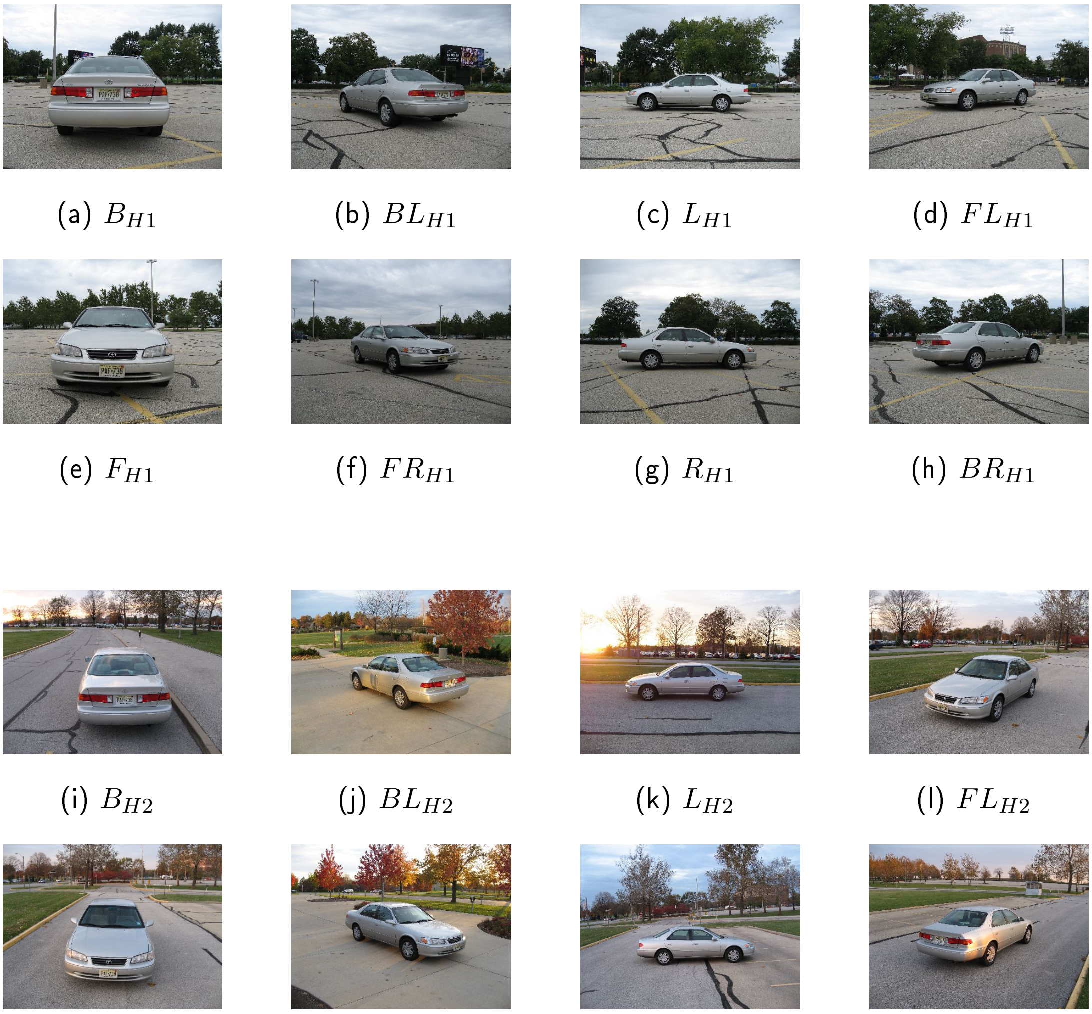
(a) $H1 - S1$ (b) $H1 - S2$ (c) $H2 - S1$ (d) $H2 - S2$

Figure 6.13: Illustration of the two heights H1 and H2 and the two scales S1 and S2 on the basis of one example car from the 3D object category dataset.

Figure 6.15(a) shows the confusion matrix of the pose estimation experiments on the 3D object category dataset car (H1) for the poses B_{H1} to BR_{H1}. We achieve an average accuracy of the pose estimation of 51%. On the one hand, there occur problems due to pose ambiguity (FL_{H1} - BL_{H1}). On the other hand, false pose hypotheses are computed because of shape symmetry of cars, e.g. F_{H1} votes for B_{H1} and vice versa or BL_{H1} votes for FR_{H1}.

Figure 6.15(b) shows the confusion matrix of the pose estimation experiments on the 3D object category dataset car (H2) for the poses B_{H2} to BR_{H2}. We achieve an average accuracy of the pose estimation of 58%. Except for the poses F_{H2} and FR_{H2}, the entries of the diagonal are above 60%. For those with less than 60% problems with symmetric views can occur (i.e. F_{H2} votes for B_{H2}; FR_{H2} votes for BL_{H2}). For these views we obtain a similar silhouette and similar shape information due to symmetry (e.g. wheels front and back windows, lights).

Due to the combination of object categorization and pose estimation, a direct comparison to other approaches is hard. Arie-Nachmison and Basri [ANB09] test their 3D Implicit Shape Model for 'car' on 160 images (5 objects, 16 viewpoints and two scalings) of the 3D object category dataset. The other 5 objects of the dataset were used for learning. Their pose estimation evaluation is only done on those images, where a car is correctly detected and only for eight viewing directions. They achieve an average accuracy rate of 48.5% with similar problems due to pose ambiguity, neighboring views, and symmetry as in our approach, which shows that these problems are not only related to the use of shape information.



(a) B_{H1} (b) BL_{H1} (c) L_{H1} (d) FL_{H1}

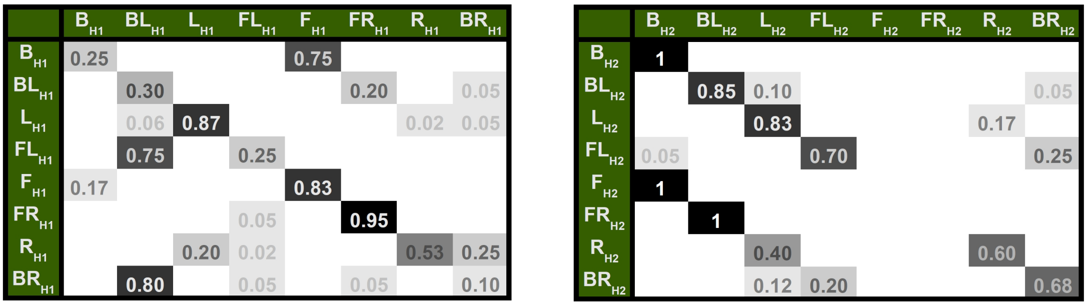
(e) F_{H1} (f) FR_{H1} (g) R_{H1} (h) BR_{H1}

(i) B_{H2} (j) BL_{H2} (k) L_{H2} (l) FL_{H2}

(m) F_{H2} (n) FR_{H2} (o) R_{H2} (p) BR_{H2}

Figure 6.14: 16 poses of a car from the 3D object category dataset (azimuth α and elevation λ denoted in the same way as for the ETH-80 dataset). (a) B_{H1}: $\alpha = 90^o$ and $\lambda = 270^o$. (b) BL_{H1}: $\alpha = 90^o$ and $\lambda = 225^o$. (c) L_{H1}: $\alpha = 90^o$ and $\lambda = 180^o$. (d) FL_{H1}: $\alpha = 90^o$ and $\lambda = 135^o$. (e) F_{H1}: $\alpha = 90^o$ and $\lambda = 90^o$. (f) FR_{H1}: $\alpha = 90^o$ and $\lambda = 45^o$. (g) R_{H1}: $\alpha = 90^o$ and $\lambda = 0^o$. (h) BR_{H1}: $\alpha = 90^o$ and $\lambda = 315^o$. (i) B_{H2}: $\alpha = 45^o$ and $\lambda = 270^o$. (j) BL_{H2}: $\alpha = 45^o$ and $\lambda = 225^o$. (k) L_{H2}: $\alpha = 45^o$ and $\lambda = 180^o$. (l) FL_{H2}: $\alpha = 45^o$ and $\lambda = 135^o$. (m) F_{H2}: $\alpha = 45^o$ and $\lambda = 90^o$. (n) FR_{H2}: $\alpha = 45^o$ and $\lambda = 45^o$. (o) R_{H2}: $\alpha = 45^o$ and $\lambda = 0^o$. (p) BR_{H2}: $\alpha = 45^o$ and $\lambda = 315^o$.

Liebelt and Schmid [LS10] test their algorithm also on the category ‘car’ of the 3D object category dataset. In contrast to [ANB09] they chose randomly 7



	B_{H1}	BL_{H1}	L_{H1}	FL_{H1}	F_{H1}	FR_{H1}	R_{H1}	BR_{H1}
B_{H1}	0.25				0.75			
BL_{H1}		0.30				0.20		0.05
L_{H1}		0.06	0.87				0.02	0.05
FL_{H1}		0.75		0.25				
F_{H1}	0.17				0.83			
FR_{H1}				0.05		0.95		
R_{H1}			0.20	0.02			0.53	0.25
BR_{H1}		0.80		0.05		0.05		0.10

(a)

	B_{H2}	BL_{H2}	L_{H2}	FL_{H2}	F_{H2}	FR_{H2}	R_{H2}	BR_{H2}
B_{H2}	1							
BL_{H2}		0.85	0.10					0.05
L_{H2}			0.83				0.17	
FL_{H2}	0.05			0.70				0.25
F_{H2}	1							
FR_{H2}		1						
R_{H2}			0.40				0.60	
BR_{H2}			0.12	0.20				0.68

(b)

Figure 6.15: (a) Confusion matrix for pose estimation on 3D object category dataset for the category car for height H1. (average accuracy 51%); (b) Confusion matrix for pose estimation on 3D object category dataset for the category car for height H2. (average accuracy 58%).

object for learning and test on the remaining 3 objects, for three scalings. Their confusion matrix for eight viewpoints (L, BL, B, BR, R, FR, F, FL) shows an average accuracy of 70%.

Min et al. [SSSFF09] have tested their 3D Class Model on 160 images (5 objects, 16 viewpoints and two scalings) of the 3D object category dataset for the category ‘car’. Similar to [ANB09] they used the other 5 objects of the dataset for learning. They achieve an average accuracy of 67% on eight viewpoints (L, BL, B, BR, R, FR, F, FL).

6.7 Discussion

When analyzing the results of the pose estimation we see that three main problems occur:

- Neighboring views: It is hard to distinguish between neighboring views e.g. P10 - P6 - P14 of the ETH-80 or FL_{H1} - F_{H1} of the 3D object category dataset. One reason is that there are no major differences in the 2D shape, another reason is the loss of depth information in 2D images.

- Pose ambiguity: Due to the lack of depth information, we have to deal with pose ambiguity. It is hard to distinguish if the object looks towards the camera or in the opposite direction e.g. P17-P11 (front-right and back-right) or P13-P15 (front-left and back-left).

- Symmetry: In the case of symmetric objects such as cars, several poses are hardly distinguishable just on the basis of 2D shape information. On the one hand, the silhouettes of cars look similar for e.g. L_{H1}-R_{H1} or B_{H1}-F_{H1} or B_{H2}-F_{H2}. On the other hand the 2D shape properties of windows, wheels or lights are also similar for those poses.

There are several ways to overcome these problems. Depth information could help in many cases. By the use of the 3D Gaussian Category Model, the depth information, 2D and 3D spatial relations between object parts and different aspects are already known (also for the 2D Gaussian Aspect Models). Furthermore, when active object categorization is possible, a second view could be used to verify a pose hypothesis. Especially for such an application it is important to know the 3D nature of an object and the relations between views. With such an active object categorization system, the problem of neighboring views and pose ambiguity might be solved in most cases. For the remaining problem of symmetry, a combination of appearance and shape information could be the solution for many object categories such as e.g. car (the red back light identifies the rear of the car).

What do we gain by learning a 3D model from stereo images and using that for pose estimation? By learning from stereo image sequences we achieve one model for all views. Only the final model has to be annotated with regard to orientation. Thus, we overcome the problem of annotating a huge set of individual images and their relations. Most of the mentioned multi-view and 3D model based methods which build their models from static images rely on view annotations. In our approach we just have to move a camera around the 3D model and compute the views directly, which also allows to produce new views. By using a stereo setup, depth information and 3D geometry of an object as well as camera poses can be computed and used for learning, detection and pose estimation.

Whereas most of the current state-of-the-art approaches split the dataset in training data and test data, our 3D model is completely separated from the dataset. We learn the 3D Gaussian Contour Category Models from our own GRAZ-STEREO-BASE-xx dataset of toy objects, i.e. we do not learn on the same dataset that we use for the evaluation of our pose estimation algorithm. We even learn the category model from toy objects and our approach is still applicable to real world objects as it is demonstrated for the category 'car'.

6.8 Conclusion

This chapter presents a pose estimation algorithm for 2D input images solely based on shape information. The method uses a probabilistic 3D Gaussian Contour Category Model based on Gaussian Mixture Models of 1D manifolds in 3D. 2D Gaussian Aspect Models are then generated using an Unscented Transformation. I have introduced a novel voting procedure to compute a pose hypothesis for an object in a given 2D input image by integrating 2D shape information and geometric information in a sophisticated similarity measure between GMMs.

I evaluate our pose estimation on two well known datasets on four different categories: the ETH-80 (cow, horses and dogs) and the 3D object category dataset (car). The experiments on the ETH-80 dataset show the applicability of pose estimation based on shape information for the categories 'cow' (average accuracy 68%), 'horse' (average accuracy 64%), and 'dog' (average accuracy 59%)) for 17 views (standard eight views + nine views from above). On the 3D object category dataset I achieve an average accuracy of 58% (H2) and 51% (H1) for the category cars.

Future work may focus on several aspects: A combination of this pose estimation algorithm and object categorization on 2D images based on a 3D Gaussian Contour Category Model would be interesting. Furthermore, one big goal is the integration in an active object categorization system.

7 Discussion and Conclusion

This thesis investigates the problem of 3D object categorization based on three main aspects:

- (How) can we model 3D shape for specific objects?
- (How) can we learn one single, pose-invariant 3D category model based on shape information?
- (How) can we use such a 3D category model for pose estimation in 2D images?

As a short summary of the outcomes, we can answer above questions with 'yes'. This thesis presents a 3D object categorization system based on 3D shape information. Section 7.1 gives a summary of the developed object categorization and pose estimation system and provides an overview of the outcomes. Afterward, I discuss limitations as well as possible improvements (Section 7.2) and I give an outlook on future work (Section 7.3).

7.1 Summary - Outcomes

I have developed a comprehensive object categorization system based on 3D shape information. It consists of three main parts: 3D geometric shape modeling by stereo reconstruction, object categorization based on 3D shape information, and pose estimation on 2D images based on a 3D category shape model.

In the first part I have presented a stereo reconstruction framework that can generate '3D contour clouds' of objects, which are sets of reconstructed 3D contour fragments (1D manifolds in 3D). For this, it was mandatory to develop the following novel methods:

- A novel stereo correspondence algorithm integrating epipolar information, 2D shape context information, and an ordering constraint into one single cost matrix.
- The concept of 3D shape context for matching 3D contour fragments in order to reduce outliers in '3D contour clouds'.

In the second part I have presented a novel probabilistic framework based on Gaussian Mixture Models for 3D contour fragments for learning one single 3D shape model per category. This framework makes use of '3D contour clouds' for specific objects based on the reconstruction approach developed in the first part. The major results are:

- A 3D Gaussian Contour Category Model, which consists of partitions of Gaussian Mixture Models including a novel similarity measure hypothesis testing framework based on local shape information and 3D geometric information.
- The demonstration of the ability of the 3D object categorization method to categorize objects from categories with small inter-class difference and large intra-class variability.

In the third part, I have presented a pose estimation algorithm for objects in 2D images, which makes use of the 3D category shape model as learned in the second part. A probabilistic voting procedure computes pose hypotheses between 2D Gaussian Aspect Models and the 2D input image based on a Gaussian Mixture Model representation. The following results were obtained:

- A novel pose estimation algorithm based on 2D Gaussian Aspect Models computed from a 3D Gaussian Contour Category Model and 2D Gaussian Contour Models for input images. Moreover, I have developed a voting procedure, which combines geometric information and local shape information in a novel similarity measure, which has been introduced in a hypothesis testing framework.

- An experimental evaluation on the well-known ETH-80 dataset and the 3D object category dataset demonstrates the applicability of 3D category models based on shape information for pose estimation.

In addition to these three main parts, I have presented a new dataset for the task of stereo reconstruction and 3D object recognition and 3D object categorization, which comprises

- The GRAZ-STEREO-BASE-EYE, and the GRAZ-STEREO-BASE-EYE-TURNTABLE datasets which consist of stereo image sequences of objects of (up to) four categories: 'toy horses', 'toy cows', 'toy dogs' and 'toy cars', and the GRAZ-STEREO-BASE-30 with objects from the category 'humans'.

- Calibration data, 2D contour information as well as estimated camera poses for all stereo image sequences of the dataset.

7.2 Limitations - Improvements

Some problems and possible improvements have already been discussed in the discussion sections of the particular chapters. Now, I discuss general limitations of the system with regard to further applications.

In my opinion the strongest limitation of the whole system is that it requires a long chain of steps until the final object categorization and pose estimation system is achieved. Another drawback is the stereo reconstruction framework based on stereo image sequences, which itself requires many data processing steps. Due to the underlying Structure-and-Motion framework, there is the limitation to stereo image sequences, where an object is presented in front of a homogeneous background. A system which works on monocular image sequences of real world objects would be preferable, as more such data is available. Current research on multi-body Structure-and Motion leads in this direction. Thus, also the size of the datasets (number of object, number of categories, real world objects) can be enlarged, as a huge amount of such data is already available.

A limitation of the 3D category model is the random feature selection algorithm. This algorithm is a simple way to find a first subset of discriminant features. However, feature selection algorithms are normally used as a first step during learning to reduce the feature space. On the one hand, such a feature selection can be improved by using e.g. boosting for selection of discriminative subsets. On the other hand, by a more sophisticated learning algorithm as e.g. Adaboost, more variability in the dataset as e.g. arrangements of legs may be learned. However, even without such a learning step, we already achieve good results.

In the experimental evaluation of the pose estimation algorithm we see the limitations relying on shape information only. For many different poses shape information is enough, but when poses have too similar 2D shape (neighboring views, pose ambiguity and symmetry) it is hard to differ between them. Possible solutions such as depth information, active object categorization, and the combination with appearance information are discussed in Section 6.7.

7.3 Perspective

Future work may focus on various aspects. Some of them have already been discussed in the last section in order to improve the whole system. In this section, I want to go one step further and consider the application in an active object categorization system.

I have already mentioned the importance of category and pose hypotheses for an active categorization system, which uses several views of an object for a robust identification. In addition to a category hypothesis, a pose hypothesis is important for planning the further steps (how to move around an object and which view will be chosen as the next one). For such category and pose hypotheses a pose-invariant representation for a category has many advantages. In this thesis I focus on object categorization of 3D models by 3D Gaussian Contour Category Models and I demonstrate its application to pose estimation in 2D images. For an integration in an active object categorization system, it would be of interest to extend our system by an object categorization system on 2D images based on our 3D Gaussian Contour Category Models.

List of Figures

List of Tables

Bibliography

[ANB09] M. Arie-Nachimson and R. Basri. Constructing Implicit 3D Shape Models for Pose Estimation. In *Proceedings International Conference on Computer Vision (ICCV)*, 2009.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):509–522, 2002.

[Bou] Jean-Yves Bouguet. Camera calibration toolbox for matlab.

[CDF$^+$04] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Proceedings European Conference on Computer Vision (ECCV)-Workshop on Statistical Learning in Computer Vision*, pages 1–22, 2004.

[CKLP07] Han-Pang Chiu, L.P. Kaelbling, and T. Lozano-Perez. Virtual training for multi-view object class recognition. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[CSD$^+$09] Forrester Cole, Kevin Sanik, Doug DeCarlo, Adam Finkelstein, Thomas Funkhouser, Szymon Rusinkiewicz, and Manish Singh. How well do line drawings depict shape? In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, volume 28, August 2009.

[CT99] T. Cootes and C. J. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–573, 1999.

[DFRS03] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics (Proceedings SIGGRAPH)*, 22(3):848–855, 2003.

[DPP09] Renaud Detry, Nicolas Pugeault, and Justus H. Piater. A Probabilistic Framework for 3D Visual Object Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31:1790–1803, 2009.

[DR07] Doug DeCarlo and Szymon Rusinkiewicz. Highlight Lines for Conveying Shape. In *Proceedings Int. Symp. on NPAR*, 2007.

[DWW04] Joeri De Winter and Johan Wagemans. Contour-based object identification and segmentation: Stimuli, norms and data, and software tools. *Behavior Research Methods, Instruments, & Computers*, 36:604–624(21), November 2004.

[EG07] H. Ebrahimnezhad and H. Ghassemian. Robust motion and 3D structure from space curves. In *Proceedings International Conference on Information Sciences, Signal Processing and their Applications*, 2007.

[FFJ$^+$08] V. Ferrari, L. Fevrier, F. Jurie, , and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(1):36–51, January 2008.

[FK10] Ricardo Fabbri and Benjamin B. Kimia. 3D curve sketch: Flexible curve-based stereo reconstruction and calibration. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1538–1545, 2010.

[FPZ03] R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 264–271, 2003.

[FTG04] Vittorio Ferrari, Tinni Tuytelaars, and Luc Van Gool. Integrating multiple model views for object recognition. In *Proceedings IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, June 2004.

[GGG03] J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures. In *Proceedings International Conference on Computer Vision (ICCV)*, volume 1, pages 487–493, 2003.

[GSCO07a] Ran Gal, Ariel Shamir, and Daniel Cohen-Or. Pose-oblivious shape signature. *IEEE TVCG*, 13(2):261–271, 2007.

[GSCO07b] Ran Gal, Ariel Shamir, and Daniel Cohen-Or. Pose-Oblivious Shape Signature. *IEEE TVCG*, 13(2):261–271, 2007.

[Her04] C. Hernández. *Stereo and Silhouette Fusion for 3D Object Modeling from Uncalibrated Images Under Circular Motion.* PhD thesis, Ecole Nationale Supŕieure des Télécommunications, May 2004.

[HP10] Peter Holzer and Axel Pinz. Mobile Surveillance by 3D-Outlier Analysis. In *Proceedings Asian Conference on Computer Vision (ACCV) - Visual Surveillance Workshop*, 2010.

[HRW07] D. Hoiem, C. Rother, and J. Winn. 3d layoutcrf for multi-view object class recognition and segmentation. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007.

[Hum00] J. E. Hummel. Where view-based theories break down: The role of structure in shape perception and object recognition. *E. Dietrich & A. Markman (Eds.). Cognitive dynamics: Conceptual change in humans and machines*, pages 157–185, 2000.

[IJL$^+$05] Natraj Iyer, Subramaniam Jayanti, Kuiyang Lou, Yagnanarayanan Kalyanaraman, and Karthik Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37(5):509–530, 2005.

[JCC09] Zhaoyin Jia, Yao-Jen Chang, and Tsuhan Chen. Active View selection for Object and Pose Recognition. In *Proceedings International Conference on Computer Vision (ICCV) - 3dRR-09*, 2009.

[JU96] Simon Julier and Jeffrey K. Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, Robotics Research Group, University of Oxford, 1996.

[JV05] Bing Jian and B.C. Vemuri. A robust algorithm for point set registration using mixture of Gaussians. In *Proceedings International Conference on Computer Vision (ICCV)*, volume 2, pages 1246–1251, 2005.

[KFR04] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Symmetry Descriptors and 3D Shape Matching. In *Symposium on Geometry Processing*, 2004.

[Kov] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[KPNK03] Marcel Körtgen, G. J. Park, Marcin Novotni, and Reinhard Klein. 3D Shape Matching with 3D Shape Contexts. In *Proceedings Central European Seminar on Computer Graphics*, 2003.

[KSP07] Akash Kushal, Cordelia Schmid, and Jean Ponce. Flexible Object Models for Category-Level 3D Object Recognition. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE Computer Society, 2007.

[KYS07] Saad M. Khan, Pingkun Yan, and Mubarak Shah. A Homographic Framework for the Fusion of Multi-view Silhouettes. In *Proceed-*

ings International Conference on Computer Vision (ICCV), Rio de Janeiro, October 2007.

[LB83] David G. Lowe and Thomas O. Binford. Perceptual Organization as a Basis for Visual Recognition. In *Proceedings AAAI*, 1983.

[LH05] Marius Leordeanu and Martial Hebert. A Spectral Technique for Correspondence Problems using Pairwise Constraints. In *Proceedings International Conference on Computer Vision (ICCV)*, volume 2, pages 1482 – 1489, 2005.

[LHM00] Chien Ping Lu, Gregory D. Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22:610–622, 2000.

[LHS07] Marius Leordeanu, Martial Hebert, and Rahul Sukthankar. Beyond Local Appearance: Category Recognition from Pairwise Interactions of Simple Features. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[LS03] Bastian Leibe and Bernt Schiele. Analyzing appearance and contour based methods for object categorization. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II – 409–15 vol.2, 2003.

[LS04] Bastian Leibe and Bernt Schiele. Scale-Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search. In *Proceedings DAGM-Symposium*, 2004.

[LS10] Jörg Liebelt and Cordelia Schmid. Multi-View Object Class Detection with a 3D Geometric Model. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1688–1695, 2010.

[LSS08] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3D Feature Maps. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[LViAN10] Meizhu Liu, Baba C. Vemuri, Shun ichi Amari, and Frank Nielsen. Total Bregman divergence and its applications to shape retrieval. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3463–3468, 2010.

[MS09] Mona Mahmoudi and Guillermo Sapiro. Three-dimensional point cloud recognition via distributions of geometric distances. *Graph. Models*, 71(1):22–31, 2009.

[OBS04] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph.*, 23(3):609–612, 2004.

[OFCD] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. *ACM Trans. Graph.*

[ÖLF09] Mustafa Özuysal, Vincent Lepetit, and Pascal Fua. Pose estimation for category specific multiview object localization. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 778–785, 2009.

[OMT05] Ryutarou Ohbuchi, Takahiro Minamitani, and Tsuyoshi Takei. Shape-similarity search of 3D models by using enhanced shape functions. *International Journal of Computer Applications in Technology (IJCAT)*, pages 70–85, 2005.

[OPZ06] A. Opelt, A. Pinz, and A. Zisserman. A Boundary-Fragment-Model for Object Detection. In *Proceedings European Conference on Computer Vision (ECCV)*, 2006.

[OPZ08] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision (IJCV)*, 80(1):16–44, 2008.

[PH02] Jong Seung Park and Joon Hee Han. Euclidean reconstruction from contour matches. *PR*, 35:2109–2124, 2002.

[Pin05] Axel Pinz. Object Categorization. *Foundations and Trends in Computer Graphics and Vision*, 1:255–353, 2005.

[PKG03] Mark Pauly, Richard Keiser, and Markus H. Gross. Multi-scale feature extraction on point-sampled surfaces. *Comput. Graph. Forum*, 22(3):281–290, 2003.

[PR09] Adrian M. Peter and Anand Rangarajan. Information Geometry for Landmark Shape Analysis: Unifying Shape Representation and Deformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(2):337–350, 2009.

[RLSP06a] F. Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 3D object modeling and recognition from photographs and image sequences. In *Towards category-Level object recognition.* Springer, 2006.

[RLSP06b] F. Rothganger, Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision (IJCV)*, 66, 2006.

[Ros05] Sheldon M. Ross. *Introductory Statistics.* Academic Press, Inc., Orlando, FL, USA, 2005.

[RP11] Vignesh Ramanathan and Axel Pinz. Active Object Categorization on a Humanoid Robot. In *Proceedings International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 235–241, 2011.

[SBC05] J. Shotton, A. Blake, and R. Cipolla. Contour-Based Learning for Object Detection. In *Proceedings International Conference on Computer Vision (ICCV)*, pages 503–510, 2005.

[SBC08] J. Shotton, A. Blake, and R. Cipolla. Multi-Scale Categorical Object Recognition Using Contour Fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(7):1270–1281, 2008.

[SCD$^+$06] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R.S. Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 519–528, 2006.

[SFF07] Silvio Savarese and Li Fei-Fei. 3D generic object categorization, localization and pose estimation. In *Proceedings International Conference on Computer Vision (ICCV)*, Rio de Janeiro, 2007.

[SFF08] Silvio Savarese and Li Fei-Fei. View Synthesis for Recognizing Unseen Poses of Object Classes. In *Proceedings European Conference on Computer Vision (ECCV)*, 2008.

[SGS10] Michael Stark, Michael Goesele, and Bernt Schiele. Back to the future: Learning shape models from 3d cad data. In *Proceedings British Machine Vision Conference (BMVC)*, Aberystwyth, Wales, 2010.

[SMKF04] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. In *Proceedings Shape Modeling International, (SMI)*, 2004.

[SP06] Gerald Schweighofer and Axel Pinz. Fast and globally convergent structure and motion estimation for general camera models. In *Proceedings British Machine Vision Conference (BMVC)*, 2006.

[SSFFS09] Hao Su, Min Sun, Li Fei-Fei, and Silvio Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *Proceedings International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009.

[SSGD03] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton Based Shape Matching and Retrieval. In *Proceedings Shape Modeling International, (SMI)*, 2003.

[SSP08] Gerald Schweighofer, Sinisa Segvic, and Axel Pinz. Online/Realtime Structure and Motion for General Camera Models. In *Proceedings IEEE Workshop on Applications of Computer Vision (WACV)*, 2008.

[SSSFF09] Min Sun, Hao Su, Silvio Savarese, and Li Fei-Fei. A multi-view probabilistic model for 3d object classes. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[SZM$^+$08] Kaleem Siddiqi, Juan Zhang, Diego Macrini, Ali Shokoufandeh, and Sven Dickinson. Retrieving Articulated 3-D Models Using Medial Surfaces. In *Machine Vision and Applications*, volume 19(4), pages 261–274, 2008.

[TFL$^+$06] Alexander Thomas, Vittorio Ferrari, Bastian Leibe, Tinne Tuytelaars, Bernt Schiele, and Luc Van Gool. Towards multi-view object class detection. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1589–1596, 2006.

[TV07] Johan Tangelder and Remco Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 2007.

[UMPB09] Markus Unger, Thomas Mauthner, Thomas Pock, and Horst Bischof. Tracking as Segmentation of Spatial-Temporal Volumes by Anisotropic Weighted TV. In *Proceedings International Conference* *on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, volume 5681, 2009.

[Wol03] Stephen Wolf. Color correction matrix for digital still and video imaging systems. Technical report, NTIA Technical Memorandum TM-04-406, 2003.

[WVRE08] Fei Wang, Baba C. Vemuri, Anand Rangarajan, and Stephan J. Eisenschenk. Simultaneous Nonrigid Registration of Multiple Point Sets and Atlas Construction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(11):2011–2022, 2008.

[XJ96] Lei Xu and Michael I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Comput.*, 8(1):129–151, 1996.

[YKS07] Pingkun Yan, Saad M. Khan, and Mubarak Shah. 3D Model based Object Class Detection in An Arbitrary View. In *Proceedings International Conference on Computer Vision (ICCV)*, Rio de Janeiro, 2007.