

Multimodal Object Perception for Robotics

Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart
zur Erlangung der Würde eines Doktors der
Ingenieurwissenschaften (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von

Björn Browatzki
aus Herrenberg

Hauptberichter:	Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl
Mitberichter:	Prof. Dr. rer. nat. Christian Wallraven
Tag der mündlichen Prüfung:	24. Oktober 2014

ABSTRACT

The ability to recognize and manipulate unknown objects is crucial for any robot to successfully work in natural environments. Object recognition and categorization is a very challenging problem, as three-dimensional objects often give rise to ambiguous, two-dimensional views. This dissertation investigates how multisensory integration methods can be employed to tackle this issue in robotic applications. First, the presented approaches are motivated by discussing aspects of biological and computational object perception. To increase the understanding of the technical context, concepts regarding software development on modern robotic systems will be outlined. The document is separated into two parts, representing two strategies for multimodal object recognition utilizing state-of-the-art robotic hardware.

One part focuses on the benefits of employing range sensing technology for the categorization of everyday objects. We present a novel object dataset that served as a testbed to study classification performance combining 2D and 3D cues. The second part incorporates object manipulation into the recognition process. A perception driven object exploration method, implemented on the humanoid robot *iCub*, will be presented. In this setup, the robot turns and moves an object in its hand in order to seek out informative views, thereby optimizing the exploration sequence. It will also be shown that, instead of relying on purely visual information, taking motor actions that link object views into account allows the robot to resolve a significant amount of ambiguity.

ZUSAMMENFASSUNG

Die Fähigkeit, unbekannte Objekte zu erkennen und zu manipulieren, ist von entscheidender Bedeutung für Roboter, um in natürlichen Umgebungen arbeiten zu können. Da dreidimensionale Objekte allerdings aus unendlich vielen Ansichten betrachtet werden können, gestaltet sich ihre Erkennung und Klassifikation oft als äußerst schwierig. Im Rahmen dieser Dissertation sollen Wege untersucht werden, wie mittels Integration verschiedener Sensoriken dieses Problem gelöst oder vereinfacht werden kann. Zunächst werden Lösungsansätze motiviert, indem auf Aspekte der biologischen und maschinellen Objekterkennung eingegangen wird. Außerdem wird der technische Hintergrund für die Entwicklung von Software für moderne Robotersysteme vorgestellt. Das Dokument gliedert sich in zwei Teile. Diese behandeln zwei Strategien für multimodale Objekterkennung mittels aktueller Robotikhardware.

Der erste Teil konzentriert sich auf die Klassifikation von Alltagsgegenständen (z.B. Tassen oder Bücher) mittels Tiefenmessung. Hierfür wurde eine neue Objektdatenbank aufgenommen, die eingesetzt wird um die Kombination von 2D und 3D Information zu untersuchen. Im zweiten Teil wird Objektmanipulation in den Erkennungsprozess integriert. Für den humanoiden Roboter *iCub* wurde eine Methodik entwickelt, die aktuelle und vorherige Messungen berücksichtigt, um den Objekterkundungsprozess zu steuern. Der Roboter hält hierbei ein Objekt in der Hand und bewegt dieses mit dem Ziel neue Objektansichten mit zusätzlicher Information zu generieren. Zusätzlich wird gezeigt, dass durch die weitere Betrachtung der Roboterbewegung, welche ausgeführt wurde, um eine Ansicht in die andere zu überführen, wichtige Information gewonnen wird. Diese ist besonders in Situationen mit widersprüchlicher visueller Information hilfreich Objekte korrekt zu identifizieren.

CONTENTS

1	INTRODUCTION	1
1.1	Towards human-centered robots	2
1.2	Challenges in robotics	3
1.2.1	Requirements	4
1.3	Focus of this thesis	4
1.4	Object recognition	5
1.5	Multisensory perception	7
1.6	Coupling perception and action	8
1.7	Building advanced robotic systems	8
1.8	Outline and contributions	10
2	MULTISENSORY OBJECT CLASSIFICATION	13
2.1	Sensing devices	14
2.1.1	Digital color cameras	15
2.1.2	Time-of-Flight cameras	15
2.1.3	Structured light scanners	16
2.1.4	Stereo reconstruction	17
2.1.5	Laser scanning devices	18
2.2	Computer vision datasets	19
2.3	The MPI-IPA dataset	21
2.3.1	Sensor combination	24
2.4	Features	24
2.4.1	Based on color data	26
2.4.2	Based on depth data	29
2.5	A multisensory object classifier	31
2.5.1	Training of single SVM classifiers	31
2.5.2	Ensemble prediction	31
2.6	Results	33
2.6.1	Overall categorization performance	33
2.6.2	ROC curves and confusion matrices	35
2.6.3	Generalizability across views and number of objects	41
2.6.4	Real-world experiments	42
2.7	Conclusion	43
3	ACTIVE IN-HAND OBJECT RECOGNITION	45
3.1	Experimental setup	47
3.1.1	The iCub robot platform	48
3.1.2	Image segmentation and feature extraction	49
3.2	The View-Transition-Map	51
3.2.1	Keyframe segmentation	51
3.2.2	Building the VTM	52
3.2.3	Using the VTM to control the robot	53
3.2.4	Implementation overview	54
3.2.5	Evaluation on the iCub simulator	57
3.2.6	Evaluation on the iCub robot	60

3.2.7	Conclusion	62
3.3	A probabilistic framework for active object recognition	63
3.3.1	Viewpoint Control	64
3.4	Object exploration and learning	66
3.5	Recognition and motion planning	67
3.5.1	Recognition by localization	67
3.5.2	Action selection	69
3.5.3	Incorporating global object information	74
3.6	Evaluation	74
3.6.1	Evaluation in simulation	76
3.6.2	Real-world evaluation	78
3.7	Conclusion	85
4	DISCUSSION	87
4.1	Future work	89
4.1.1	Tactile feedback	89
4.2	General conclusions	95
	BIBLIOGRAPHY	97

LIST OF FIGURES

Figure 1	Possible tasks for cognitive, human-centered robots: a) assistive robots, b) playmate robots in child education, c) robots for mentoring and assistance in manipulation tasks, d) robots that teach movement exercises, e) personal robots for the elderly, f) robots for surveillance and protection of children and adults. From [Schaal 07].	2
Figure 2	Computer rendering of a room full of ‘chairs’ (from [Bülthoff 03]). How many chairs do you see and how many would a robot see?	5
Figure 3	PMD CamCube time-of-flight camera. Image source: http://en.wikipedia.org/wiki/File:PMDCamCube.jpg	16
Figure 4	Structured light depth recovery. Image source: http://en.wikipedia.org/wiki/File:3-proj2cam.svg	17
Figure 5	Microsoft Kinect. Image source: http://en.wikipedia.org/wiki/File:Xbox-360-Kinect-Standalone.png	18
Figure 6	Principle of a laser triangulation sensor. Image source: http://en.wikipedia.org/wiki/File:Laserprofilometer_EN.svg	19
Figure 7	The Caltech101 dataset. Image source: http://www.vision.caltech.edu/Image_Datasets/Caltech101/averages100objects.jpg	20
Figure 8	The RGB-D Object Dataset. Image source: rgbd-dataset.cs.washington.edu/imgs/rgbd_dataset2.png	20
Figure 9	Sensor setup of Care-O-bot [®] 3 for data acquisition. One stereo rig augmented with a range camera.	22
Figure 10	Color and depth images of categories in our dataset. First view of first object for each category. Number of exemplars in parentheses.	23
Figure 11	Illustration of the Pyramids of Histograms of Oriented Gradients (PHOG) extraction process. In this example, the feature contains three pyramid layers.	28
Figure 12	Extraction process of the self-similarity descriptor.	28
Figure 13	Shape index: Basic surface shapes are mapped to a real value.	30
Figure 14	The illustration shows the three color layers in Fourier space. The red rectangle indicates the high energy frequencies that form the Depth Buffer feature vector.	31
Figure 15	Composite classifier. 2D and 3D features are classified using support vector machines. Results are combined by multilayer perceptrons.	32
Figure 16	ROC curves for both modalities separately and in the combined case.	37
Figure 17	Output of SVM classification.	39

Figure 18	Confusion matrices: Combined cues (left), 3D only (middle), 2D only (right)	40
Figure 19	Classification results for increasing number of training objects per class. 18 views per object.	41
Figure 20	Classification rates for increasing number of training views per objects. Six objects per class.	42
Figure 21	Classification rates for two symmetrical objects. Symmetrical objects show low sensitivity to view count.	43
Figure 22	Classification rates for two asymmetrical objects. Asymmetrical objects benefit from additional views.	44
Figure 23	Recognized object classes in three exemplary scenes—the objects were not part of the dataset.	44
Figure 24	System components forming a perception action loop. . . .	47
Figure 25	The iCub humanoid robot. Implementation and evaluation platform of the presented object recognition method.	48
Figure 26	Segmentation process: A background model is trained on the area between the rectangles and applied to the area inside the smaller rectangle. Low values in the resulting probability map indicate the presence of an object.	49
Figure 27	Left: Joint space diagram. Recorded keyframes are placed in two-dimensional joint space. Each circle represents 45° of object rotation. Right: View-Transition-Map (VTM). Cells contain joint differences. Blue values (above main diagonal) are recorded, all remaining values derived from these. Arrow colors correspond to colors in joint space diagram. . . .	52
Figure 28	VTM lookup. The transition associated with the respective top matching keyframes is compared to the currently executed transition.	53
Figure 29	Schematic overview of the object recognition system using View-Transition-Maps.	54
Figure 30	Recognition process.	56
Figure 31	Objects used for evaluation in the iCub simulator. 24 rectangular boxes with differently colored sides. Shown are segmented keyframes used in the recognition process.	57
Figure 32	Confusion matrix for VTM recognition in simulation.	58
Figure 33	Confusion matrix for VTM recognition in simulation. Objects rotated by 30 degrees about ϕ	59
Figure 34	Recognition performance for different amounts of object rotation.	60
Figure 35	Objects used for evaluation on the real iCub robot.	61
Figure 36	Confusion matrix for VTM recognition of tea boxes.	62
Figure 37	Confusion matrix for VTM recognition of plastic cups. . . .	63
Figure 38	Illustration of the view sphere. Green area indicates accessible viewpoints on the view sphere. Viewpoint angles φ and θ are computed from gaze vector G and hand coordinate system N . Objects are located at the origin of N	64

Figure 39	Example of an exploration sequence executed to learn a new object. The dashed red line shows the exploration path on the view sphere. Keyframes are marked by black dots. . . .	66
Figure 40	In this representation of the view sphere regions with high expected information gain are indicated in red, low information gain in blue. We want to find the action that moves the objects from the current viewpoint to a viewpoint with high information gain.	69
Figure 41	Recognition sequences for a white cup marked with a blue and a red dot. Top left: exploration using motion planning. Bottom: recognition of the same cup but without planning. Top right: the six objects available in the dataset. For planned exploration, heatmaps visualize the expected utility of actions leading to respective positions on the viewsphere. Red color indicates regions with high expected information gain. Bar plots below the object views show the joint probability distribution at time t	73
Figure 42	Particle boosting based on global image statistics across learned objects. In this example, features from two different objects (triangles and squares) are marked in two-dimensional space. The three ellipses represent three clusters among these features. For an unknown feature z the probability of being a triangle or a square is calculated based on the evaluation of the clusters.	75
Figure 43	Iteration accuracy in simulation.	77
Figure 44	Iteration entropy in simulation.	78
Figure 45	Accuracy after 15 iterations for different amounts of object rotation.	79
Figure 46	Objects used for evaluation on the real iCub robot.	79
Figure 47	Results of object recognition experiments on the iCub. . . .	80
Figure 48	Recognition performance for the unmarked brown plastic cup after 10 iterations.	82
Figure 49	Observers viewpoints during 10 recognition trials. Dot sizes correspond to iteration number.	84
Figure 50	Experimental tactile fingertips on the iCub.	91
Figure 51	Time series showing intensity values read from tactile sensors during object grasping. Each row represents one sensor (12 sensors per finger).	92

Figure 52	Joint angles for the index and middle finger during grasping. For each finger, all three joints are summed up. Line colors correspond with object colors. Octopus: pink, soft ball: blue, hard ball: green, cup: grey.	94
-----------	--	----

LIST OF TABLES

Table 1	Classes in dataset.	25
Table 2	Descriptors	27
Table 3	Classification performance	34
Table 4	Results by class on our dataset.	35
Table 5	Results by class on RGBD dataset.	36

ACRONYMS

- VTM View Transition Map
- SVM Support Vector Machine
- RANSAC RANdom SAmple Consensus
- TOF Time-of-Flight
- CCD Charge-Coupled Device
- CMOS Complementary Metal–Oxide–Semiconductor
- SIFT Scale Invariant Feature Transform
- SURF Speeded-up Robust Feature
- HOG Histogram of Oriented Gradients
- PHOG Pyramids of Histograms of Oriented Gradients
- SC3D Shape Context 3D
- VGA Video Graphics Array
- CIE Commission internationale de l’Eclairage
- IIT Istituto Italiano di Tecnologia
- GMM Gaussian Mixture Model

CPU Central Computing Unit

DOF Degree Of Freedom

KNN K-Nearest Neighbour

BOW Bag Of Words

INTRODUCTION

What I cannot create, I do not understand.

— Richard Feynman

The idea of creating machines that resemble humans in shape and skills has inspired scientists, engineers and authors alike for generations. Science fiction authors have conceived countless visions of worlds in which robots assist, replace and even rebel against us.

The reality, however, is much less exciting. Robotic research is progressing slowly, struggling with a multitude of challenges from low level hardware to high level cognition. The approach to overcome the limits of current technology usually boils down to reducing the level of uncertainty the robot is facing. One way of doing this is by controlling the environment that robot is operating in and removing the human from the scene. Today most robots are employed in manufacturing, executing exactly predefined and preprogrammed tasks. Once a solution to the task is found, it can be repeated with no or little adaptation. This allows to operate at very high speed yet still maintaining a level of precision superior to any human worker. In these scenarios the human is not only removed from the setting, he is very often not even allowed to approach the robot. An industrial robot arm could severely injure someone not paying attention.

Such an approach, however, contradicts the initial vision of a human-like assistant. Instead of removing the human from the picture, we need to do the opposite: Bringing the robot closer to us—into our homes and work places. Robots need to be able to cope with our environment instead of having us design the world to suit the needs of the machines.



Figure 1: Possible tasks for cognitive, human-centered robots: a) assistive robots, b) playmate robots in child education, c) robots for mentoring and assistance in manipulation tasks, d) robots that teach movement exercises, e) personal robots for the elderly, f) robots for surveillance and protection of children and adults. From [Schaal 07].

1.1 TOWARDS HUMAN-CENTERED ROBOTS

An illustration of the roles a robot might adopt in the future is shown in Figure 1. It includes fields such as elderly care as well as rehabilitation or household assistance. All these tasks require the robot to possess a certain set of cognitive skills. Typical application would encompass domestic and workplace environments as well as for example hospitals and retirement homes. The world the robot operates in will not only be largely unknown, but it will also be changing constantly. The ability to cope with this kind of variation and to react appropriately to novel situations is a crucial requirement for any such machine. Learning and adaptation play a vital role in achieving this level of independence. Once robots and humans work in close contact it also becomes mandatory to assure safe interaction. Hardware and control needs to allow for compliant motion. Rigid mechanical elements can easily hurt or even kill a human being. Motor controllers must react and adapt to resistance, especially if motors are able to produce fast and powerful movements.

In many cases it can be required that humans touch, move or guide the robot. This imposes further restrictions regarding physical attributes such as size or weight.

Another important requirement is the ability to understand human commands. People will neither be able nor willing to specify tasks in a formal, technical way that would be easy to interpret computationally. The machine will have to process human language and recognize its semantic meaning. The same holds in reverse direction. Output has to be given in a human understandable way that does not assume any previous technical knowledge of the user.

1.2 CHALLENGES IN ROBOTICS

If we think of robots that are able to serve us in everyday life, we tend to attribute sophisticated cognitive abilities to them. However, cognition is not an isolated skill, but it emerges from a combination of various basic skills. The robotic research community has identified and addressed a large set of skills and requirements that need to be developed in order to accomplish high-level tasks as described above. In the following, I will list some of the main research areas and including issues. By no means is this intended to be a complete list of challenges research in cognitive robotics is facing.

LOCOMOTION

- Obstacle avoidance
- Navigation and mapping
- Bipedal walking

PERCEPTION

- Scene analysis and segmentation
- Object detection
- Object identification
- Object categorization
- Grasping and manipulation

HUMAN INTERACTION

- Face recognition
- Facial expression recognition
- Speech analysis and synthesis
- Detection of attention
- Interpretation of gestures and body postures

1.2.1 REQUIREMENTS

For these skills, we can identify a set of common requirements. Depending on the specific skill these requirements must be fulfilled to a certain degree in order to truly master the skill. These include:

- Compliant design
- Multi-sensory integration
- Unsupervised and supervised learning
- Closed loop perception/action coupling

1.3 FOCUS OF THIS THESIS

In this work, I am going to tackle a set of the key issues in robotics mentioned in Section 1.2. As I will point out in Section 1.4, objects play a central role in the way we perceive and interact with the world. Therefore, the main focus of this work is put on the recognition and classification of *three-dimensional, real-world objects* by a robotic agent. For us humans object perception is an active, multi-modal process. I will address this idea in more detail in Section 1.5 and Section 1.6.

With these biological and psychological findings in mind, I am particularly interested in how *sensor fusion* and the *coupling of perception and action* can enhance robotic object perception. Especially the fact that three-dimensional objects often give rise to ambiguous, two-dimensional views serves as a motivation to search for perceptive processes that integrate complementary sensory cues and incorporate active methods.



Figure 2: Computer rendering of a room full of ‘chairs’ (from [Bülthoff 03]). How many chairs do you see and how many would a robot *see*?

1.4 OBJECT RECOGNITION

Humans are able to recognize, classify and manipulate objects even without having seen this particular object instance before. From the appearance of the unknown object alone, enough information is extracted to assign it to a specific object category and infer how to manipulate it. However, many tasks that appear to be very simple for a human being turn out to pose a major difficulty for a robotic system. As an example, detecting an object in a room and grasping it, represents one of the challenges that still can only be solved under limited assumptions. For cognitive robots, however, these abilities are key requirements. Since most tasks such a machine is facing consist of interacting with objects, the ability to detect, recognize and classify these at a level of expertise comparable to a human being is a crucial requirement.

How many chairs do you count in Figure 2? What appears like a simple task turns out to raise fundamental questions about what actually constitutes an object and how to separate objects from the background and each other. Following [Gibson 66], many authors argue that "objectness" arises through the ability of an agent to manipulate a confined part of the world. Formulating the problem in this way has inspired a number of implementations in robotic research. For example,

Fitzpatrick et al. segment objects from the background by having a robot poke and move object candidates [Fitzpatrick 03]. In [Montesano 08] a robot acquires object interaction skills by learning affordances through observation and imitation. Kraft et al. [Kraft 08] build object models based on low-level visual features that are assembled to 3D representations through incorporation of information obtained by grasping and rotating objects. More recently Schiebener et al. [Schiebener 11] are able to segment, learn and recognize objects by allowing a robot to push object candidates and use the resulting feedback to verify hypotheses.

Authors have addressed the question about "objectness" also in a purely visual context. For example, Heitz and Koller [Heitz 08] separate the world into *stuff* and *things*. This terminology was made popular (in this context) by Forsyth et al. [Forsyth 96] but is an established differentiation in philosophy and linguistics. Homogeneous regions with unifying color or texture are referred to as *stuff*, whereas *things* are rather characterized by distinct and individual shapes. In [Alexe 10] Alexe et al. continue the discussion and base their interpretation of "objectness" strongly on the idea of visual saliency [Itti 98]. A salient image area is a part of the image that stands out from the rest of the image. Most methods run filters across the image that are sensitive to certain visual features (color, texture) [Itti 98, Hou 07, Gao 07] or compare small image regions with their neighborhoods [Liu 07, Cheng 11]. This low-level approach is contrasted by authors that argue that objects and background can only be discriminated by taking into account semantic knowledge about objects and object classes as well as spatial relations (e.g., [Leibe 04, Li 09]).

The process of recognizing an object from a visual image is a main subfield of computer vision. Computer vision deals with the question of how to find and implement computational methods to extract semantic information from a still image or video sequence. This field of research received a tremendous amount of attention in the last decades. Early work regarding object recognition was dealing with the basic question of how to represent an object. Biederman et al. [Biederman 87] argued that objects are encoded in the human brain in form of three-dimensional structures. This belief was replicated in computational implementations by composing objects out of basic three-dimensional shapes, so called Geons. Later, psychophysical studies by Bülthoff et al. [Bülthoff 92] gave evidence that aspects of object representation in humans are rather explained following a two-dimensional view-based model. The view-based interpretation has become increasingly popular since then. In the beginning of computer vision, it was common to manually design models of object classes using expert rules and heuristics. Regarding only

flat 2D views of objects, however, made it possible to describe objects exploiting statistical methods [Bishop 06, Schölkopf 02]. Instead of being crafted by an engineer, the object description is inferred from data. There no longer is an explicit understanding of an object or object class. An object is whatever matches a pattern that is learned from a body of example data.

1.5 MULTISENSORY PERCEPTION

What is often missed when designing a computational recognition system, is the fact that in nature perception of the environment does not solely rely on visual input? Learning and recognizing objects is a multisensory process for humans. Grasp, touch, smell and other cues are combined into a multimodal representation of objects.

In fact, nature has endowed many species that employ sensors beyond vision such as infrared-sensitive receptors (snakes/bees) or echolocation (bats). Many species live in environments that are better explored through modalities other than vision. They have developed senses that serve special purposes crucial to their survival.

Coming back to human perception and looking at how infants explore objects makes us understand how basic and natural multi-modal processing is to us. Infants grasp things, turn and throw them around, smell them, put them in their mouths and so on. By doing this, they acquire a wealth of different, heterogeneous stimuli. Especially the haptic modality supplies many object and material properties (e.g., size, weight, compliance, etc.) that help to create more meaningful representations of form and function than relying on vision alone [Lederman 09]. Many object properties, materials for example, are learned through haptic exploration. Metal is cold, heavy and hard, whereas fabric is light and deformable. Supplementary proprioceptive information puts objects into a body centered reference frame which provides cues like object size without even touching.

In engineering the fusion of multisensory input plays an important role. There are many problems that require integration of different sensor modalities. In modern positioning systems, for example, GPS signals and measurements from accelerometers are combined into a joint prediction. By fusing both sensor inputs these systems operate more reliably. In situations when only one modality is available (e.g., indoors) the other one will compensate. As pointed out in Section 1.2, localization is an important topic in robotics as well. Sensor fusion techniques are

already commonly applied for robot localization. Thrun et al. [Thrun 05] provide an extensive study. Probabilistic Monte Carlo methods proved to work well as sensor fusion methodology in these scenarios. We shall see in Section 3.3 that this approach can also be applied very well to cue integration in object recognition tasks.

1.6 COUPLING PERCEPTION AND ACTION

Very often perception is regarded as a static process in which sensory signals arrive and the receiver has no way to impact the information he obtains. Being able to steer the input stream, however, greatly extends and often simplifies perception. When we deal with an object and are unsure about some attribute, we immediately plan actions to gain more or better input. This could be touching to learn about temperature or material. It could be grasping to find out the weight or a manipulation, such as shaking a can to 'feel' its fill level. If it is too dark, we get up to switch on the light or move to a brighter place. We must realize, however, that this requires a considerable amount of planning. What action should be executed? Will the action be worth the effort? Is the execution safely possible under the current conditions? To answer these questions there needs to be a way to predict the expected outcome of an action.

In computer vision and robotics, active methods have been applied to a number of problems. Active vision is a field in computer vision that deals with the question of how to move the camera or focus area in order to maximize the information gain [Aloimonos 88, Dutta Roy 04]. Similarly, the robot localization methods described above, can be extended such that based on the current state, optimal positions for further measurements are estimated [Burgard 97, Kümmerle 08, Fairfield 08]. The object recognition algorithms I am going to present in Chapter 3 allow the robot to manipulate the object. I will also address the question of how to plan and select the next best action.

1.7 BUILDING ADVANCED ROBOTIC SYSTEMS

Modern systems could not be developed without relying on established libraries, frameworks, architectures and algorithms. Being familiar with state-of-the-art methods and software is a crucial requisite to building new systems, especially when

reaching the level of complexity of an intelligent humanoid robot. In the following, I will lay out the technological environment of our implementations. The systems I am working on can be split into several architectural layers. These layers build on top of each other—increasing in degree of abstraction.

HARDWARE

We find the actual robot and computer hardware at the bottom of the stack. All setups consists of a heterogeneous collection of custom robot hardware, consumer and industry grade devices (e.g., cameras or range sensors) as well as desktop computers, servers, PC-clusters with varying platforms and architectures. I worked with two different robots/setups: Implementations in Chapter 2 are targeted for the service robot *Care-O-bot*[®] 3 and the methods presented in Chapter 3 are run and tested on the humanoid robot *iCub*.

OPERATING SYSTEM AND MIDDLEWARE

Our software runs on common desktop operating systems. Currently, most operating systems are supported, such as Windows, Linux, or MacOS. By using robot operating systems as middleware, a high level of platform independence is achieved. On the *Care-O-bot*, *ROS* (Robot Operation System) [Quigley 09] is used as middleware; on the *iCub*, the application software is run on *YARP* (Yet Another Robot Platform) [Metta 06]. These robot operating systems act as link between higher level modules that perform task specific purposes and the robot device drivers. Another important feature of these middleware systems is that software can easily be split into modules and the workload distributed across multiple computing nodes. This is very important as real-time performance is crucial.

TOOLKITS

I rely intensively on established libraries and toolkits. Reimplementing standard algorithms is not only extremely inefficient, it is also prone to errors. Software that is shared with a large community is usually well designed and thoroughly tested. A widespread library for computer vision related tasks is *OpenCV* [Bradski 00]. Throughout all implementations and projects I make use of it for handling images, basic mathematics, and most of the machine learning algorithms mentioned in the next paragraph. For more advanced mathematical representations and calculations *LibEigen* [Guennebaud 10] has proved to be well suited. In Chapter 2 I employ *PCL* (Point Cloud Library) [Rusu 11] for dealing with 3D point cloud data. All modules running on the

robots were written in C++. *Boost*¹, a collection of well-crafted C++ libraries facilitated development and led to more robust and platform independent code.

ALGORITHMS

Cognitive systems need to be able to adapt and react to sensory input from the environment. This requires supervised and unsupervised learning algorithms that recognize patterns from sample data and put new information into the context of previously obtained data. Such data driven representation and reasoning is applied to a wide range of problems in computer vision, robotics, and many other challenging areas. Statistical learning algorithms act as vital building blocks in our systems. For example, I apply classifiers, such as *K-nearest-neighbor*, *Support Vector Machines*, *Neural Networks* in Chapter 2; use *Gaussian Mixture Models* and *Graph Cuts* for image segmentation in Chapter 3. There are other algorithms used (*RANSAC*, *K-means clustering*, etc.) throughout our implementations. We will go more into detail on the actual methods in the respective sections.

1.8 OUTLINE AND CONTRIBUTIONS

This chapter gives a general introduction to our field of interest and provides background and motivation for the proposed systems and solutions in the following chapters.

In Chapter 2 I study the benefits of employing range sensing technology for categorization of everyday objects. I present a novel object dataset that served as a testbed to study classification performance combining 2D and 3D sensor data.

Chapter 3 focuses on active methods for object recognition. This approach incorporates object manipulation into the recognition process. This chapter details a perception-driven object exploration method, implemented on the iCub humanoid robot. In this setup, the robot turns and moves an object in its hand in order to seek out informative views, thereby optimizing the exploration sequence. I will also show that, instead of relying on purely visual information, taking motor actions that link object views into account allows the robot to resolve a significant amount of ambiguity.

¹ <http://www.boost.org/>

I conclude with a discussion of the results in Chapter 4 and look at possible ways to continue and extend the presented approaches.

The main contributions of this work include:

- A novel, large-scale 2D + 3D object dataset. Targeted as testbed for research in object classification.
- Investigation of the combination of 2D and 3D input for object classification.
- A probabilistic framework for active object recognition integrating visual and proprioceptive information.
- A motion planning algorithm to optimize object exploration sequences.

MULTISENSORY OBJECT CLASSIFICATION

*If we spoke a different language,
we would perceive a somewhat different world.*

Ludwig Wittgenstein

In robotics, the advent of new sensor technologies, such as time-of-flight sensors or laser scanners has opened up new possibilities for shape processing going beyond 2D color cameras for object recognition in a robotic context. It is already common to employ laser scanners or other ranging devices [Borenstein 97, Thrun 05] for tasks such as navigation and self-localization. Here, I want to study the effect of incorporating range, or 3D¹, data into object classification methods. The target platform for our evaluation is the service robot *Care-O-bot*[®] 3 [Parlitz 08, Reiser 09] developed by the Fraunhofer IPA in Stuttgart, Germany (cf. Figure 9). It is equipped with two color cameras and a time-of-flight (TOF) camera. The TOF camera emits modulated infrared light and uses the phase shift of the reflected light to measure the distance to the reflection surface.

Classification requires training data in order learn the common attributes that describe a specific category. To acquire this data, a large dataset containing 154 object exemplars belonging to 18 categories was recorded. The object categories were chosen from typical household and office scenarios. I then evaluated classification performance on this dataset using data gathered by 2D and 3D sensors and studied how cue combination of both cues can lead to enhanced categorization results.

Object recognition based on 3D geometry information has been extensively studied in the past. Early approaches focused on surface curvature to match a query object with previously learned sample objects. Besl and Jain [Besl 85] used Gaussian curvature and mean curvature at surface locations to classify points into categories

¹ As I use algorithms from computer graphics developed for 3D applications, I will refer to the range data as "3D" rather than 2.5D.

like peaks, pits, ridges and valleys. Faugeras and Herbert [Faugeras 86] detected primitive features such as points, lines, planes, and quadric patches by analyzing the surface curvature.

As the literature on object categorization based on 2D information is vast, I chose to focus here on the state-of-the-art in 3D object processing. Most approaches based on 3D information deal with recognizing previously seen objects [Hetzl 01]. Often the focus lies on detecting specific objects in complex scenes using local surface descriptors on key points [Johnson 99, Ruiz-Correa 01, Frome 04]. An extensive survey of 3D object recognition techniques is given by Campbell and Flynn in [Campbell 01]. Classification tasks, posing the challenge of assigning class labels to unknown objects, have gained less attention so far. Ruiz-Correa et al. introduced symbolic surface signatures to label surface regions in range scans [Ruiz-correa 03]. Objects were recognized by assigning regions to the object classes snowmen, rabbits, and dogs. A part-based classification approach was proposed by Huber et al. [Huber 04]. Eight classes of vehicles were separated into front, middle, and back part. Based on Spin Images, shape parts are recognized and the object class inferred using a generative model.

The majority of literature on 3D object classification deals with cases in which the object geometry is available in the form of polygon meshes. A common scenario is similar shape retrieval from 3D object databases [Funkhouser 03, Shilane 05]. Numerous approaches have been proposed to compare 3D models and calculate a measure of similarity. A selection of popular algorithms includes Spherical Harmonics [Kazhdan 03], Extended Gaussian Images [Horn 84] or Shape Distributions [Osada 02]. Many of the algorithms proposed in the context of shape retrieval can be applied to incomplete objects and point clouds as well. Bustos et al. provide an exhaustive overview of shape matching approaches [Bustos 06].

2.1 SENSING DEVICES

Arguably the most important aspect in recognition and classification is the type and the quality of the input data. All subsequent processing builds on the information that is contained in this data. In conclusion, understanding and choosing sensing devices is a crucial step that will affect the recognition performance and limitations to a great extent. I will briefly introduce the most common sensing technology humanoid and service robots are currently equipped with.

2.1.1 DIGITAL COLOR CAMERAS

Digital cameras have become extremely popular in the last few years. They have become so affordable and small that we carry them around in our pockets in form of compact cameras or cellphones. Hence, it is not surprising that digital color cameras find wide application in modern robotics as well.

There are two prevalent sensor technologies: charge coupled devices (CCD) and complementary metal-oxide semiconductors (CMOS). Charge-coupled devices were invented in the 1970s in the Kodak labs. A CCD is a matrix of photon collecting semiconductors. During exposure, photons are accumulated into an electric charge. This charge is proportional to the light intensity and is then read out line-wise and quantized into digital values.

Complementary metal-oxide semiconductors have replaced CCD technology in most low-cost, consumer applications. Due the mass production of CMOS semiconductors for micro controllers and CPUs, manufacturing cost has dropped significantly. Whereas for CCDs the electric charge in the capacitors is read off from the chip pixel wise and processed afterward, sensor elements on CMOS chips are equipped with additional circuitry to read out all elements directly on the chip.

The basic technology for obtaining color images is the addition of a so called Bayer filter. It works by splitting each pixel into four pixels, each sensitive to either red, green or blue light (two for green due to the higher sensitivity of the human eye to green colors).

2.1.2 TIME-OF-FLIGHT CAMERAS

Conceptually similar to digital color cameras are time-of-flight (TOF) cameras. TOF cameras capture depth values instead of light intensities. Modulated infrared light is emitted and based on the phased shift of the reflected light the round-trip time of the light beam is measured. From this time and the known speed of light the distance to the reflecting surface can be calculated. The entire scene is captured at once, which allows high frame rates (up to 100fps). The system is very compact since no additional hardware or further processing or calibration is necessary, as for example for stereo vision. Depth values are obtained from a TOF sensor as easily as color information from an RGB camera. The lack of moving parts makes it also more robust and less expensive than, for example, laser scanning devices (cf. Section 2.1.5).



Figure 3: PMD CamCube time-of-flight camera.

However, TOF cameras suffer from a couple of drawbacks. Since these devices are working with light, the surface type is important. Transparent materials, such as glass, and specular surfaces, such as polished metal, will not send the light beam back reliably and often lead to missing parts in the depth image. Another critical aspect is background light. Sun light has a 50 times higher illumination strength than the light emitted by the TOF camera. Even though constant background light can be subtracted from the measurements, it introduces noise. Since light emitted from two or more cameras will lead to interferences, the application of multiple TOF cameras at the same time is difficult. Solutions to this problem include using different modulation frequencies or time multiplexing, i.e., illuminating the scene iteratively by each camera.

2.1.3 STRUCTURED LIGHT SCANNERS

Structured light scanners reconstruct the 3D shape of a scene by projecting light pattern onto the scene and analyzing the resulting distortions. For this, additional cameras capture the projected pattern. There are multiple approaches for creating these patterns. Most common are stripe patterns, but also other shapes have shown to lead to good results. A similar technique is to use two laser beams and look at the interferences between the two patterns.

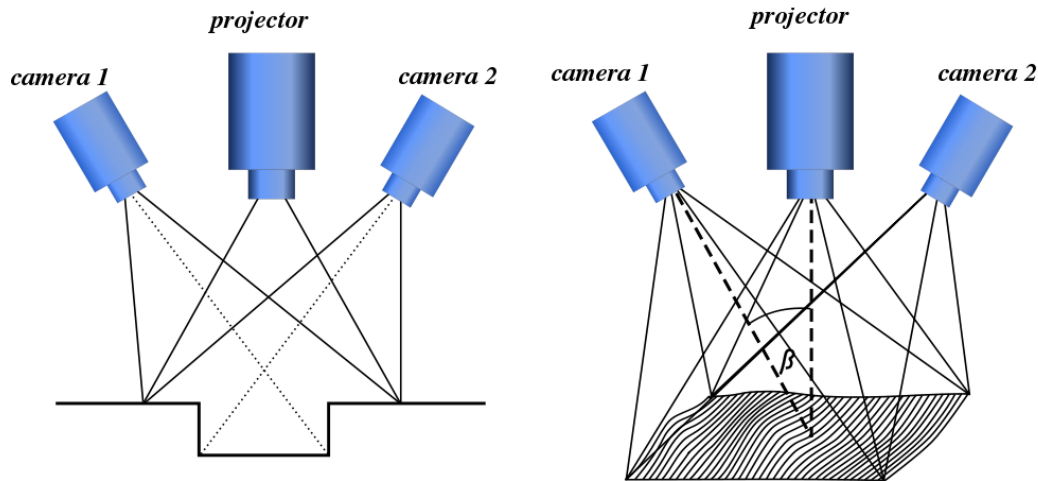


Figure 4: Structured light depth recovery.

It is possible to use infrared instead of visible light. This makes the scene illumination invisible, which is important if other vision tasks have to be performed at the same time. This would be a basic requirement for the application in a service robot scenario. Depending on the intended task, range and resolution can be adjusted. They are affected by the size of the projected pattern as well as its optical quality. The accuracy can range from few micrometers at very close range to millimeters at ranges of several meters.

One specific structured light system has recently received much attention: the Microsoft Kinect™, developed by the Israeli company PrimeSense. It was initially sold as an input device for the Microsoft Xbox 360™ (Figure 5) game console. The Kinect features a structured light sensor working with infrared light. It operates at a frame rate of 30Hz, producing colored depth images with a resolution of 640x640 pixels. Higher resolutions are possible at lower frame rates. Detailed specifications are not available due to its closed architecture. Being consumer grade hardware, the Kinect™ is very affordable and has found its way into many robotic and computer vision applications.

2.1.4 STEREO RECONSTRUCTION

Using stereo reconstruction, 3D images can be obtained by combining 2D images from two cameras. 3D coordinates for image pixels can be inferred if the relative translation and rotation of both cameras are known. Based on these camera parameters, corresponding image positions are projected as lines into 3D space. The intersection of the two lines defines the 3D coordinate of the image point. This



Figure 5: The Microsoft Kinect camera. A low cost depth sensor, initially intended as gaming input device, but grown into a very popular tool for research and engineering

process is referred to as triangulation. In the case of ideal pinhole cameras the projections are defined by epipolar geometry. In this case triangulation is a trivial calculation. However, in real cameras the projection is subject to small errors (e.g., lens distortion). In this case, lines do not necessarily intersect anymore and 3D coordinates have to be estimated. This is usually done by minimizing an error measure over all image points.

As mentioned above, one initial step is finding correspondences between 2D coordinates in both images. This can be achieved by extracting interest points and matching these between the images. Interest points can be edge and corner points but also more advanced features, such as SIFTs [Lowe 99] or SURFs [Bay 08] (cf. Section 3.1.2).

2.1.5 LASER SCANNING DEVICES

3D laser scanning is mostly referred to as LIDAR (LIght Detection and Ranging). This technology is employed in robotics mostly for environment perception and navigation (cf. SLAM [Durrant-Whyte 06, Bailey 06]) but also for object perception (e.g., [Triebel 07, Kümmerle 08]). LIDAR systems emit a near infrared, visible, or ultraviolet light beam. The laser beam is usually reflected by an oscillating mirror, so that it travels across the scene. Returning light is collected by a photodetector. It consists of a CMOS or CCD chip that outputs an intensity matrix similar to a common 2D camera (cf. 2.1.1). These devices allow range sensing at high distances and high precision but are usually more expensive than other range sensing hardware.

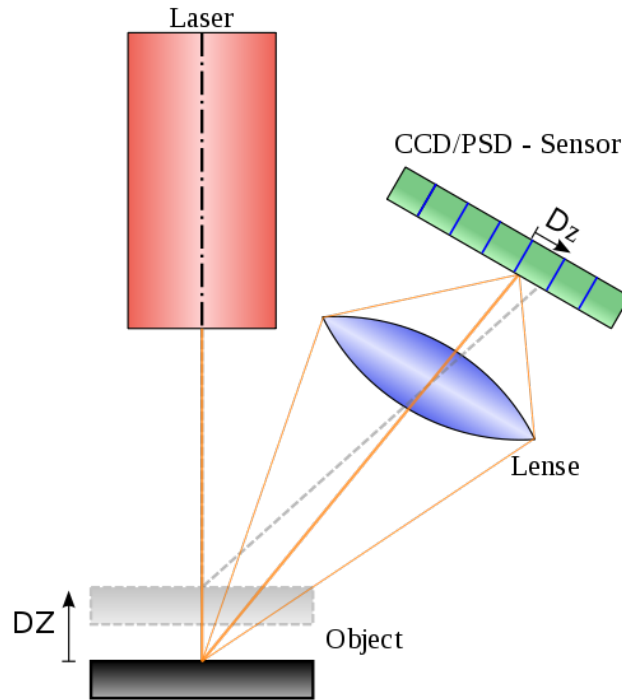


Figure 6: Principle of a laser triangulation sensor. Two object positions are shown.

2.2 COMPUTER VISION DATASETS

Our task is to categorize objects using 2D and 3D information. As pointed out in the beginning, a set of training data is needed as a description of the respective object categories. Common properties within training classes can then be inferred using statistical machine learning algorithms.

There are many well-known image datasets for the evaluation of object categorization algorithms based on 2D information. Arguably one of the most popular ones in computer vision research is the Caltech-101 dataset [Fei-Fei 04]. It contains images from 101 categories with high intra-category variability. An extended version, the Caltech-256 dataset [Griffin 07], contains 256 categories with a proper taxonomic structure. Other popular databases include the Graz-01 [Opelt 04] database, the ETHZ shape dataset [Ferrari 06], or the PASCAL Visual Object Class database [Everingham 09] or the MIT-CSAIL database (also known as LabelMe) [Torralba 04]. Furthermore, there are specific databases focusing on certain object types, such as cars (UIUC [Agarwal 04]), horses (INRIA Horses [Ferrari 08]), or faces [Gross 05].

Most computer vision literature focuses exclusively on information obtained from 2D images. In that case the existing databases offer a suitable test-bed to evaluate algorithms and compare results. However, for 3D data obtained, for example, from laser scans, or as in our case range cameras, there are few comparable

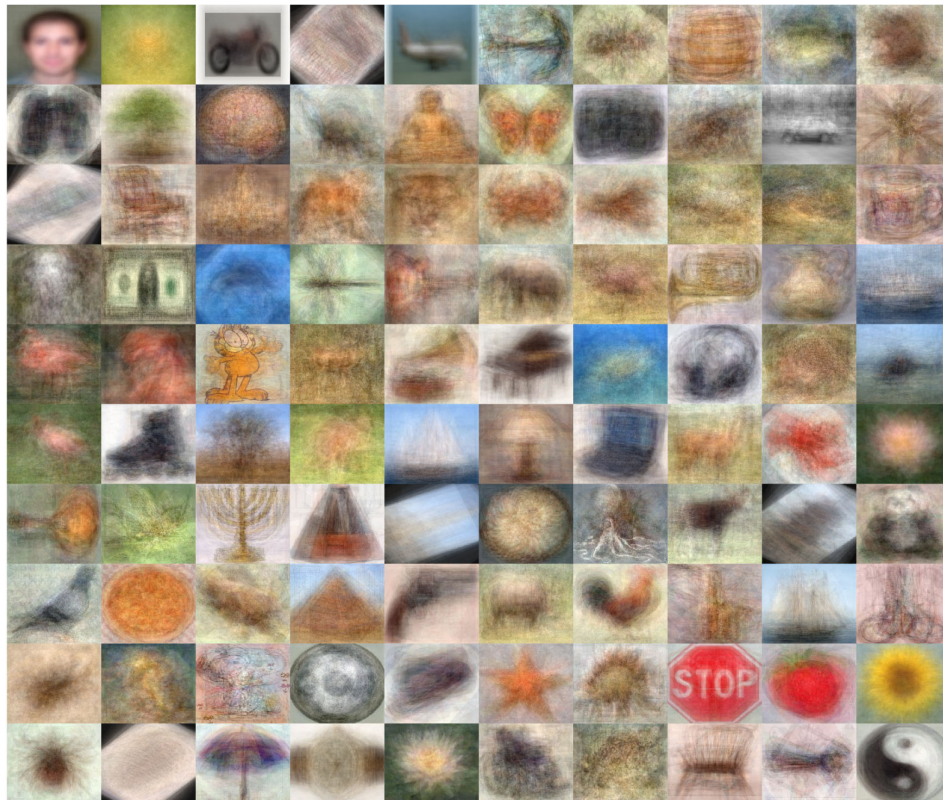


Figure 7: Caltech101 dataset. Averages of all images for each category.



Figure 8: The RGB-D Object Dataset.

databases, or the data is rather sparse. Sun et al. [Sun 10] have collected a data set containing three object categories (mice, mugs, staplers) with 10 object instances each. They obtain depth information using a structured-light stereo camera. Lai et al. [Lai 11] have recently introduced the RGB-D dataset (Figure 8). This database contains color and depth information of 300 objects from 51 categories of household objects and is organized in a hierarchical structure. The dataset was used to train an object recognition system capable of detection 20 specific instances of objects as well as 4 object classes (bowl, cup, coffee mug, soda can) in cluttered scenes.

There are larger datasets for 3D model comparison, such as the Princeton Shape Benchmark [Shilane 05]. These datasets contain complete meshes of 3D models instead of scanned views of real objects. It is possible to simulate a scanning process and obtain views, however, I believe that working on synthetic data is an insufficient replacement for real-world data. Furthermore, these datasets do not contain 3D models with 2D appearance information (textures, materials, etc.). One is often forced to decide between 2D and 3D data.

2.3 THE MPI-IPA DATASET

To serve as a suitable test set for 2D+3D classification, there is a set of basic requirements that need to be met by the dataset:

1. Joint representation: Color and shape information must both be available for the same object.
2. Class size: Each class must contain a minimum number of exemplars for training.
3. Number of classes: Too few classes does not allow to study how performance differs across classes and which object types are best represented by which feature type or combination pattern.
4. Type of objects: To assure that results are transferable to actual performance in the intended service robot application scenario, types of objects should be contained that are typically encountered in domestic environments.

Unfortunately, no available dataset could fulfill all these requirements. The dataset by Sun et al. [Sun 10] contains only three object categories. The RGB-D dataset

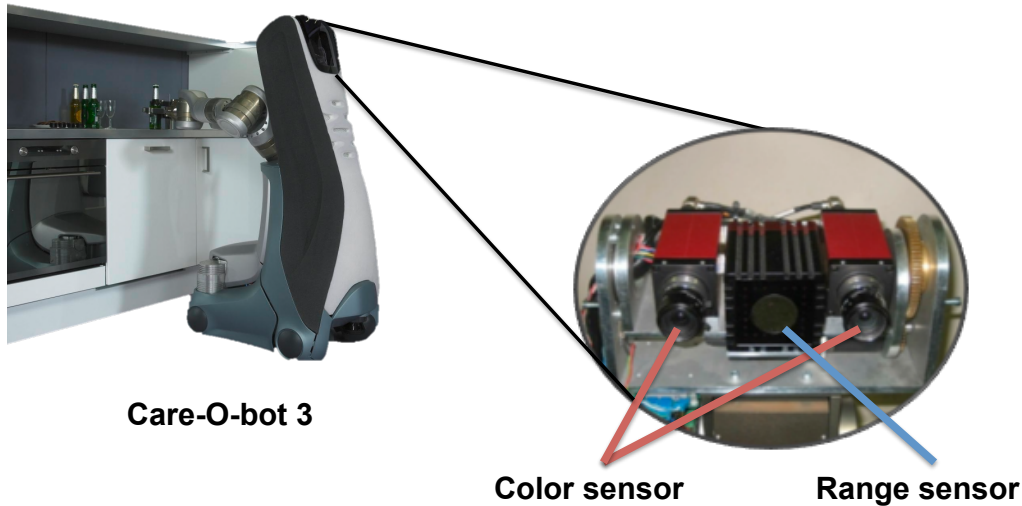


Figure 9: Sensor setup of Care-O-bot[®] 3 for data acquisition. One stereo rig augmented with a range camera.

[Lai 11] consists of a sufficient number of categories, but most of them include only very few object instances. Furthermore, when starting with my initial work, this dataset did not exist yet. Since the number of classes is very high and I was interested in the results of the classification algorithm on this dataset, I ran evaluations despite the sometimes low number of training instances per class. Results are reported in Section 2.6.

To fully satisfy the listed requirements, a new object dataset was created in collaboration with Fraunhofer IPA. We recorded 18 categories of objects that are likely to be encountered by a robot operating in a household environment. Each category contains between 3 and 14 objects—most with 9-10 objects. In Figure 10 all categories are depicted including the respective number of exemplars. Each object was put on a step-motor-controlled turntable in their default orientation (except for silverware objects and scissors, which were propped up on a stand) and we recorded views every 10° around the vertical axis, yielding 36 views per object. In total, the dataset contains 154 objects with $154 \times 36 = 5544$ views. Every view consists of two high-resolution (1388×1038 px) 2D color images and a range scan obtained from a PMD[™] CamCube 2.0 time-of-flight camera. The resolution of the range images is 204×204 px with an accuracy of approximately $\pm 1\%$ with respect to the measured distance. The dataset is available for download ².

² www.kyb.mpg.de/~browatbn

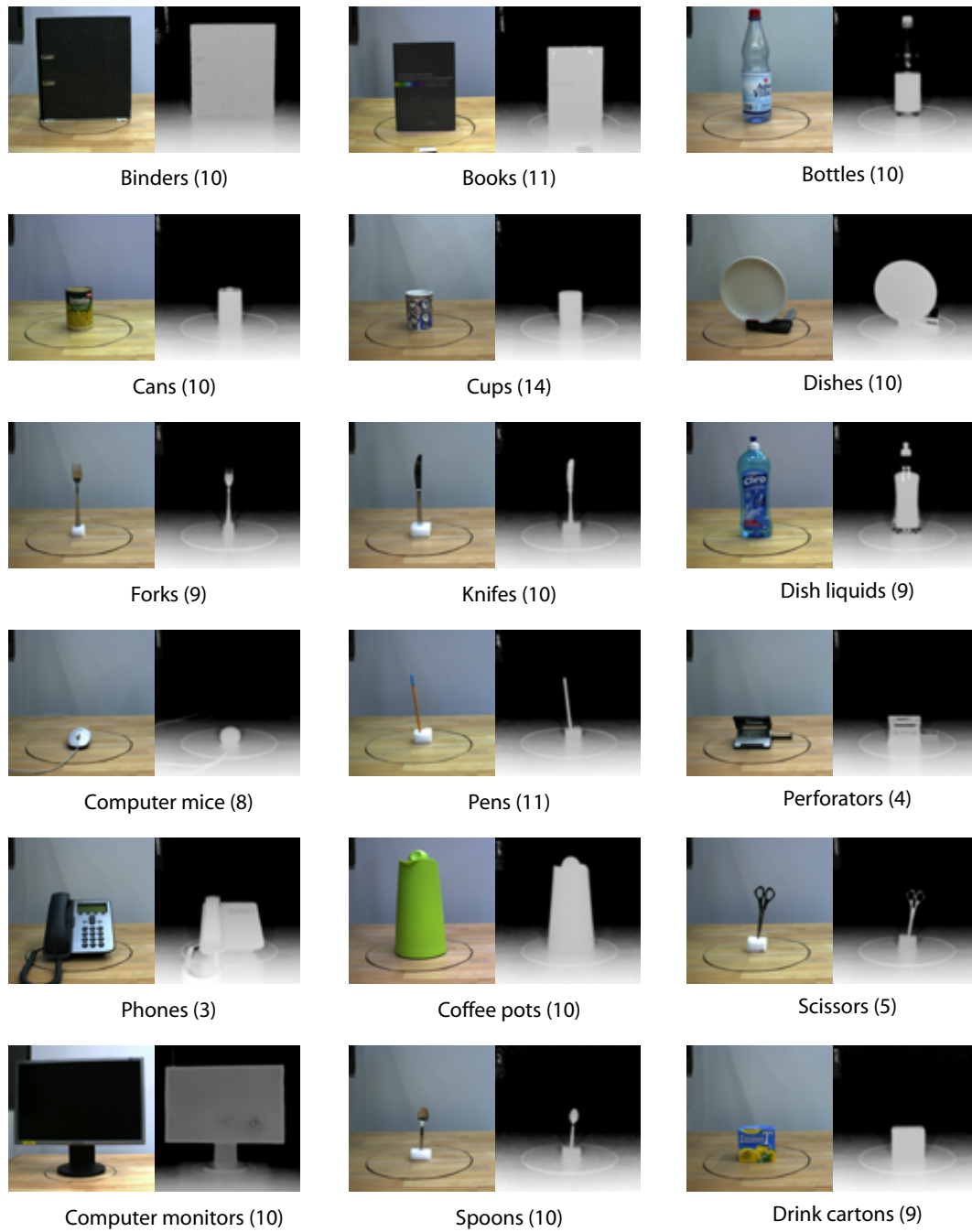


Figure 10: Color and depth images of categories in our dataset. First view of first object for each category. Number of exemplars in parentheses.

2.3.1 SENSOR COMBINATION

To obtain a joint representation of color and range data, we employed sensor fusion according to [Fischer 10]. Both cameras were calibrated to map color values onto 3D coordinates of the depth image. We estimated intrinsic and extrinsic parameters for both cameras using standard calibration tools such as Bouguet’s Matlab calibration toolbox³. Intrinsic parameters are used to correct lens distortions. From the extrinsic parameters the translation vector $T_{\text{tof}}^{\text{col}}$ and the 3×3 rotation matrix $R_{\text{tof}}^{\text{col}}$ are determined. A 3D coordinate x_{tof} retrieved from the time-of-flight sensor can then be mapped onto the corresponding 3D coordinate in relation to the 2D color camera x_{col} by evaluating

$$x_{\text{col}} = R_{\text{tof}}^{\text{col}} \times x_{\text{tof}} + T_{\text{tof}}^{\text{col}} \quad (1)$$

To compute the corresponding 2D color image coordinate u_{col} [pixels] from the 3D coordinate x_{col} [meters], x_{col} is normalized by dividing it through its z-coordinate before applying the intrinsic matrix M as follows

$$u_{\text{col}} = M \times x_{\text{col}} \quad (2)$$

The procedure is repeated for each pixel of the 3D time-of-flight camera. In order to take advantage of the 1388×1038 high resolution color image, the 204×204 low resolution range image from the time-of-flight camera is resized by a factor of 3 using bilinear interpolation prior to the sensor fusion process. The result is an image of size of 612×612 pixels containing 3D coordinates and color information for each pixel (similar to the VGA resolution of the Microsoft[®] Kinect[™]). By artificially increasing the image size of the 3D range image, more color information is preserved during fusion. This is due to the fact that each interpolated range value is assigned a color value from the native color image.

2.4 FEATURES

In the proposed approach, objects are represented by extracting a set of features from both 2D and 3D data. These features capture specific properties of the presented objects, such as information on edges, gradients, interest points, or color (based on the 2D image), and information, such as curvature or surface orientation, obtained from the 3D camera. It should be noted that the expressiveness of

³ www.vision.caltech.edu/bouguetj/calib_doc/

Table 1: Classes in dataset.

Class	# Objects	Class	# Objects
Books	11	Forks	9
Bottles	10	Knives	10
Binders	10	Pens	11
Cans	10	Perforators	4
Coffee pots	10	Phones	3
Cups	14	Scissors	5
Plates	10	Screens	10
Dish liquids	9	Spoons	10
Drink cartons	9	Mice	8

the features *varies between object classes*, and one feature might give a confident description of objects belonging to one category, whereas for other categories the same feature might not provide much helpful information. The feature representations are used as training samples for different classifiers. For each feature and each object category, one classifier is trained. Each classifier is then used to yield a prediction on whether an unknown object is a member of a specific object class. The quality of the features for all categories is analyzed and used as a parameter for the final fusion of the single classification results. Thus, the recognition results are based on an ensemble of shape and appearance features that best capture the properties of the respective object class.

Four 2D descriptors from the color images as well as four 3D descriptors from the range scans are extracted. In both cases, the set of descriptors is intended to exploit different properties, aiming at providing complementary information. For 2D data the feature set comprises Speeded Up Robust Features (SURF) [Bay 08], Pyramids of Histograms of Oriented Gradients (PHOG) [Bosch 07], Self Similarity Features [Shechtman 07], and color histograms.

The 2D descriptors are widely used in current computer vision research. For 3D data, however, the choice of feature descriptors is not as large. To cover a wide range of shape characteristics, these descriptors were selected: 3D Shape Context (SC3D) [Körtgen 03], Depth Buffer [Heczko 02], Shape-Index Histograms [Koenderink 92], and MD2 Shape Distributions [Osada 02].

SURF, Self Similarity, and SC3D are local feature descriptors. To transform these local features into a global descriptor, the well-known bag-of-words (BoW) method (e.g., [Fei-Fei 05]) is applied. A collection of feature vectors is taken from various objects across all object classes and clustered in the respective feature space. The resulting set of cluster points (vocabulary) is used to quantify the local features. A histogram is created that represents the local feature distribution in respect to the vocabulary entries. Vocabulary size was kept at 50 throughout all experiments. Varying this parameter did not lead to significant changes in the quantization process.

I experimented with incorporating absolute object size (in meters) as an additional cue. This, however, did not result in an increase in classification performance. This is probably due to the fact that size information is already implicitly encoded by some of the 3D descriptors. The size of regions of interests for 3D shape contexts, for example, is defined by a fixed metric value. Consequently, features vector of larger objects will be slightly different from those of smaller objects with the same shape. If this behavior is considered an advantage or a disadvantage, ultimately only depends on the problem to be solved.

Most of the features listed above are descriptive enough for being used to build strong classifiers but still fast enough to compute so that real time application is feasible. Extraction times range between <1 ms for color histograms and ≈ 250 ms for Depth Buffers on a standard desktop PC with a 3GHz dual-core CPU and 2GB RAM. In Table 2 average extraction times for all descriptors are listed. Times were taken on a standard desktop PC with a 3GHz dual-core CPU and 2GB RAM. All descriptors are implemented in C++.

2.4.1 BASED ON COLOR DATA

2.4.1.1 *Speeded-Up Robust Feature (SURF)*

SURF [Bay 08] is a detector and descriptor for local image features. Interest points are detected in the gray scale image by searching for maximum values of the determinant of the Hessian matrix as introduced in [Mikolajczyk 01] and [Lindeberg 98]. The process is speeded-up by working on integral images [Viola 01]. To achieve invariance in regard to the scale of the image content, the Hessian matrix is created with box filters of increasing size. Similar results are often achieved by iteratively sub-sampling the image, thus creating an pyramid of images [Lowe 99].

Table 2: Descriptors

	Descriptor	# Dims	BoW	Time (ms)
2D	SURF	64	✓	63
	PHOG	150		15
	Self-Similarity	91	✓	110
	Color	225		<1
3D	Shape Distributions	64		31
	Shape Index	50		78
	Shape Context 3D	165	✓	234
	Depth Buffer	256		16

The SURF descriptor describes the image content in the neighborhood of the interest point. Haar wavelet [Haar 10] responses are computed for this area and concatenated into a normalized one-dimensional feature vector. Again, the computation is speed-up by relying on integral images.

SURF is conceptually similar to the popular SIFT features by Lowe et al. [Lowe 99] but, according to [Bay 08], outperform these in terms of computation speed and repeatability performance.

2.4.1.2 *Pyramids of Histograms of Oriented Gradients (PHOG)*

The PHOG descriptor [Bosch 07] gives a global description of the image gradient structure. An image is subdivided iteratively into four subimages, creating a pyramid structure containing multiple layers of image cells. These cells are becoming smaller with each layer. To describe the image content, Histograms of Oriented Gradients (HOG) [Dalal 05] are extracted for each cell. HOGs are one-dimensional vectors representing the distribution across edges orientations within a certain image region. The PHOG extraction process is illustrated in Figure 11.

2.4.1.3 *Self-Similarity*

This feature captures internal self-similarities in images. It was proposed by Shechtman et al. [Shechtman 07]. In contrast to most other representations of image con-

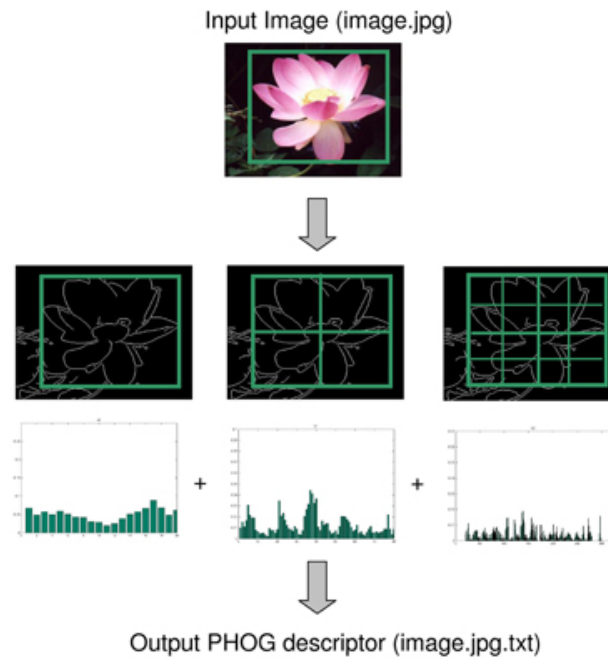


Figure 11: Illustration of the Pyramids of Histograms of Oriented Gradients (PHOG) extraction process. In this example, the feature contains three pyramid layers.

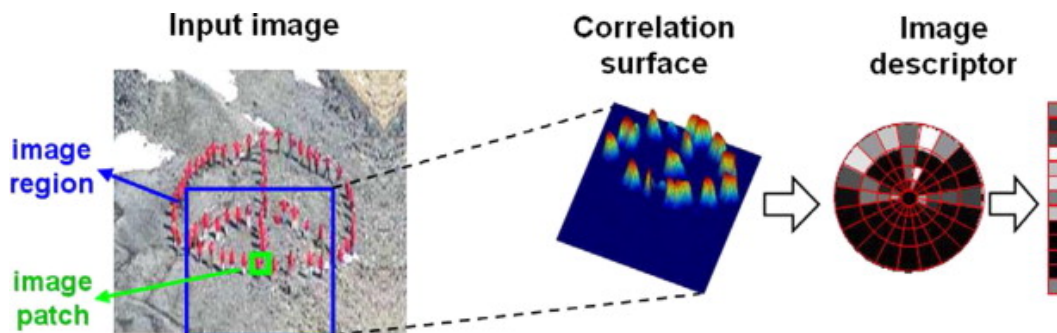


Figure 12: Extraction process of the self-similarity descriptor.

tent, this approach does not rely on describing the color or gray value information at certain locations but rather measures co-occurrences within an image.

For a certain image location a self-similarity descriptor is calculated by correlating an image patch with its local neighborhood. This results in a correlation surface which is transformed into log-polar coordinates and binned into a two-dimensional histogram. Histogram dimensions represent angle and (log-)distance to the center of the feature point. This process is depicted in Figure 12. Self-similarity descriptors are calculated for each position on a regular grid across the image.

2.4.1.4 *Color Histogram*

Color histograms are a simple and efficient way to describe the global color distribution of an image. First, the image is transformed to CIE L*a*b* [Wyszecki 68] color space. Then, all pixels are assigned to a histogram bin according to their a* and b* value. The L* value is omitted as only color but not luminance should be regarded. The two-dimensional histograms are normalized and transformed into a one-dimensional vector.

2.4.2 BASED ON DEPTH DATA

2.4.2.1 *Shape Distributions*

Shape distributions are an easy to compute global descriptor that was originally proposed for 3D model retrieval [Osada 02]. Shape characteristics are sampled according to specific shape functions. One such function, for example, collects Euclidean distances of random pairs of surface points. This function was termed D2. All distances are binned into a normalized one-dimensional histogram. This histogram captures the distribution of a certain shape feature.

Instead of D2 other shape functions can be applied: for example, regarding the angle between three random point (A1) or the distance between random surface points and the objects centroid (D1). However, Osada et al. [Osada 01] reported that the random point distance criterion (D2) leads to the best results while requiring the least computation time. Consequently, I chose this metric in my implementation.

2.4.2.2 *Shape Index Histograms*

The shape index [Koenderink 92] indicates the type of the surface shape (saddle, ridge etc.) around point p. Principal curvatures K_1 and K_2 at p are transformed into a scalar value s:

$$s = \frac{\pi}{2} \arctan \frac{K_2 + K_1}{K_2 - K_1} \quad (K_2 \geq K_1) \quad (3)$$

The shape index s ranges between $[-1, 1]$. In Figure 13 a selection of shape types are shown with their corresponding shape index value. Shape indices are calculated across the object surface and the value distribution transformed into a one-dimensional histogram.

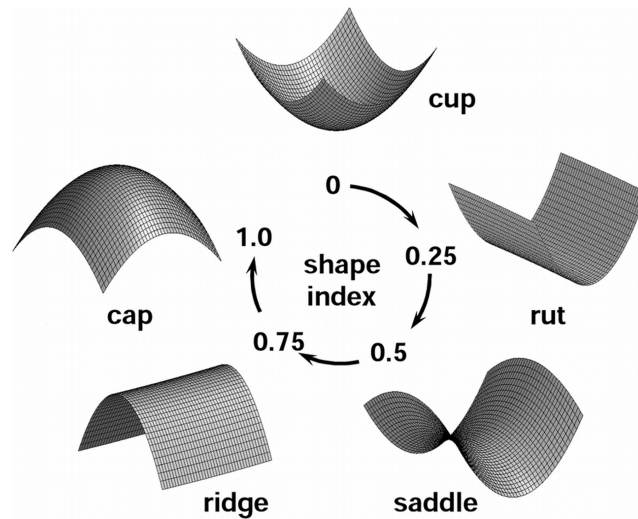


Figure 13: Shape index: Basic surface shapes are mapped to a real value.

2.4.2.3 *Shape Context 3D*

3D shape contexts are a straightforward extension of 2D shape contexts proposed by Belongie et al. [Belongie 02]. This shape descriptor was introduced by Körtgen et al. [Körtgen 03]. I use a simplified version of the original SC_{3D} descriptor. Histograms of point distributions around key points are created. Instead of three dimensions for elevation, distance, and rotation I only use elevation and distance bins. This was done since the origin for rotation around the key point normal is not defined. This issue was resolved in the original paper by repeating the descriptor for multiple origins and searching for a matching subpart during recognition. This is, however, only possible for keypoint matching not training of classifiers, as in our case.

2.4.2.4 *Depth Buffer*

The Depth Buffer [Heczko 02] feature is another fast to compute global shape descriptor. A depth image is created by rendering distance values into a gray scale 2D image. Next, the image is transformed into the Fourier space and a 16x16 sub window in the center containing high energy frequencies is selected. The amplitudes are normalized and written into a one-dimensional feature vector. This process can be regarded as a form of data compression. The resulting feature is similar to a low-pass filtered and downsampled version of the original depth image. It still contains most of the shape information, however, at a much lower dimensionality.

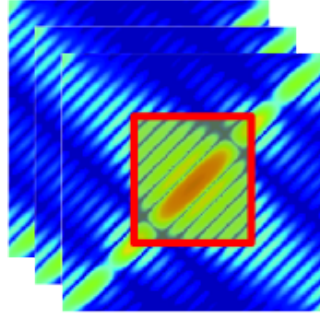


Figure 14: The illustration shows the three color layers in Fourier space. The red rectangle indicates the high energy frequencies that form the Depth Buffer feature vector.

2.5 A MULTISENSORY OBJECT CLASSIFIER

A composite classifier (illustrated in Figure 15) was created to merge 2D and 3D feature data. It consists of two layers: On the first layer, support vector machines (SVMs) perform classification for each feature type and each class separately. The second stage merges the individual SVM results for each class using multilayer perceptrons.

2.5.1 TRAINING OF SINGLE SVM CLASSIFIERS

After extracting features, one support vector machine for each feature type and each class is trained. If n is the number of classes and d the number of feature types, $n \times d$ classifiers are created in total. SVM parameters are optimized through cross-validation on the training set. Each of these classifiers predicts whether an unknown sample is likely to be a member of a certain object class based on specific feature type. It is obvious that we do not always obtain a consistent prediction across different feature types for a certain class. One class might be more dependent on information supplied by a specific feature than a different class.

2.5.2 ENSEMBLE PREDICTION

To obtain a joint prediction from the different classifiers, a multilayer perceptron for each object class is trained. The outputs of the single classifiers are used as training samples. In our case, a training sample for the perceptron would be an eight-dimensional vector produced by the eight different descriptors. To evaluate the performance of either 2D or 3D data independently, the MLPs are trained

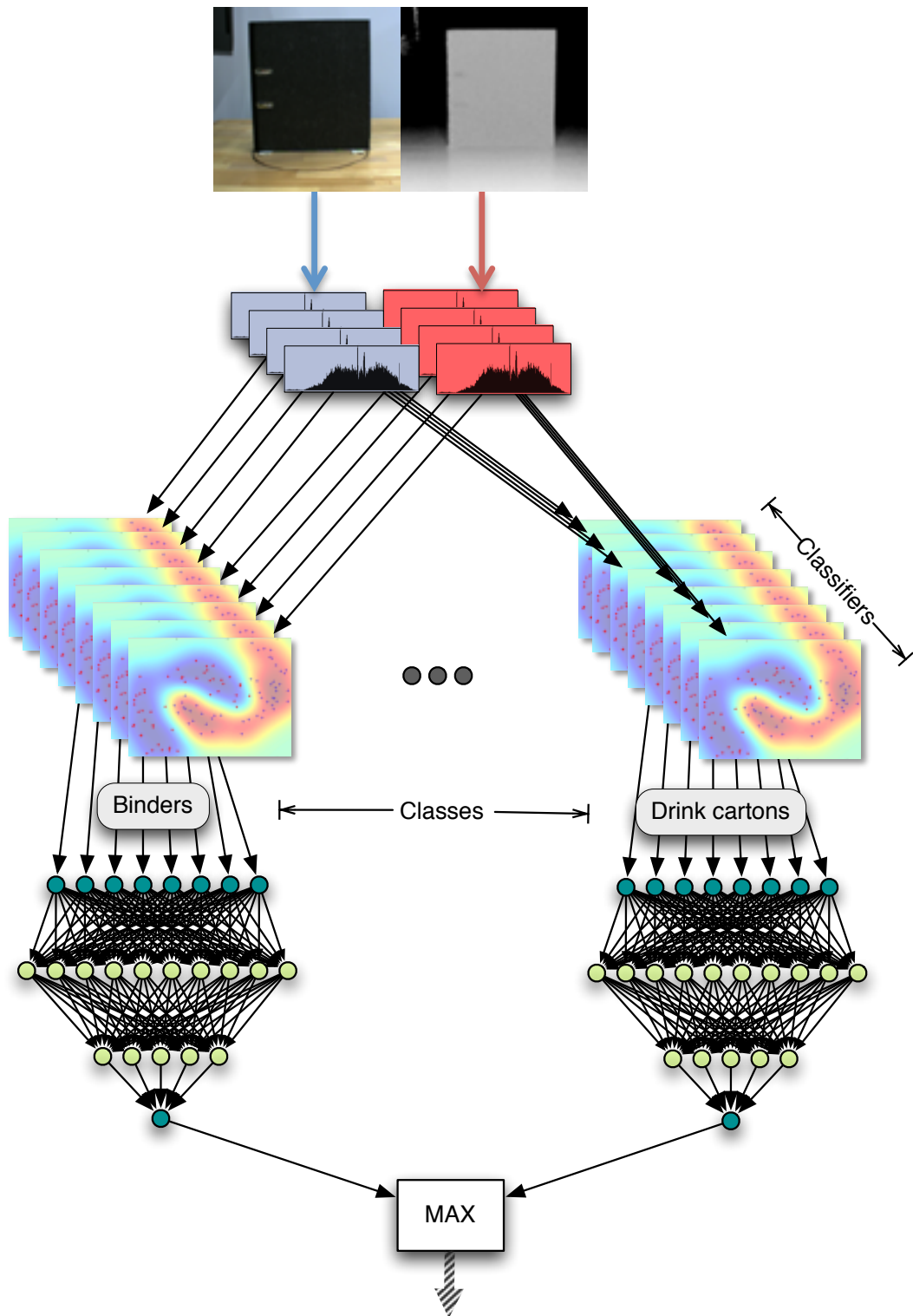


Figure 15: Composite classifier. 2D and 3D features are classified using support vector machines. Results are combined by multilayer perceptrons.

with the four-dimensional input vector retrieved from the four descriptors of one modality. The structure of the network is not defined a priori. The free parameters of the MLP, that is, the number of layers as well as the number of nodes per layers are found through cross-validation on the training set. Cross-validation is done for each class separately.

2.6 RESULTS

As a baseline for further evaluations, I tested the 2D classification performance of the presented approach on the Caltech-101 database. Using a standard training setup of 30 training samples, I obtained a recognition rate of 60.1% by relying on the combination of the four 2D features. The performance stays somewhat below results reported in current literature (e.g., [Boiman 08]), which might be due to more extensive parameter tuning and more optimized classifier kernels that these systems employ. Nevertheless, for simple 2D features without any additional shape/configuration modeling of the categories, these performance levels are encouraging—especially as the features can be evaluated in near real-time.

2.6.1 OVERALL CATEGORIZATION PERFORMANCE

For evaluation, the database was randomly split into training and test sets⁴. Six objects per class were used for training, the remaining objects were put into the test set. For each object, 18 views are selected for training and 18 views for testing. The views were equally distributed across the 36 available views, i.e. one view every 20°. The training set consisted of 82 objects with a total of 1476 views. The test set contained 74 objects with 1332 views. The procedure was repeated multiple times with random splits. Features were extracted for each view and classifiers were trained according to Section 2.5. Additionally, I run the same tests on the Lai et al. *RGB-D* [Lai 11] (cf. 2.2) dataset. As mentioned in Section 2.3, many categories only contain few exemplars. Since the number of classes is very high, however, I was interested in seeing how individual classes would perform. I excluded classes with less than 6 objects per class. The remaining 36 categories were evaluated following to the same training and testing procedure as on our dataset.

⁴ For the evaluations, the classes *perforator* and *phone* are excluded due to the low number of exemplars. Furthermore, *forks*, *knives*, and *spoon* are merged into the joint category *silverware*.

Table 3: Classification performance

		Accuracy			
		MPI-IPA		RGB-D	
	Descriptor	Single	Combined	Single	Combined
2D	SURF	42.4%		31.5 %	
	PHOG	69.9%	66.6%	59.6%	54.9%
	Self-Similarity	41.7%		26.5%	
	Color	26.6%		28.6%	
3D	Shape Distributions	25.4%		13.1%	
	Shape Index	34.6%	74.6%	26.1%	68.2%
	Shape Context 3D	55.2%		51.9%	
	Depth Buffer	72.9%		57.7%	
2D + 3D			82.8%		77.3%

Classification results are listed in Table 3 for both datasets. The values represent average classification accuracy normalized by class size. In case only 2D descriptors are used, we obtain 66.6% (54.9% RGB-D) correct categorization, whereas the 3D descriptors yield 74.6% (68.2% RGB-D) correct results. Combining both 2D and 3D descriptors, the performance increases significantly to 82.8% (77.3% RGB-D) correct.

The combination of 2D features does not lead to a significant improvement of the results. On both datasets, the best single descriptor (PHOG) performed even slightly better than all features combined. For 3D there is a small performance gain on our dataset. On the RGB-D dataset there is 9.5% increase over the best single descriptor (Depth Buffer). However, the results improve clearly on both datasets if the two modalities are fused. This is due to the different object characteristics that the features latch on to.

In Table 4 and Table 5 categorization performance is reported for each class individually. Best results are marked in green, second in yellow, and worst in red. The patterns in the table make it easy to recognize that some classes are more sensitive to 2D information (e.g., *silverware* or *scissors*), whereas for other classes 3D cues are more effective (e.g., *drink cartons* or *books*). The worse performance of

Table 4: Results by class on our dataset.

Class	Accuracy (%)		
	2D only	3D only	Combined
Binders	81.9	92.5	94.2
Books	60.9	90.4	89.1
Bottles	36.9	73.9	80.8
Cans	52.2	84.7	62.5
Coffee pots	72.8	68.3	80.6
Cups	76.0	77.5	94.6
Plates	98.1	93.9	97.5
Dish liquids	44.8	47.4	62.2
Computer mouse	88.9	100.0	98.3
Pens	54.2	68.2	75.3
Scissors	80.0	53.3	70.0
Monitor	78.9	84.4	92.2
Silverware	75.6	63.0	80.8
Drink carton	12.2	55.6	81.9

3D cues in some cases could also be attributed to the lower spatial resolution of the depth sensor compared to the high-resolution RGB camera. Thin structures, such as the handle of a scissor, are not always captured reliably. For the majority of classes, however, the combination of both information pathways leads to an increase in performance.

2.6.2 ROC CURVES AND CONFUSION MATRICES

The rate of correct results is only one part of the story. First of all, one might wish to specifically set acceptance thresholds for the different classes, in order to control the number of false alarms versus hits. The ROC curves for the 2D, 3D and combined cases are shown in Figure 16. These results show that the 3D cues are always better than the 2D cues, and that the combined case always provides a clear increase over the single modalities. From these curves, the EER (Equal-Error-Rates), which provide a good indication of the trade-off between false alarms and hits are determined as: 2D (12.0%), 3D (8.7%), Combined (6.2%).

In some cases, it might be interesting to look at the pattern of confusions to determine, for example, the degree of generalizability of the features, or to provide

Table 5: Results by class on RGBD dataset.

Class	Accuracy (%)		
	2D only	3D only	Combined
Apple	40.6	85.6	71.7
Ball	35.0	22.2	61.7
Ball pepper	61.7	71.7	69.4
Bowl	8.3	52.2	51.7
Calculator	18.0	24.3	52.4
Cell phone	58.7	43.3	50.7
Cereal box	27.1	95.5	93.3
Coffee mug	33.1	91.9	79.2
Dry battery	64.7	49.4	65.5
Flashlight	44.4	67.3	66.2
Food bag	19.4	72.5	77.8
Food box	28.3	58.3	68.7
Food can	63.3	83.8	87.5
Food cup	42.8	78.9	86.1
Food jar	22.2	22.8	29.4
Garlic	80.0	65.0	75.6
Glue stick	78.9	92.8	100.0
Hand towel	39.4	82.8	87.8
Instant noodles	62.5	75.8	90.3
Keyboard	88.7	76.5	88.8
Kleenex	95.0	53.1	98.3
Lemon	98.3	60.0	100.0
Marker	44.9	63.6	66.3
Notebook	26.7	92.8	77.2
Onion	77.8	84.4	96.1
Plate	88.3	96.1	97.8
Pliers	68.9	80.0	87.8
Potato	82.2	26.1	82.2
Shampoo	57.8	92.8	78.9
Soda can	73.3	98.3	77.2
Sponge	21.9	71.4	75.6
Stapler	61.9	69.4	75.6
Tomato	65.8	28.9	86.1
Tooth brush	34.4	79.4	57.2
Tooth paste	96.7	78.9	98.3
Water bottle	66.5	66.9	74.3

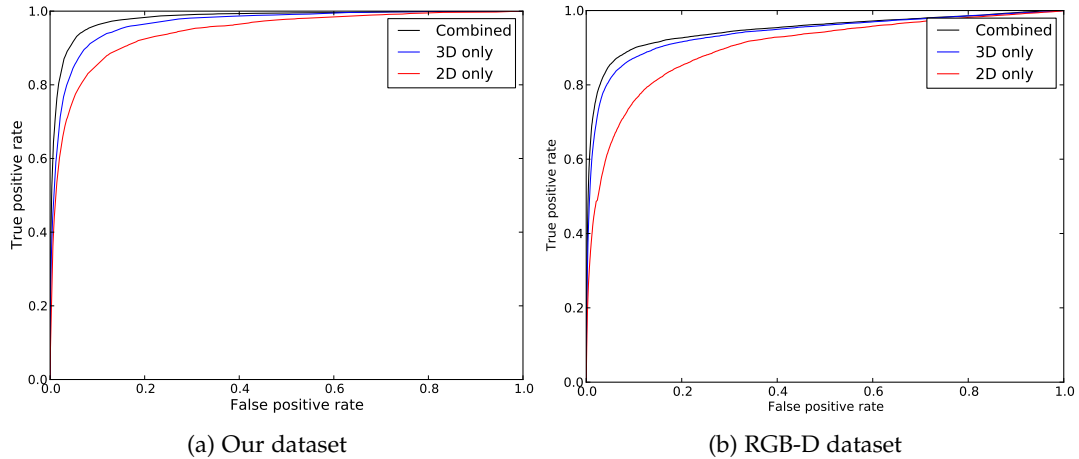


Figure 16: ROC curves for both modalities separately and in the combined case.

a different, more effective clustering of categories. Figure 18 shows the confusion matrices for all features, as well as separately for the 2D and 3D features. On a more fine-grained scale this is also shown in Figure 17. Confusion matrices represent SVM predictions performance for each feature type individually. In the top row you find results for the 2D case, in the bottom row for 3D. Accuracy varies strongly between classes and feature types. This should be seen as an indicator of how important feature fusion strategies are to obtain high overall accuracy. Consistent with the previous categorization results, joint predictions (18) are more consistent if based on 3D features than on 2D features (cf. number of non-zero off-diagonal elements in each matrix). Furthermore, if we examine the categories that are often confused in the combined case (e.g., *dish liquids* and *bottles*) in Figure 18, we see that those contain the classes that also are confused for both 2D and 3D data. In contrast at least one modality is able to clearly distinguish between two classes, the result in the combined case is determined by the more descriptive modality. This does not only suggest that the two modalities capture different class properties but also that they can be combined very effectively.

Finally, the remaining confusions after cue combination include categories that are hard to distinguish not only for a computational system. We observe high confusion rates for *cans* and *drink cartons*, as well as *dish liquids* and *bottles*. Especially if only 2D data is used *drink cartons* are very likely to be confused with *cans*, as in the 2D images both objects appear as rectangular items with varying texture. Even here, however, the addition of 3D features (which will add the curvature of cans) reduces the number of confusions substantially. As an additional observation, frequently confused categories might rather be distinguished by their functionality than by their appearance/shape. Although the algorithm is able to generalize from

the current exemplars in the dataset, it can be expected that the inherent category structure will be better captured as the number of exemplars increases.

In summary, employing 3D descriptors gives better results than 2D descriptors. However, by combining both cues an even better overall performance is obtained—with significant increases for many individual object classes.

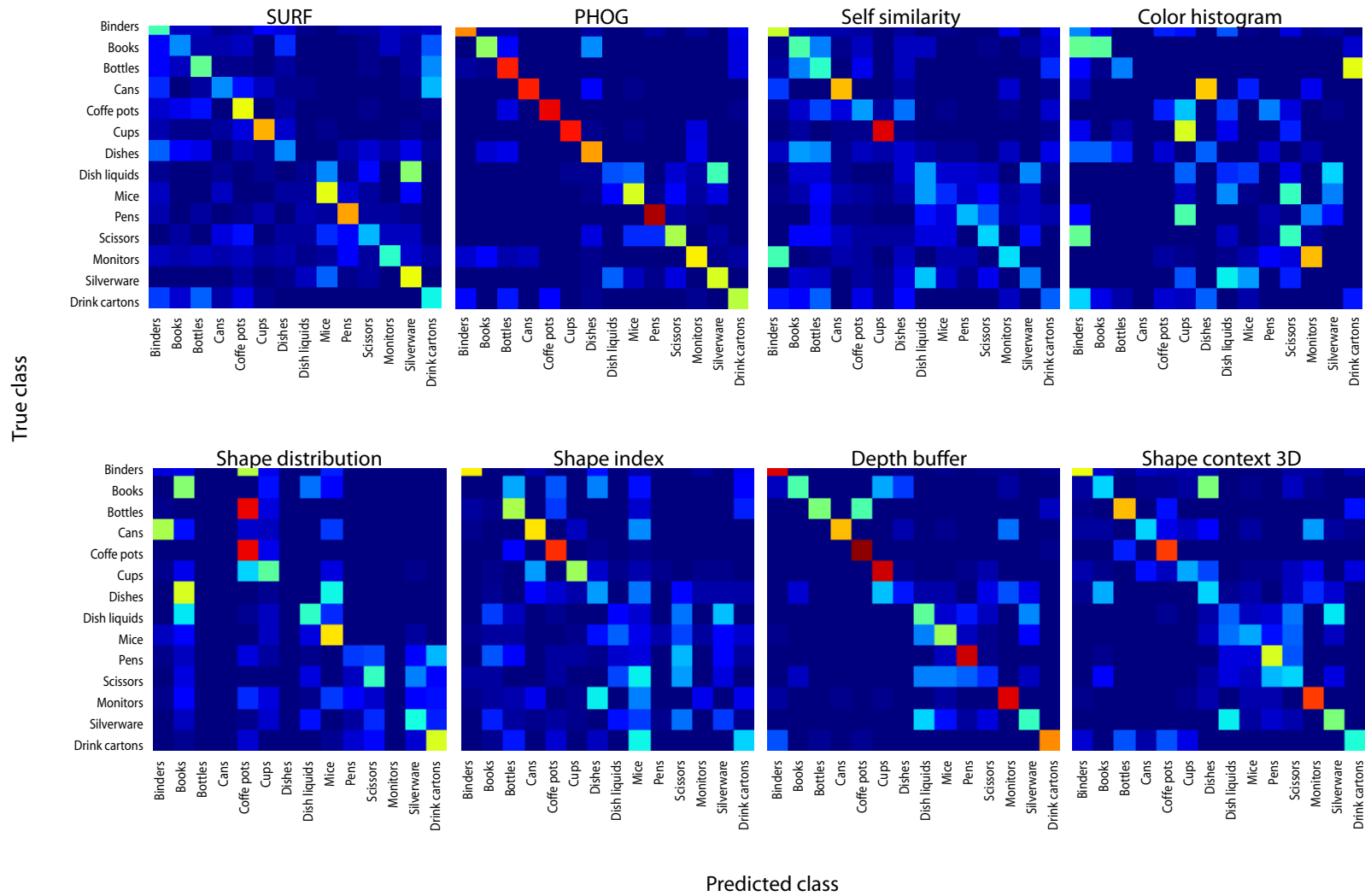


Figure 17: Output of SVM classification.

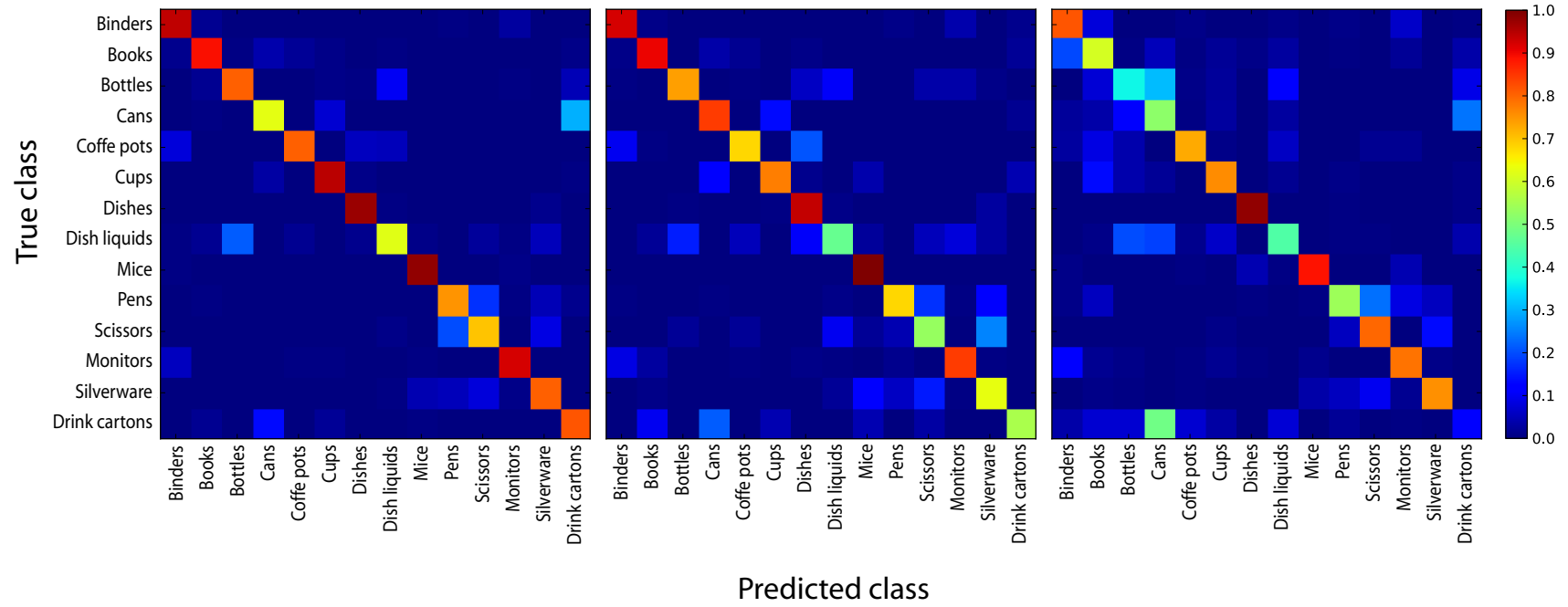


Figure 18: Confusion matrices: Combined cues (left), 3D only (middle), 2D only (right)

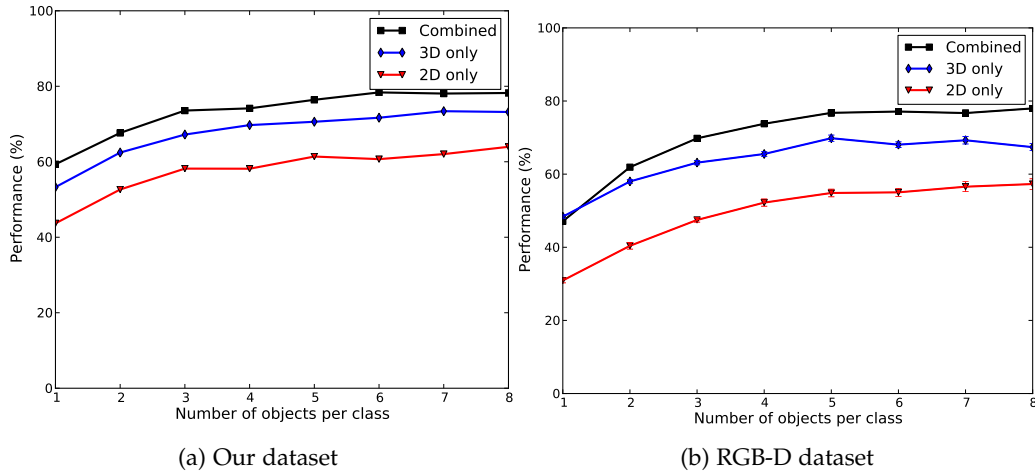


Figure 19: Classification results for increasing number of training objects per class. 18 views per object.

2.6.3 GENERALIZABILITY ACROSS VIEWS AND NUMBER OF OBJECTS

The classification results with respect to the number of objects used for training are plotted in Figure 19. The number of views per object was kept constant. As before, a view was selected every 20° , resulting in 18 views for each object. Evaluation was carried out on the remaining objects in the dataset. For small classes at least one object was retained in the test set. As a result, for the class *scissors* the number of training objects is limited to 4. The evaluation for each object count was repeated 30 times with random splits into training and test set. It is not surprising to see the performance rise as the number of objects is increased—again, one cannot expect generalization of such variable categories to happen from only one exemplar.

Figure 20 shows the classification performance with respect to the number of views per object. The evaluation procedure remained the same with the difference of selecting 6 training objects for each category and altering the number of views per object. We see that in contrast to object count the number of views seems to play a minor role, as long as a minimum amount of object orientations is covered. With 3-4 views—for 2D and 3D data combined—near optimal performance is achieved. Since views are equidistantly distributed, 3 views results in one view every 120° . Our data suggests that already a small number of views can be sufficient to obtain an adequate sampling of the object surface. In contrast, when using only 2D or 3D information, more views are needed to reach peak performance. This seems reasonable as less information is encoded in each view.

If we again look at the results for single classes, we see that the benefit of additional views depends strongly on the shape characteristics of the respective class.

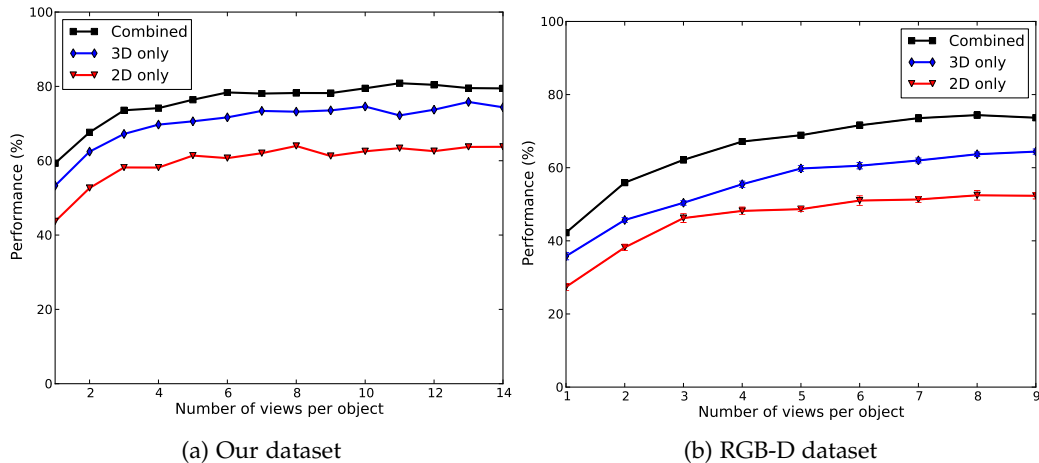


Figure 20: Classification rates for increasing number of training views per objects. Six objects per class.

The comparison of Figure 21 and Figure 22 shows that classes of asymmetrical objects such as *binders* or *drink cartons* exhibit a much steeper increase in classification accuracy than classes of more round and symmetrically shaped objects such as *cups* and *bottles*. Interestingly, in the case of *drink cartons*, this increase is only visible in the 3D domain and the combined case—this is due to the fact, that the drink cartons as a category contain very different labels and therefore 2D appearance measures will have problems with a clear identification of the overall category, whereas the 3D shape is much better defined for this category.

2.6.4 REAL-WORLD EXPERIMENTS

The classifier was trained on all objects in the dataset and a recognition experiment conducted in a test scenario. Scenes containing unknown objects—not contained in the dataset—were recorded. The task was to find objects in the scene and assign them to one of the available classes. Since the focus is not on scene segmentation, basic techniques were employed to separate objects from the background and to find object candidates for classification. Segmentation benefits greatly from the ability to exploit 3D data as planes (such as the supporting table) can be found easily through well-known algorithms such as the Hough transform [Duda 72] or RANSAC [Fischler 81]. Additionally, bounding boxes can be specified in terms of absolute 3D coordinates in order to focus processing only on a certain region of interest in the world (in our case 100x120 cm). As a last step mean shift clustering of the remaining points in Euclidean space was performed with the resulting modes being regarded as object candidates for testing.

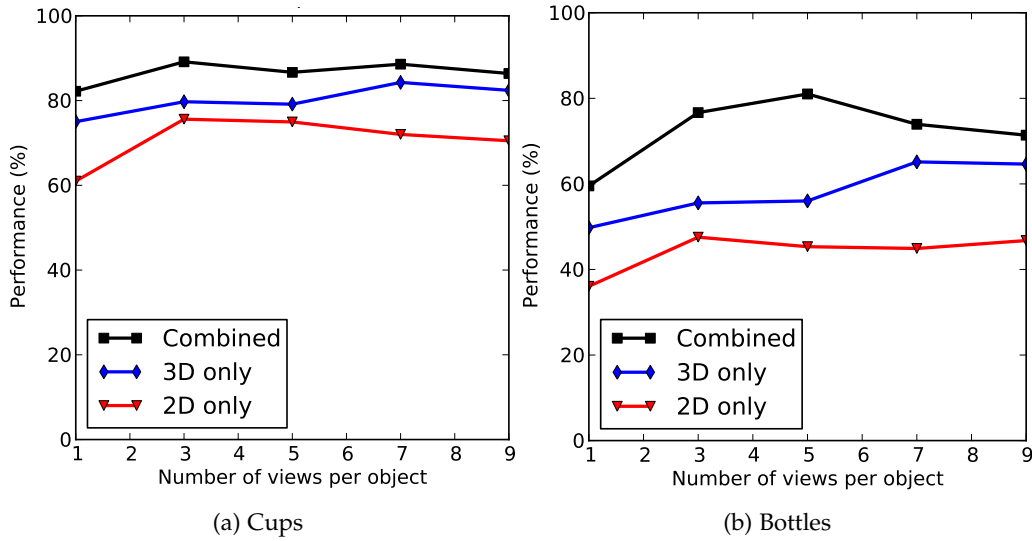


Figure 21: Classification rates for two symmetrical objects. Symmetrical objects show low sensitivity to view count.

Three scenes with images of recognized objects are shown in Figure 23. Values in parentheses denote class confidence ranging from -1 to $+1$. Most objects are recognized at various rotation angles (see examples of cups and of the binder). However, if the object is oriented so that the 3D shape is very similar to multiple categories, the predicted class needs to be determined by the 2D appearance of the object. This can sometimes lead to wrong classifications as in the case of the milk carton (rightmost figure), which gets misclassified as a "book" due to the strong similarity in shape of the two categories for this viewpoint. Although these initial results are encouraging, further studies must be carried out to quantify the recognition performance and test the robustness in more cluttered environments.

2.7 CONCLUSION

I introduced a new dataset for joint 2D and 3D object categorization containing data of real-world objects. The dataset was designed to allow for good generalization to real-world applications. Using this dataset, I demonstrated that the incorporation of range data is highly beneficial for object categorization tasks. The fact that combination of multiple sensory inputs leads to increased recognition performance is not surprising—however, the performance gains are sometimes substantial.

Multimodal object representations that integrate cues beyond visual information might offer even more potential to capture the variety of features defining the ob-

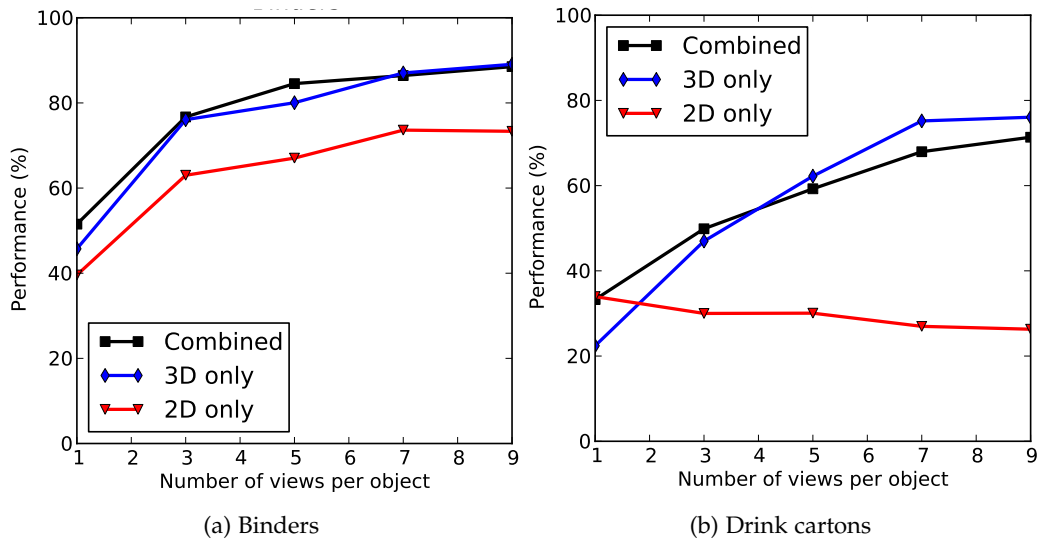


Figure 22: Classification rates for two asymmetrical objects. Asymmetrical objects benefit from additional views.

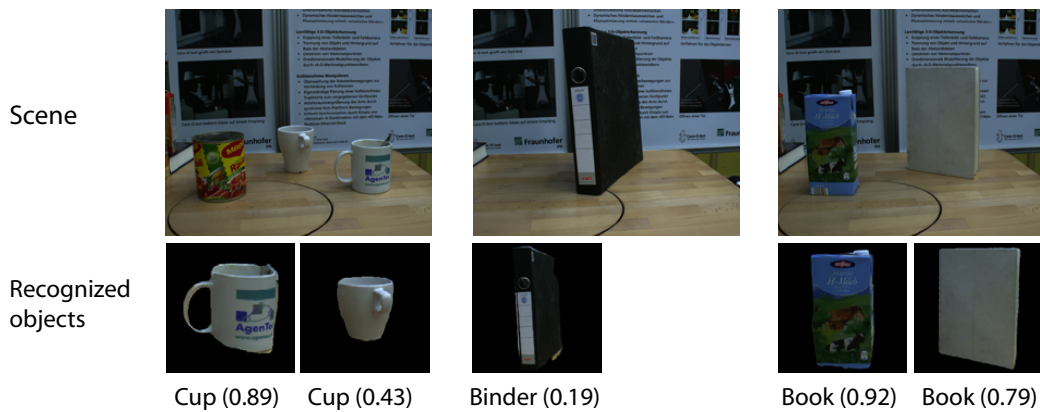


Figure 23: Recognized object classes in three exemplary scenes—the objects were not part of the dataset.

jects in our environment. How to acquire and employ such rich representations has to be studied in future work. I believe that multisensory 3D object classification is a research topic that will gain even more importance in the future as range sensors have become a consumer-level product. I hope that by supplying the research community with this dataset and by demonstrating the merits of sensor combination, more work in this domain will follow.

ACTIVE IN-HAND OBJECT RECOGNITION

*When it is obvious that the goals cannot be reached,
don't adjust the goals, adjust the action steps.*

— Confucius

We define and interact with the world by subdividing it into objects that serve a specific purpose for us. A process that comes natural to us, yet poses a difficult problem for any computer vision system. One difficulty of this is the inherent ambiguity that arises through the fact that a three-dimensional object can be seen from an infinite number of viewpoints. Different objects may thus lead to the same or similar two-dimensional views. The ability to grasp objects and explore them using various senses (vision, touch, smell, etc.) enables us to obtain the information necessary to recognize and make use of the respective item. I believe that such exploratory skills are a key asset for cognitive robots, therefore propose active, multimodal methods for object recognition. This chapter details how perception and action can be coupled to create a closed-loop exploration and recognition scheme on the humanoid robot iCub. Objects are placed in the robot hand and the robot then executes exploratory actions to discover new information and evaluate previous hypotheses.

This approach shares the philosophy of the active vision paradigm. It has been shown [Aloimonos 88, Bajcsy 88, Ballard 91] that by enabling an observer to actively control the sensory input, many vision ambiguities can be resolved. This fundamental idea soon was adopted by the robotics community and has led to a number of systems that implement active object exploration and recognition methods. In the following, studies focusing on active vision for view selection and/or object recognition are discussed.

Wilkes et al. [Wilkes 94], for example, drive the camera to new locations that are expected to yield informative object views. Dickinson et al. [Dickinson 97] consider objects as compositions of basic volumetric parts. The camera is moved in order to

disambiguate between aspects that share a similar visual appearance using a pre-computed aspect prediction graph. Based on the view-based object representation of Murase and Nayar [Murase 95], Paletta et al. [Paletta 00] propose a recognition system in which an object is placed statically on a turn table and the next view-point for the camera is computed so that the expected entropy loss is maximized. Callari and Ferri [Callari 01] fit shape primitives to range scans and infer object probabilities using neural networks. A sensor, mounted on the end-effector of a robot, is then moved around the viewsphere of the object to minimize the entropy of the object probability distribution. Denzler and Brown [Denzler 02] use a maximum mutual information criterion to select camera parameters that allow to zoom in on an informative location in a static scene.

Omrčen et al. [Omrčen 07] address the basic sensorimotor processes that have to be provided to allow a dexterous exploration of an unknown object. In [Welke 09] and especially [Welke 08] active methods have been studied that are comparable to my approach in the sense that objects are inspected from multiple viewpoints to resolve ambiguities and to distinguish between objects. However, these systems lack the direct interaction with the object and are therefore not able to take into account the additional cues active exploration on a humanoid robot offers.

I also distinguish my work from that of for example [Krainin 11] or [Welke 10] whose focus is on exploration and building object models instead of discrimination as presented here. Furthermore, there are approaches that do not interact with the object directly, but rely on robot locomotion for object detection [Saidi 07, Andreopoulos 11] or modeling [Foissotte 08].

The proposed recognition system can be split into two fundamental parts. One part is the perception side which comprises components for data acquisition and reasoning, and the other part is the action side that consists of the robot hardware and controllers. Perception and action are directly linked together resulting in one being determined by the outcome of the other. This principle is often referred to as Perception-Action-Loop and is visualized in Figure 24 [Siciliano 08]. In the following, I discuss some of the components that need to be provided in order to close this loop. I start with presenting the modules that are used in all following implementations. The first and certainly most salient one is the robot platform that was used (Section 3.1.1). Next are standard computer vision modules that to pre-process the visual input data. I will describe segmentation and feature extraction in Section 3.1.2. The from our perspective most relevant and interesting subparts are object recognition and motion control. I will illustrate two possible ways of tackling these challenges in Section 3.2 and 3.3. There I describe two approaches to actively

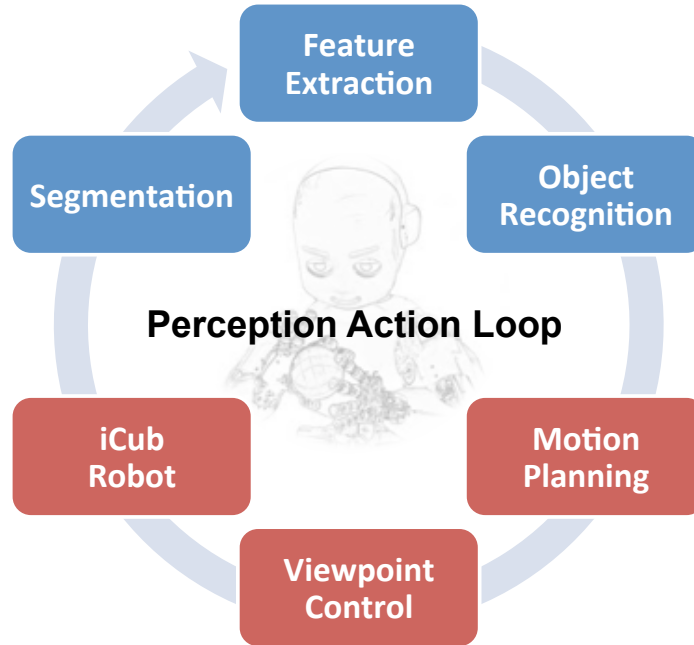


Figure 24: System components forming a perception action loop.

control the recognition process. In Section 3.2 I present an intuitive and simple way to make use of the ability to manipulate an unknown object and produce new views to be incorporated into the current hypotheses. Section 3.3 addresses some severe shortcoming of the simple approach by regarding recognition and motion planning in a probabilistic fashion avoiding hard decisions.

3.1 EXPERIMENTAL SETUP

The platform used for implementations and experiments is the iCub robot (see Section 3.1.1 and Figure 25), developed by the Italian Institute of Technology (IIT) in Genova, Italy. In setup for this study, an object is placed in the hand of the iCub which rotates the object to produce new views during the recognition process. The object manipulation sequence is not predefined and will be different for every object. At all time, the robot keeps the gaze directly on the object. The position of the palm is calculated using forward kinematics and fed as target position into the iCub gaze control module [Pattacini 10a]. This module, part of the iCub software stack, controls neck and eye motors to focus on the supplied target. The following implementations use the right robot arm and the left camera/eye. Thus, the vision system is monoscopic. However, disparity cues from stereo vision are incorporated in the image segmentation process presented in Section 3.1.2.

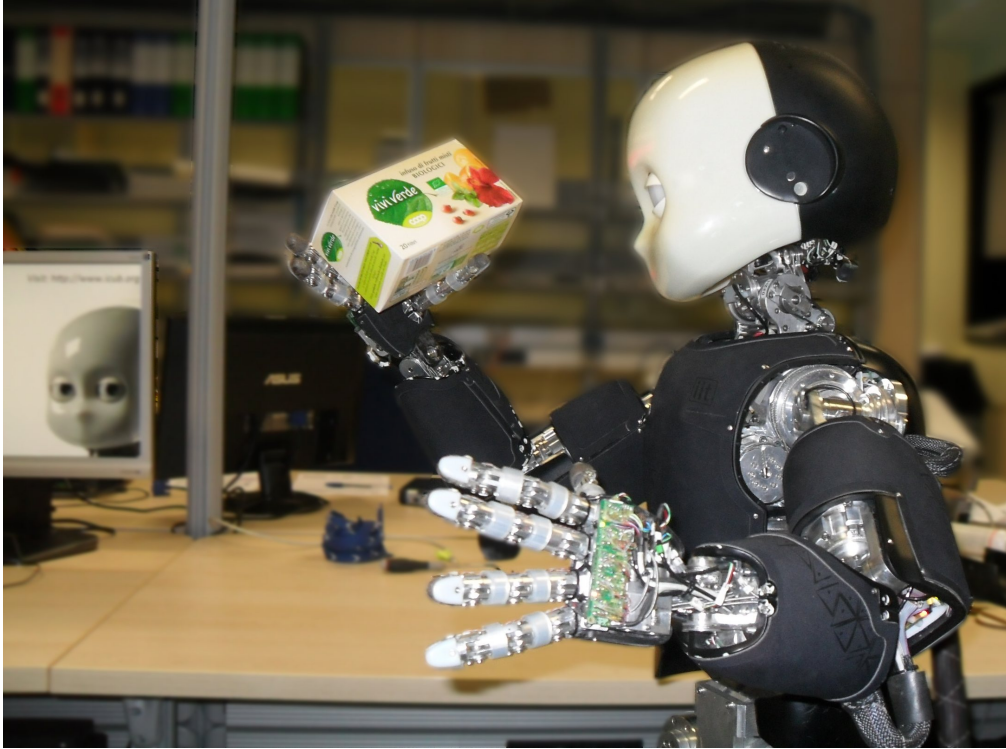


Figure 25: The iCub humanoid robot. Implementation and evaluation platform of the presented object recognition method.

3.1.1 THE ICUB ROBOT PLATFORM

The iCub (Figure 25) is an open-source humanoid robot designed as a result of the “RobotCub” project, a collaborative European project aiming at developing a new open-source cognitive robotics platform. Measuring 105cm in total height, the iCub robot is approximately the same size as a three year old child. The iCub is the ideal platform to undertake research in cognitive systems [Broz 09, Sandini 07, Tikhanoff 11], as it has fully articulated hands, which allow for dexterous manipulations, as well as a head-and-eye system, which permits very precise and accurate movements that are required for vision. Furthermore, the iCub robotic platform is equipped with visual, vestibular (for balance and spatial orientation), auditory, and haptic sensor capabilities. The iCub humanoid robot is fully articulated and many of the parts were designed from the ground up, such as the head, torso, two arms and hands, and two legs.

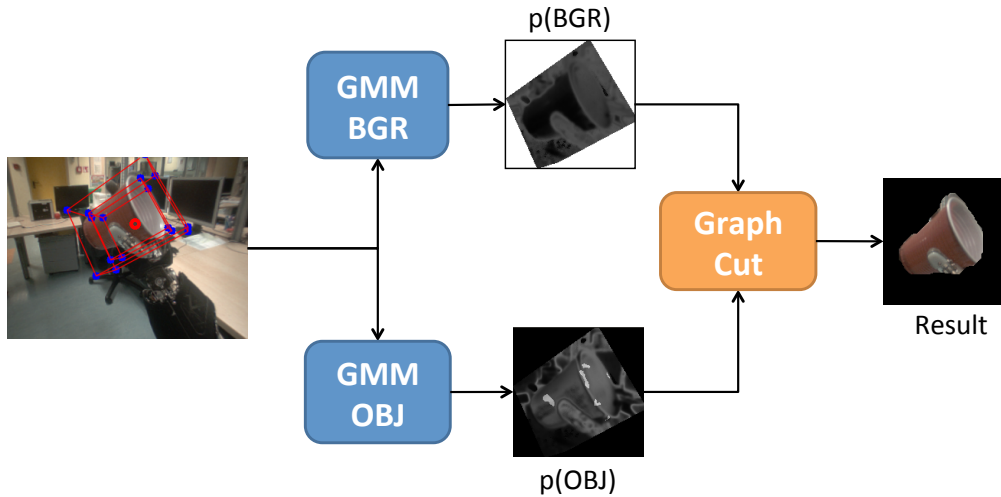


Figure 26: Segmentation process: A background model is trained on the area between the rectangles and applied to the area inside the smaller rectangle. Low values in the resulting probability map indicate the presence of an object.

3.1.2 IMAGE SEGMENTATION AND FEATURE EXTRACTION

The object in the hand of the robot occupies only a small area of the image obtained from the robot camera. Using the whole image for recognition would result in a poor recognition performance, as the scene background would introduce a high amount of noise. One advantage of the hardware setup is that we can exploit kinematics of the robot to calculate the Cartesian position of the hand. Adding an offset to this point gives us a good starting point for where the object would be localized. Since intrinsic and extrinsic camera parameters are known it is possible to compute the U/V-coordinate in the image based on a 3D Cartesian point. We now assume that object sizes do not exceed certain limits and specify a 3D bounding box around the previously calculated object center. After projecting this bounding box onto the 2D image area, the image can be cropped to a rectangular sub-image containing the 2D bounding box.

To obtain a more accurate object contour, I implemented a fast foreground/background segmentation algorithm. For this, two Gaussian Mixture Models (GMM) (see e.g., [Bilmes 98]) are created per image, one for the background and one for the object; in contrast to many similar approaches, which create models for each image location. GMMs are commonly used for background removal tasks. They are specified by K normal distributions $\mathcal{N}_k(\mu, \Sigma)$ with mean μ and covariance Σ as well as a weight w_k . Since the camera is moving, we need to be able to deal with a quickly changing background. Therefore, models for each incoming image are created and directly applied to the same image.

The area outside the bounding box is used as training area for the background model. Another bounding box, similar to the first one, is created for the object model. This bounding box, however, is smaller and supposed to contain little or no background. In Figure 26 the bounding boxes are depicted in the left image.

Pixel intensities in CIE $L^*a^*b^*$ color space are input samples for calculating $\mathcal{N}_{1,\dots,K}(\mu, \Sigma)$. The optimization is carried out using Expectation-Maximization [Dempster 77] (relying on the C++ implementation from the OpenCV library [Bradski 00]). As number of pixels is low, training is completed within approximately 50ms on a dual-core 2.5 GHz mobile CPU. The trained GMM is then applied to the subimage containing the object. The probability P_{BGR} of an image pixel I being considered as background is computed by evaluating the weighted PDF of the multivariate distribution defined by:

$$P_{\text{BGR}}(I) = \sum_{k=1}^K \frac{w_k}{\sqrt{|2\pi\Sigma_k|}} \cdot \exp^{\frac{1}{2}(I-\mu)^T \Sigma_k^{-1} (I-\mu)} \quad (4)$$

The resulting probability maps are then combined using a graph cut. Each pixel in the probability maps is treated as a node in a graph. Neighboring pixels are connected via edges (8 connected) and each node is assigned the object and background probability as source and sink capacities. The maxflow algorithm (implementation of [Boykov 04]) is used to compute the graph cut. The result is a binary labeling of each image pixel. Since we are working with a binocular system, we can exploit disparity cues to improve the segmentation result by increasing the probability value of pixels with high disparity (i.e., close to the camera) in the object probability map before executing the graph cut. In Figure 26 this is visible in form of small bright patches in the object probability map.

The image content is encoded by feature vectors extracted from the segmented object images. Color histograms in CIE $L^*a^*b^*$ space (100 bins for a^* and b^*) represent a basic appearance descriptor. I intentionally refrained from using more advanced features for two reasons: first, we need an image descriptor that is insensitive to inaccuracies in the segmentation process. Although the segmentation process most of the time provides a good object image, it sometimes outputs an insufficiently segmented image. Second, the purpose of the presented system is in demonstrating a generic perception-action driven framework that can improve recognition results for any feature type—more sophisticated descriptors such as PHOG [Bosch 07] (cf. Section 2.4.1.2) or SIFT [Lowe 99] (cf.) may be more specific, but are both expensive to compute and may only work in limited cases of textured objects.

3.2 THE VIEW-TRANSITION-MAP

This section describes a first attempt to combine visual and proprioceptive information. It features so-called View-Transition-Maps (VTM) [Wallraven 07] in order to control the robot arm during the recognition process of an object. First implementations were run on the BabyBot robot at the University of Genoa, Italy [Wallraven 07]. I reimplemented and extensively studied this method on the SL robot simulator [Schaal 01]. The robot simulated here was a Barrett WAM robot arm. The iCub in contrast is an anthropomorphic humanoid robot that is fully articulated. I integrated the VTM setup on the iCub and conducted evaluations to study its ability to distinguish very similar office and household objects.

VTMs consist of keyframes that are annotated with the joint angles configuration received from the motor sensors of the robot. Keyframes or keyviews are snapshots of the camera images obtained from the robot camera that bear high visual significance (cf. Section 3.2.1). Keyframes are recorded by having the robot move an object and keeping a steady gaze onto the object (cf. Section 3.1). The object is placed statically in the hand of the robot and a predefined motor program is executed. This program consists of changing two of the wrist joints. One leads to an up/down motion the other one to a left/right rotation. As this exploratory movement is executed, the camera input stream is segmented into keyframes.

3.2.1 KEYFRAME SEGMENTATION

Recording all images without filtering would result in a vast amount of data that is intractable, both in terms of memory consumption and computational effort. However, keeping too few views would lead to a low recognition performance. To select only a representative subset of views from the continuous stream of input images, only certain keyframes [Wallraven 07] are retained based on the amount of visual change occurring. This is measured by converting the image to a lower dimensional feature representation and computing the feature distance between subsequent images. This representation should be insensitive to clutter and artifacts introduced through preprocessing steps such as segmentation. Good results are obtained by converting the images to CIE $L^*a^*b^*$ color space and selecting high energy frequencies of the Fourier-transformed color planes. A frame is marked as keyframe if the feature distance to the last keyframe drops below a predefined threshold. Each keyframe is annotated with the robot joint configuration it was extracted on. A few dozens of these keyframes are usually sufficient to represent the

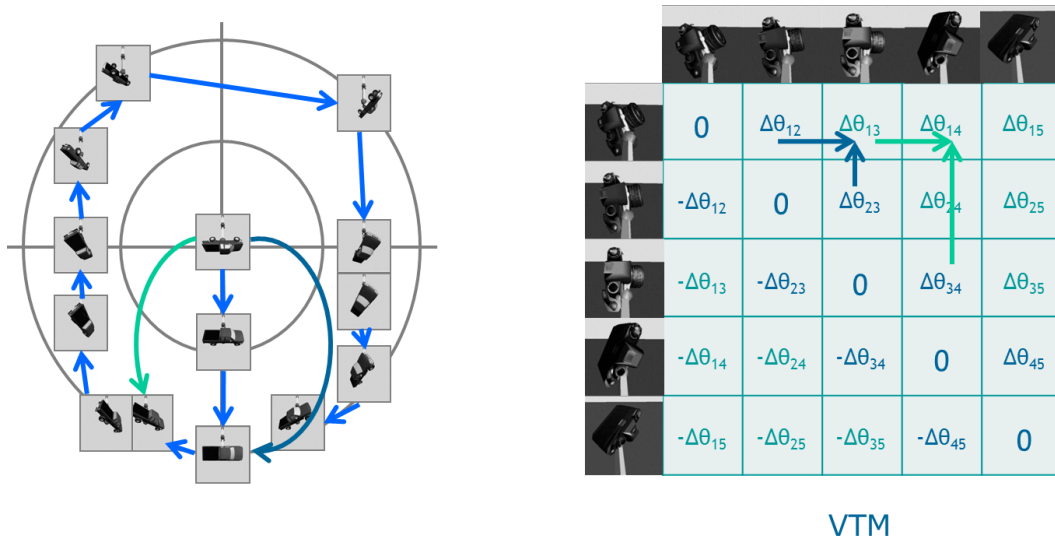


Figure 27: Left: Joint space diagram. Recorded keyframes are placed in two-dimensional joint space. Each circle represents 45° of object rotation. Right: View-Transition-Map (VTM). Cells contain joint differences. Blue values (above main diagonal) are recorded, all remaining values derived from these. Arrow colors correspond to colors in joint space diagram.

appearance of an object. In Figure 27 keyframes are placed in a two dimensional joint space diagram. The dimensions correspond to the two wrist joints that were controlled during the exploration run.

3.2.2 BUILDING THE VTM

The VTM is created according to the following procedure (also illustrated in Figure 27):

1. The first keyframe is annotated with joint state $s_0 = (\theta_1, \dots, \theta_n)^\top$, where θ_i denotes the configuration of joint i .
2. For the next keyframe the transition $t_{i-1,i}$ is calculated as,

$$t_{i-1,i} = s_i - s_{i-1} = \begin{pmatrix} \theta_{i,1} \\ \vdots \\ \theta_{i,n} \end{pmatrix} - \begin{pmatrix} \theta_{i-1,1} \\ \vdots \\ \theta_{i-1,n} \end{pmatrix} = \begin{pmatrix} \delta\theta_{i,1} \\ \vdots \\ \delta\theta_{i,n} \end{pmatrix}. \quad (5)$$

Transition $t_{i-1,i}$ is stored at $VTM(i-1, i)$.

3. Transitions to all previous keyframes are computed. The transition sequence $s_1 \xrightarrow{t_{1,2}} s_2 \xrightarrow{t_{2,3}} s_3$ can be replaced by $s_1 \xrightarrow{t_{1,2}+t_{2,3}} s_3$. Consequently, remaining

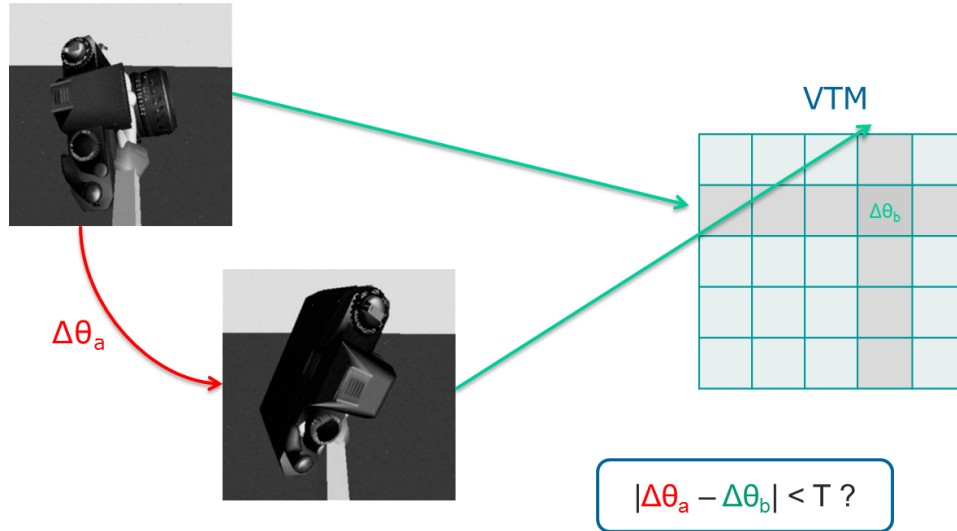


Figure 28: VTM lookup. The transition associated with the respective top matching keyframes is compared to the currently executed transition.

elements can be filled by iteratively merging previous single motions into new joint motions:

$$\text{VTM}(k, l) = \text{VTM}(k, l-1) + \text{VTM}(k+1, l) \quad (6)$$

The results of this process is a triangular matrix that holds the motor command to move the robot from joint configuration of keyframe i to keyframe j . For $j > i$ we take $\text{VTM}(j, i)$ instead of $\text{VTM}(i, j)$ and invert the transition found at this position. This means that $t_{i,j} = -t_{j,i}, \forall i, j$. Since there is no motion between identical keyframes the main diagonal of this matrix is always empty.

3.2.3 USING THE VTM TO CONTROL THE ROBOT

The VTM can be used to control the recognition process when exploring an object. The entries in the VTM correspond to the motion that has to be executed to bring an object from one pose to another one. Assuming the robot holds the object in a particular pose and observes a two-dimensional view of this object, the VTM can be queried for a matching keyframe. If such a match is found, the current object hypothesis can be verified by executing a transition stored for this keyframe and checking if the resulting view corresponds with the linked target keyframe in the VTM. This process allows to incorporate proprioceptive information into the recognition process since the result of the recognition does not only depend on visual matches but also takes into account the relation (i.e., differences in joint configuration) between views. For example, an object matched with a keyframe of

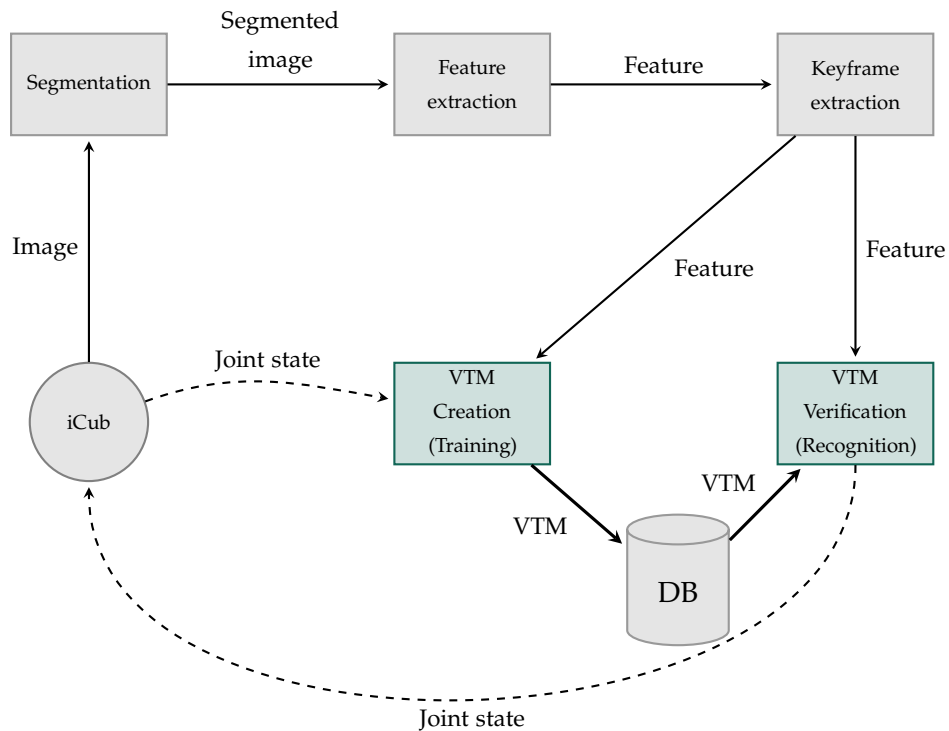


Figure 29: Schematic overview of the object recognition system using View-Transition-Maps.

a front view of the object has to match the side view if rotated by 90° . This VTM lookup is depicted in Figure 28.

3.2.4 IMPLEMENTATION OVERVIEW

The described method was integrated on the iCub. In Figure 29 an implementation overview is shown. As you can see, the robot is controlled in a closed loop. As mentioned in Section 3.1.2, the robot gaze controller tracks the object in the hand and camera images are segmented to remove background and hand/arm. Feature vectors are extracted and fed into the keyframe segmentation module discussed in Section 3.2.1. During object learning (Section 3.2.2) incoming keyframes are used to build a new VTM representation. When the object is explored and the VTM created, it is stored on disk in an object database. To recognize an unknown object, new keyframes are matched with the ones stored in the object database. Two keyframes

KF_i and KF_j are compared by computing the Euclidean distance between their color histogram feature vectors f_i and f_j :

$$d(KF_i, KF_j) = \|f_i - f_j\| = \sqrt{\sum_{k=1}^n (f_{i,k} - f_{j,k})^2} \quad (7)$$

Suppose the robot recorded keyframe KF_c . The VTM is queried with KF_c . It returns a three-tuple consisting of a matched keyframe KF'_c , an expected target keyframe KF'_t and a transition $T_{c,t}$.

$$KF'_c, KF'_t, T_{c,t} = \text{VTM}(KF_c). \quad (8)$$

Now, the motor command defined by transition $t_{c,c'}$ is executed. Assuming we are currently in joint state S , the target configuration S' is given by:

$$S' = S + T_{c,c'} = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} + \begin{pmatrix} \delta\theta_1 \\ \vdots \\ \delta\theta_n \end{pmatrix}. \quad (9)$$

On completion, the new object view KF_t is compared to the expected target KF'_t . KF'_t is the view that is stored in the VTM associated to the previously matched keyframe and the executed transition. A total matching score V is computed for the executed action:

$$V = \exp^{-\alpha d(KF_i, KF_j) d(KF'_i, KF'_j)} \quad (10)$$

Resulting values are scaled to a reasonable domain using α as a normalization constant.

3.2.4.1 Recognition process

The following subsection details the recognition process that was executed during the recognition experiments using the VTM approach. It is schematically illustrated in Figure 30.

The process starts with having the iCub execute the same exploration sequence as when learning an new object. Again, features (cf. 3.1.2) and keyframes (cf. 3.2.1) are extracted. The object database is now searched for a candidate object that passes a keyframe matching threshold. If no candidate object can be found, the exploration sequence continues until the next keyframe is extracted. If there is a good candidate, a transition is selected from the VTM. This is done randomly but it is possible to impose constraints such as requiring a minimum transition length. In any case, the selected transition has to lead to a valid joint configuration,

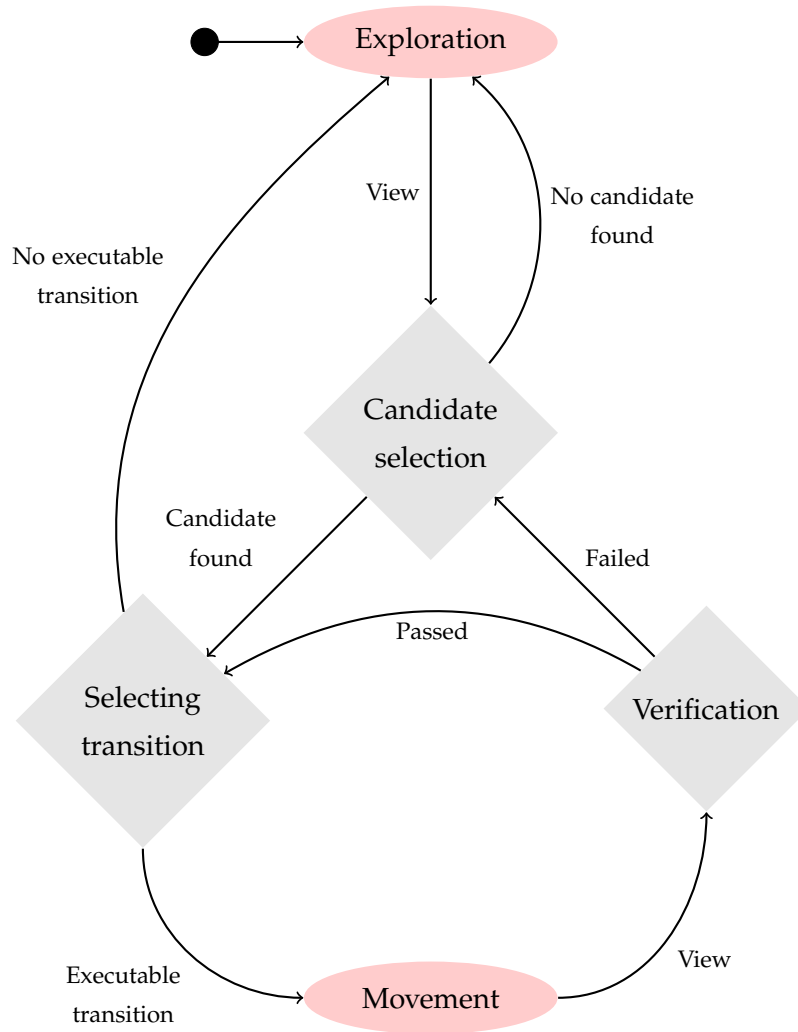


Figure 30: Recognition process.

i.e., resulting in a configuration that does not exceed valid joint limits. In case no such transition can be found for the current robot pose, the iCub will once again continue to follow the predefined exploration sequence.

Assuming we have found an executable transition, the robot now turns the object according to this transition. The resulting object view is now compared to the expected keyframe. If the visual difference is below a predefined threshold, the process continues with the selection of a new transition starting from the now achieved keyframe. If the verification fails, we look for another object that can be matched with the current view.

This process continues until a certain number of such verification trials are completed. For the following experiments, matching scores were recorded for 15 successful verification runs.

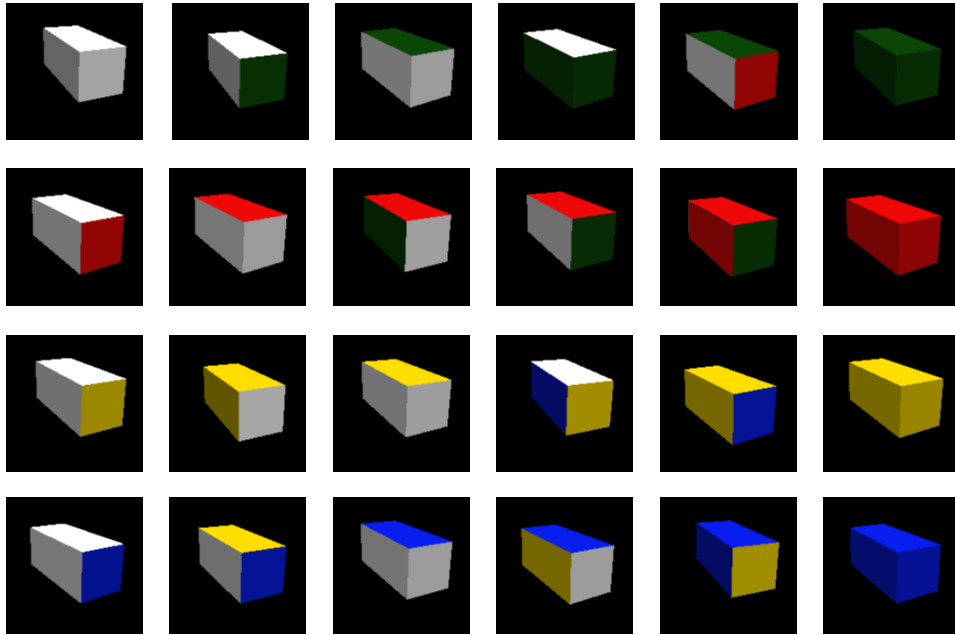


Figure 31: Objects used for evaluation in the iCub simulator. 24 rectangular boxes with differently colored sides. Shown are segmented keyframes used in the recognition process.

3.2.5 EVALUATION ON THE ICUB SIMULATOR

To test the validity of the approach under controlled conditions, recognition experiments were conducted on the iCub simulator [Tikhanoff 08]. 24 rectangular boxes (Figure 31) were created using common 3D modeling tools and assigned different colors to three of the sides. Thus, all objects contain sides that are identical to sides of other objects. This means that single views can easily be confused. As the objects, however, are distinguishable from other viewpoints, we expect that after some exploration time enough information is gathered to correctly identify most of them.

A confusion matrix visualizing the results is shown in Figure 32. Most objects are recognized reliably which is indicated by red colors on the main diagonal. The overall recognition accuracy across all objects and trials ranges at 73.0%. If we look closer, we see that especially single color objects (*box_yellow*, *box_red*, *box_blue*) have high accuracy. These objects are easy to recognize since by only relying on color information the viewpoint is irrelevant in these cases. The reason why the completely white box (*box_w*) is confused with the same incorrect object is that feature vectors for entirely white views are identical for all objects. The result in this situation is undefined. In this implementation the first object in the database is returned.

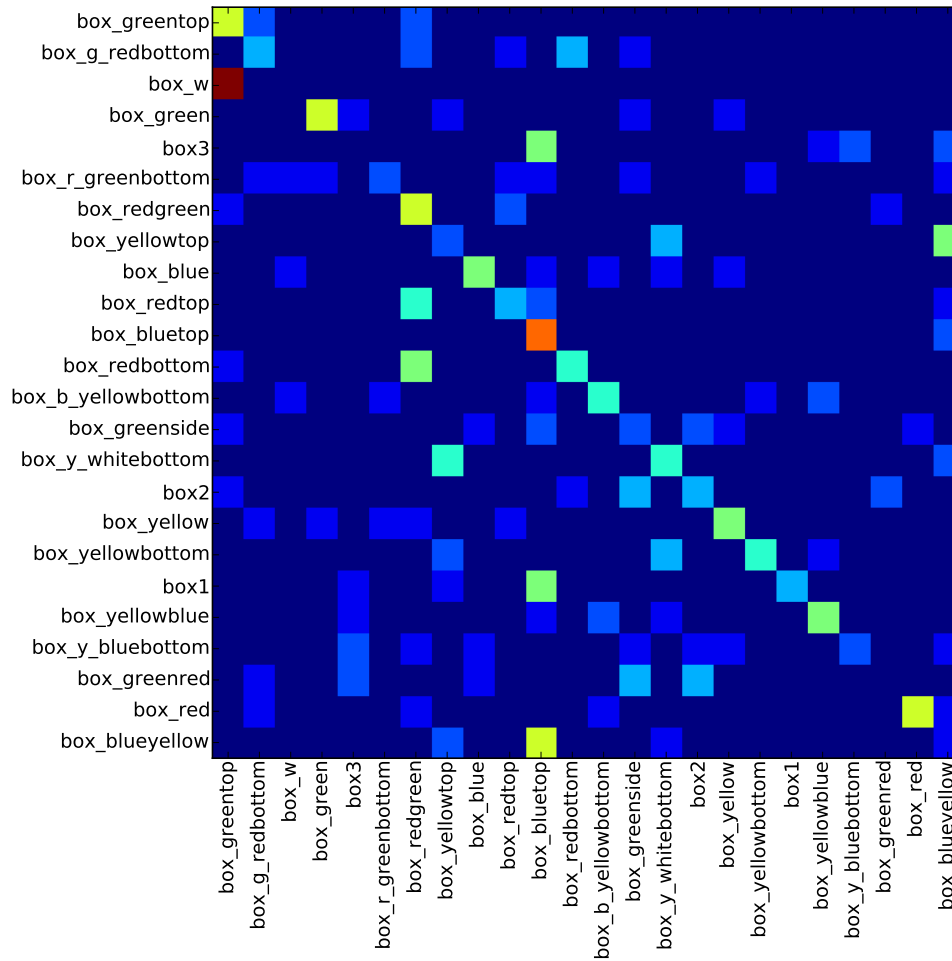


Figure 33: Confusion matrix for VTM recognition in simulation. Objects rotated by 30 degrees about ϕ .

Results shown so far are valid for the case when the object to be recognized is placed in the robot hand in exactly the same way as during learning. This, of course, is the optimal condition and in reality rarely the case. To test the robustness in more realistic situations the boxes were rotated by increasing values about the ϕ -axis in both directions. The resulting recognition performances are plotted in Figure 34. We see that the performance drops quickly as the object is not placed in the learned configuration. The differences in the results between positive and negative rotations are based on the specific object exploration path as well as on limits in the motor space. During recognition the robot can only achieve a subset of the learned configurations. To compare against the ideal case of no object rotation, Figure 32 shows the same confusion matrix with a pose change of $+30^\circ$ in Figure 33. There are no objects left that are still recognized reliably. The total accuracy has decreased to 35.8%.

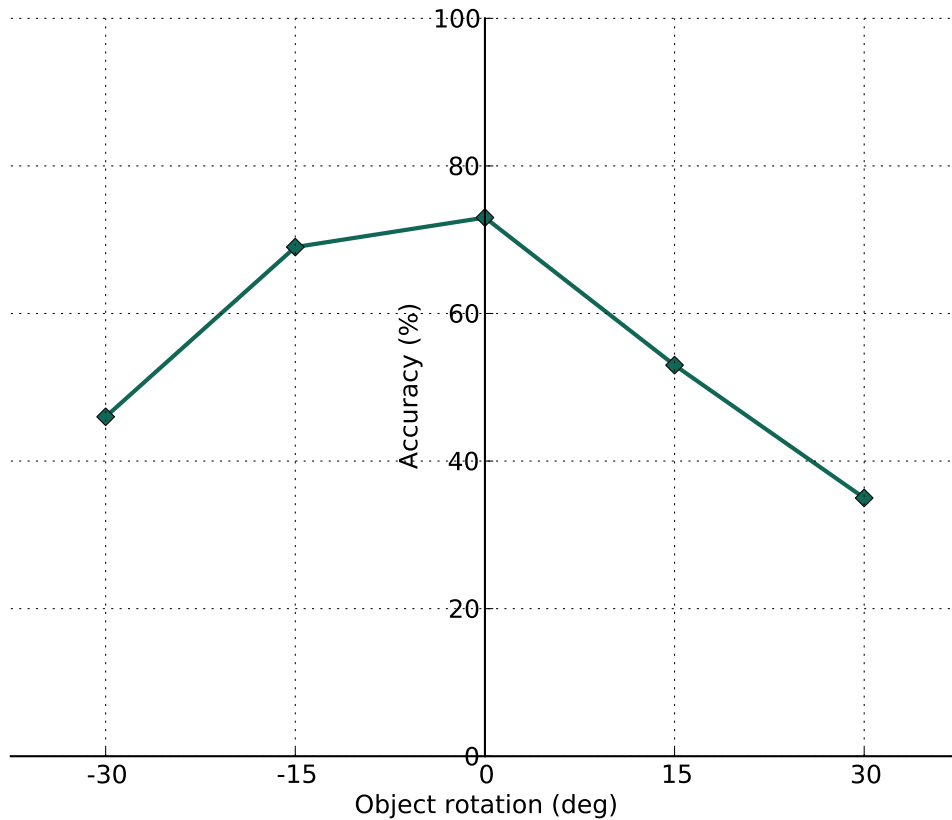


Figure 34: Recognition performance for different amounts of object rotation.

3.2.6 EVALUATION ON THE ICUB ROBOT

Two experiments with different sets of objects were conducted on the real iCub. For the first experiment, four empty tea boxes from the same brand were collected. These boxes look similar but they bear different illustrations of the type of tea they contain. Furthermore, two boxes were rotated so that a different the small side was visible (cf. Figure 35a). This way each box provided at least two meaningful viewpoints. After learning, the boxes were placed in the hand of the iCub rotated by 90° around the longest axis. Recognition results are shown in Figure 36. Two of the objects are recognized reliably, but the other ones were confused with similar alternatives most of the time.

For the second experiment, six brown plastic cups were chosen with five of them marked at one location with colored tape. This way they were distinguishable from the other cups only from a very limited set of viewpoints. On two of the cups, for example, the marker is placed inside the cups. These cups can only be recognized by looking directly inside—a challenging task when the movement is not pre-programmed. In Figure 37 the results for this experiment are shown. The average accuracy lies at 66.6% with results varying between 100% for the cup with



(a) Four paper tee boxes. These objects all have the same form and vary only slightly in appearance.



(b) Six brown plastic cups. The cups are identical except for a small colored sticker on five of them.

Figure 35: Objects used for evaluation on the real iCub robot.

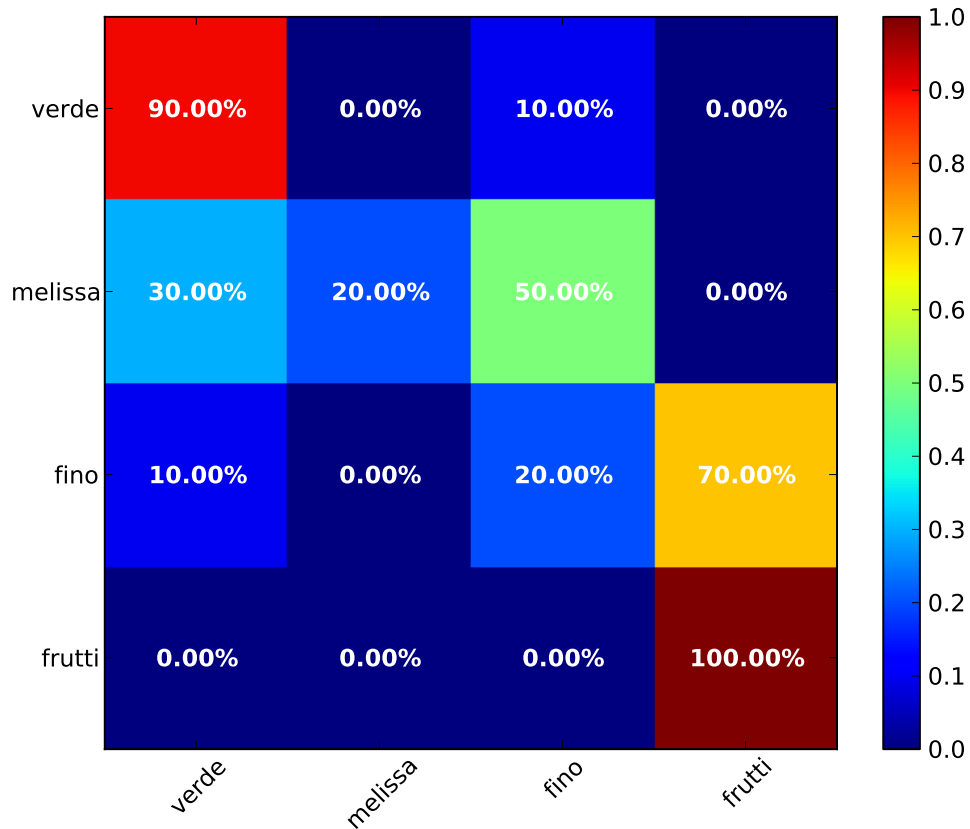


Figure 36: Confusion matrix for VTM recognition of tea boxes.

the marked rim (*rimcup*) as well as for the unmodified cup (*normcup*) and 16.7% for the cup with the blue sticker on the side (*bluecup*).

3.2.7 CONCLUSION

What was already visible in the first experiments in simulation becomes even clearer on the iCub: there is a hit or miss pattern in the results. Predictions are very 'sharp', leading to two effects: One is that often objects are either always recognized or never recognized. The other observation is that only few cells in the confusion matrices are occupied. This is caused by the fact that the algorithm makes hard choices and concentrates further evaluations on their outcome. This characteristic is a shortcoming of this approach. Instead, it would be desirable to keep track of many hypotheses at the same time and reason based on entire probability distributions. This insight was motivation for the implementation of a more sophisticated method based on a probabilistic interpretation of the data. This method is presented in the next section.

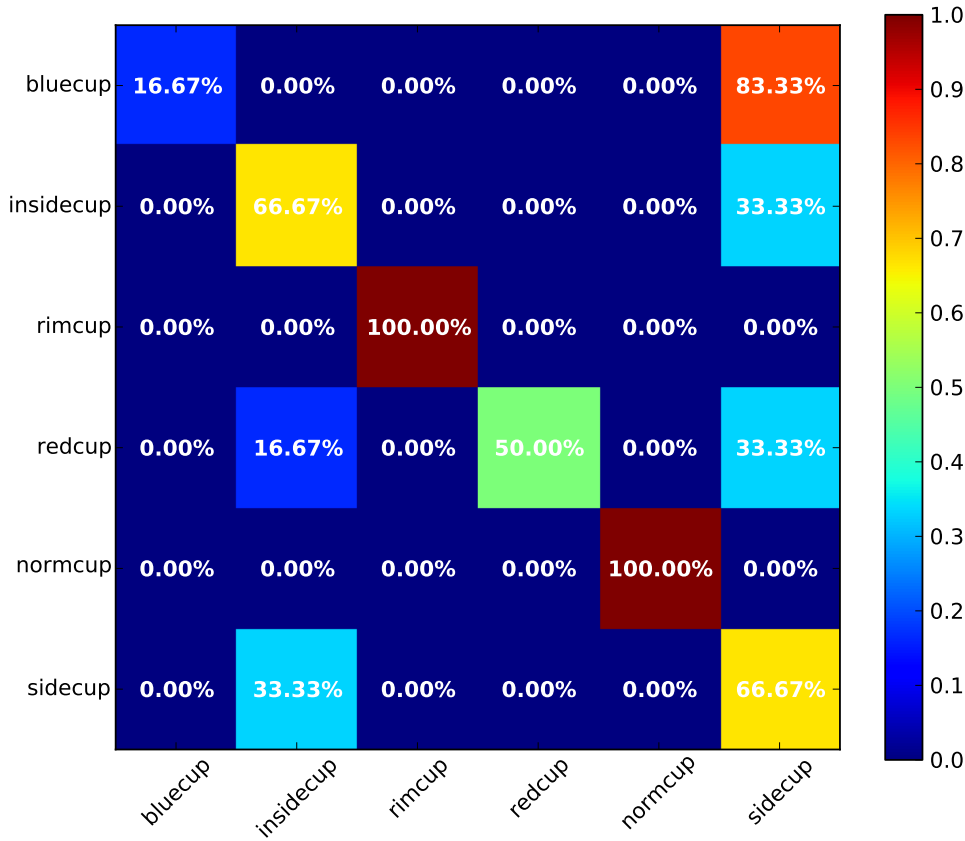


Figure 37: Confusion matrix for VTM recognition of plastic cups.

3.3 A PROBABILISTIC FRAMEWORK FOR ACTIVE OBJECT RECOGNITION

How can we find an object manipulation sequence that minimizes the number of actions needed to correctly identify the object? The presented solution to this question is based on the observation that for a given, unknown object, it will be most efficient to look for a view that yields the most additional information for discriminating it from similar ones. Hence, based on the current view, the associated object probabilities, and the history of actions, a motion should be selected that is expected to yield the highest information gain. In this case, actions consist of the rotation of an object in an object-centered coordinate system and involve the whole kinematic chain from hand to eye. The actions are executed using learned inverse kinematics in a 15 degree-of-freedom space.

Hypotheses about the object in question are created and updated as the recognition progresses. I define a hypothesis as an estimate about an object and the viewpoint onto this object, which gives rise to a specific 2D view. A viewpoint is defined as a location on a view sphere centered around the object. I adopt proba-

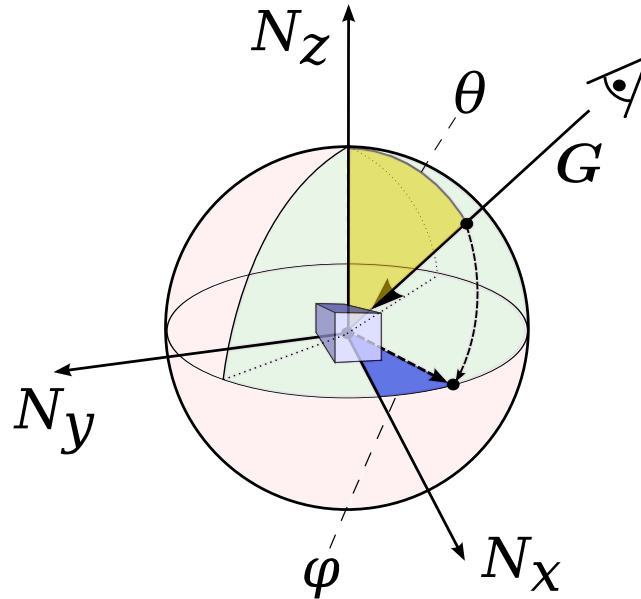


Figure 38: Illustration of the view sphere. Green area indicates accessible viewpoints on the view sphere. Viewpoint angles φ and θ are computed from gaze vector G and hand coordinate system N . Objects are located at the origin of N .

bilistic Monte Carlo localization methods to maintain a high number of hypotheses in parallel. By running particle filtering, regarding hypotheses as particles, one can take viewpoint changes into account in the form of proprioceptive information obtained from the robot arm. Object probabilities are calculated based on these hypotheses and an action is selected which is expected to minimize the uncertainty of the current estimate. In a nutshell, this formulation reduces the recognition problems to a localization problem which can be solved using established filtering techniques.

3.3.1 VIEWPOINT CONTROL

To explore an object systematically, one needs to be able to describe already seen viewpoints and desired viewpoints in an efficient and compact way. However, the joint space of a humanoid robot that needs to be controlled in order to achieve a desired viewpoint is usually high dimensional. On the iCub, the whole kinematic chain that can be used to manipulate an object within the field of view consists of 15 degrees of freedom (DOFs): 7 DOFs for the arm and wrist, 3 DOFs for the torso, and 5 DOFs for head and eye [Pattacini 10b]. In principle, one could fix some of these joints and, for example, only move the wrist while keeping a steady gaze onto the hand. However, due to motor constraints the space of accessible

viewpoints would be very limited. Therefore, all possible DOFs are incorporated to increase the range of motion as much as possible and to ensure that a high number of object views can be generated without re-grasping. A viewpoint is described in terms of two parameters, azimuth φ and elevation θ . They are defined in respect to the reference frame N of the robot hand and the gaze vector G between hand and eye as

$$\theta = \text{acos}(G \cdot N_z), \quad (11)$$

$$\varphi = \text{acos}(G \cdot N'_x). \quad (12)$$

N'_x denotes the vector N_x after tilting the plane $N_x \times N_y$ to be orthogonal to G . To improve readability, in the following I refer to (θ, φ) as a viewpoint ϕ . See Figure 38 for an illustration of the viewpoint definition.

A linear-weighted nearest neighbor search is employed to map from viewpoint ϕ to joint states \mathbf{q} . Sample points for the search are collected through random movement. Each sample consists of the current gaze angles and the joint angles of the DOFs.

To obtain a configuration in joint space that leads to a desired viewpoint, recorded samples are accumulated and a weighted joint average

$$\hat{\mathbf{q}} = \sum_i^N W_i \mathbf{q}_i \quad (13)$$

calculated from the N nearest neighbors according the sample weights W_i given by:

$$W_i = w_\Phi \Phi \langle \phi_i, \phi \rangle + w_X \|\mathbf{x}_i - \mathbf{x}\| + w_q \|\mathbf{q}_i - \mathbf{q}\|, \quad (14)$$

where $\Phi \langle \cdot \rangle$ denotes the angle between two gaze vectors. The second and third terms are optional and comprise additional optimization tasks. The middle term optimizes for a low Euclidean distance to a desired location in 3D space. The last term penalizes large changes in joint space to reduce jerkiness. These optimization goals are assigned individual weights w_Φ , w_X , and w_q .

The calculations are simple and run at real-time performance even for a large number of samples. To speed up computation even further, a subset of unweighted samples is selected solely based on viewpoint ϕ using an approximate nearest neighbor search to only take these samples into account for weighting.

The resulting poses $\hat{\mathbf{q}}$ produce viewpoints with a deviation of usually less than 3° from the desired viewpoint. The accuracy is sufficient since viewpoint changes below this margin are not expected to cause relevant changes in the image.

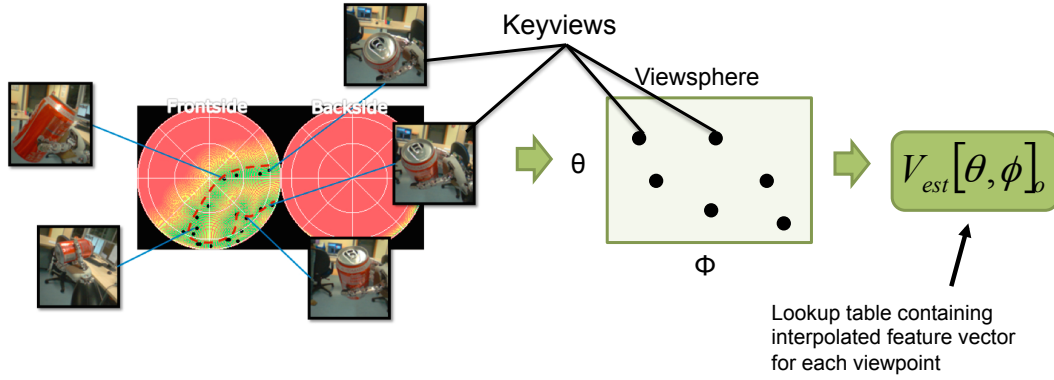


Figure 39: Example of an exploration sequence executed to learn a new object. The dashed red line shows the exploration path on the view sphere. Keyframes are marked by black dots.

3.4 OBJECT EXPLORATION AND LEARNING

For view-based three-dimensional object recognition, sample views need to be acquired from various viewpoints. These collections of views are then converted into representations of objects or object classes. It is desirable that the acquisition of these views is performed in an efficient way in terms of exploration time and motor actions. This point is addressed by keeping track of already seen viewpoints and selecting target positions that minimize

$$\phi_{t+1} = \operatorname{argmin}_{\phi'} e(\phi', \phi_t) \quad (15)$$

with

$$e(\phi', \phi_t) = \alpha \tau(\phi') \cdot (1 - \alpha) \Phi \langle \phi', \phi_t \rangle. \quad (16)$$

Equation (16) results in low values for viewpoints that are close to the current viewpoint but add as much new information as possible. The parameter α controls how fast the object is moved to new orientations. Low values result in a slow motion in which the object is examined carefully. High values, in contrast, lead to a coarse but fast exploration. The function $\tau(\cdot)$ in Equation (16) defines how well a certain view has been seen before and is defined as

$$\tau(\phi) = \max_{v \in V} 1 - \frac{\Phi \langle \phi, \phi_v \rangle}{\text{fov}}, \quad (17)$$

with fov denoting the angle of the field-of-view and V being all previously visited positions on the view sphere. We obtain values in the interval $[0,1]$ with 0 for a completely unknown view and 1 for an exact viewpoint match.

3.5 RECOGNITION AND MOTION PLANNING

Active recognition in the context of a humanoid robot differs from static (multi-view) object recognition scenarios by offering two valuable benefits. First, the object in question can be manipulated. Thus additional information can be generated by creating new views. Second, by incorporating proprioceptive information from joint states one does not have to rely on an unrelated set of views but can also take into account the viewpoint differences that caused the change in appearance.

Each view can be treated as an observation that adds information about object probabilities. Sequences of observations can be combined to form joint distributions of object probabilities. However the question arises of how to calculate object probabilities and how to select actions that lead to short sequences with discriminative views. This problem is addressed in the following.

3.5.1 RECOGNITION BY LOCALIZATION

Object recognition in this context can be regarded as a localization problem in which the goal is to find the most probable location on the view spheres of the objects. I define such a location as $x = (o, \phi)$, with o determining the object and ϕ defining the view angle. The probability distribution over all positions x is estimated at every recognition iteration t . We can calculate the posterior probability of x given a sequence of actions a and observations z as follows:

$$\begin{aligned} p(x_t) &= p(x_t | z_{1:t}, a_{1:t}) \\ &= p(z_t | x_t) \cdot \int p(x_t | x_{t-1}, a_t) \cdot p(x_{t-1}) dx_{t-1}. \end{aligned} \quad (18)$$

Equation (18) defines the posterior of a recursive Bayesian filter. As the exact solution is intractable, a particle filter is run to achieve a Monte Carlo approximation. The particle filter produces a discrete estimation of the true distribution using a large set of weighted samples, or particles $\{x^i, w^i\}_{i=1}^N$. The approximated posterior can be stated as

$$p(x_t) \approx p(z_t | x_t) \sum_{i=1}^N w_t^i \cdot p(x_t | x_{t-1}^i, a_t). \quad (19)$$

This particle filter implementation is based on the bootstrap filter proposed in [Gordon 93] and outlined in Algorithm 1. New particle positions are predicted by taking into account the viewpoint change from the last observation. It is important

to note that this information is acquired through proprioception by the active robot. After adding Gaussian noise $\mathcal{N}(\mu, \Sigma)$, the particle update rule is given by

$$x_t \leftarrow q(x_t^i | x_{t-1}^i, a_t) = x_{t-1}^{i,\phi} + a_t + \mathcal{N}(\mu, \Sigma). \quad (20)$$

New particle weights are computed based on the expected view at a certain location on the view sphere. This view is estimated from the extracted keyframes (see Section 3.4). To obtain an estimate on all possible particle positions, keyframes are interpolated on points that were not directly observed. The interpolation is done by calculating a weighted average from the k -nearest neighbors. The weight is set proportional to the angular distance to the neighboring keyframes. The interpolation needs to be done only once per object and can be precomputed offline. Using the resulting maps of view estimates $V_{est}[\cdot]_{k=1}^K$ as look-ups, the likelihood of an observation with feature vector z given a particle x^i can efficiently be calculated by

$$p(z|x_t^i) = (V_{est}[x_\phi]_{x_o} - z)^2, \quad (21)$$

and the particle weight updated by

$$w_t^i = w_{t-1}^i \cdot p(z|x_t^i). \quad (22)$$

Since the view sphere is limited due to motor constraints, one needs to decide on how to handle particles that leave the part for which we have gathered data. In this implementation these particles are set to low weights but do not discarded entirely. It is possible that the object is located differently in the hand of robot and currently observed views were not visible during training. Assigning lower weights, however, is necessary since otherwise hypotheses would be maintained that will not be updated in the future. Low weights will eventually result in a higher probability of being discarded during resampling.

To remove particles with low weights, particles are resampled in each iteration. According to the sampling and importance resampling (SIR) procedure [Gordon 93], each particle is replaced with another particle that is picked proportional to its likelihood of giving rise to the current observation (Equation 21). It is important to base the *resampling* only on the current view instead of taking into account past iterations. This way it is avoided that, in the absence of discriminative input, the filter converges to an arbitrary mode.

After resampling, the object probabilities $P_{1,\dots,K}$ are calculated by integrating the weights of the particles associated with object k ,

$$p_t^k = \frac{\sum_{i=1}^N \delta_{k,\pi_i} w_t^i}{\sum_{i=1}^N w_t^i}. \quad (23)$$

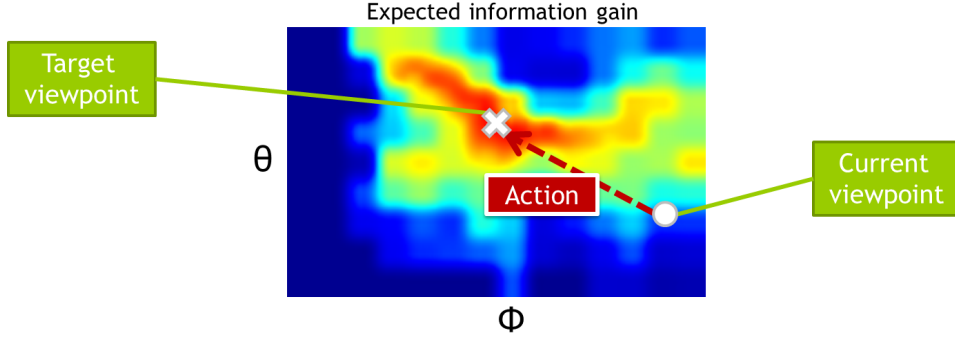


Figure 40: In this representation of the view sphere regions with high expected information gain are indicated in red, low information gain in blue. We want to find the action that moves the objects from the current viewpoint to a viewpoint with high information gain.

In Equation (23) $\delta_{i,j}$ refers to the Kronecker delta returning 1 for $i = j$ and 0 otherwise. The entropy H_t of the current object probability distribution can now be calculated as

$$H_t = \sum_{k=1}^K p_t^k \log p_t^k. \quad (24)$$

H_t describes the uncertainty of the current predictions and serves as an indicator of when a confident assumption can be made. Furthermore, H_t is used as an optimization target for determining the next action as described in the following subsection.

3.5.2 ACTION SELECTION

We are looking for an action a that is expected to lead to a viewpoint that maximizes the expected information gain. This can be formulated as

$$a = \arg \max_{\tilde{a} \in A} E[I_{\tilde{a}}] = H_t - E[H_{t+1}^{\tilde{a}}], \quad (25)$$

where $I_{\tilde{a}}$ denotes the information gain by executing action \tilde{a} . H_t is the current entropy across object probabilities. The entropy $H(X) = -\sum_{x \in X} x \log x$ of a random variable X can be utilized to indicate the information content of the current estimate. To maximize the information gain, we need to find an action that minimizes the expected entropy. The expected entropy can be calculated by integrating over the range of expected observations m produced by action \tilde{a}

$$E[H_{t+1}^{\tilde{a}}] = \int_m H(P_t(m)) dm. \quad (26)$$

Algorithm 1 Particle filter object recognition

for $k = 1 \rightarrow K$ **do** ▷ Initialization

for $n = 1 \rightarrow N/K$ **do**

$i \leftarrow k \cdot n + n, \pi_i \leftarrow k$

 Draw particle x_i randomly from view sphere
of object k .

 Assign initial weights:

$w_0^i = p(z_0|x_0^i)$

end for

end for

$t \leftarrow 0$

repeat ▷ Iteration

$t \leftarrow t + 1$

 Execute action a_t .

 Acquire observation z_t .

 Update particles:

for $i = 1 \rightarrow N$ **do**

$\tilde{x}_t^i \leftarrow q(x_t^i|x_{t-1}^i, a_t)$

$\tilde{w}_t^i \leftarrow w_{t-1}^i p(z_t|\tilde{x}_t^i)$

end for

 Normalize weights:

$\tilde{w}_t^i \leftarrow \tilde{w}_{t-1}^i / \sum_{j=1}^N \tilde{w}_t^j, \quad i = 1 \rightarrow N$

 Resample particles:

 Select N particle indices $j_i = 1 \rightarrow N$ proportional to particle likelihood of current observation:

$j_i \leftarrow l \propto \frac{p(z_t|\tilde{x}_t^l)}{\sum_{j=1}^N p(z_t|\tilde{x}_t^j)}, \quad l = 1 \rightarrow N$

$x_t^i \leftarrow \tilde{x}_t^{j_i}, w_t^i \leftarrow \tilde{w}_t^{j_i}$

 Calculate object probabilities:

$P_t^k = \sum_{i=1}^N w_t^i \delta_{k, \pi_i} / \sum_{i=1}^N w_t^i, \quad k = 1 \rightarrow K$

 Calculate entropy:

$H_t = \sum_{k=1}^K P_t^k \log P_t^k$

until $H_t < H_{des}$

Equation (26) is approximated by sampling observations $z_{1,\dots,M}$ at view sphere locations $\tilde{x} = g(\tilde{x}|x_t, \tilde{a})$ from the view estimation map V_{est} . Consequently $E[H_{t+1}(\tilde{a})]$ can be stated as

$$E[H_{t+1}(\tilde{a})] \approx - \sum_m^M H(P_t(m)) \frac{p(z_m|\tilde{a})}{\sum_m p(z_m|\tilde{a})}. \quad (27)$$

Object probabilities $P_t(m)$ defined previously in Equation (23) are dependent on the set of particles \tilde{x} propagated by action \tilde{a} and the sampled view z_m :

$$P_t^k(m) = \frac{\sum_i^N \delta_{k,\pi_i} w_i p(z_m|\tilde{x}_i)}{\sum_i^N w_i p(z_m|\tilde{x}_i)}. \quad (28)$$

An observation z_m is assumed to occur with a probability only dependent on the position on the view sphere and not being dependent on the action performed. Thus, $p(z_m|\tilde{a}) = 1/M$ for all z_m . This, however, does not imply that actions do not affect the expected observations. The sampling position of the observation still depends on the action and its start position. Finally, after inserting Equation (28) into Equation (26) we obtain

$$E[H_{t+1}(\tilde{a})] \approx - \frac{1}{M} \sum_m^M \sum_k^K \frac{\sum_i^N \delta_{k,\pi_i} w_i p(z_m|\tilde{x}_i)}{\sum_i^N w_i p(z_m|\tilde{x}_i)} \cdot \log \frac{\sum_i^N \delta_{k,\pi_i} w_i p(z_m|\tilde{x}_i)}{\sum_i^N w_i p(z_m|\tilde{x}_i)} \quad (29)$$

Unfortunately, the evaluation of Equation (29) is time-consuming. However, it is not necessary to know the exact entropy values to find a favorable action. Instead, by following the reasoning in [Fairfield 08] an action \tilde{a} can be searched that leads to viewpoint which introduces a high amount of variance across different objects:

$$\tilde{a} = \arg \max_{\tilde{a} \in A} D_{\tilde{a}} \approx E[D_{\tilde{a}}] = \sum_i^{\hat{M}} \sum_j^{\hat{M}} (\tilde{z}_i - \tilde{z}_j) \cdot \kappa_{i,j} \quad (30)$$

with

$$\kappa_{i,j} = \begin{cases} \alpha & \text{if } i = j \\ \beta & \text{else} \end{cases} \quad (31)$$

where \tilde{z} represents the expected observation feature vector

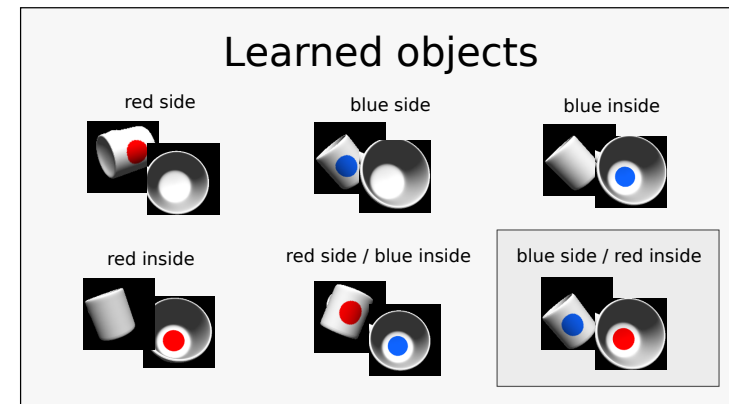
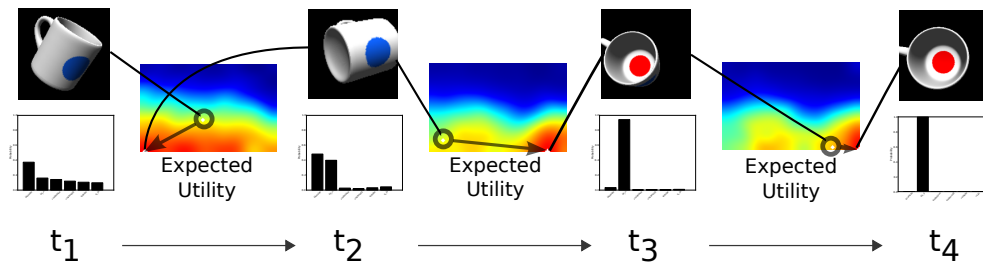
$$\tilde{z} = V_{est}[g(\tilde{x}|x, \tilde{a})] \quad (32)$$

To speed up computation, Equation (30) is not evaluated over all particles x^M . Instead only a subset of particles $x^{\hat{M}}$, $\hat{M} < M$ is considered with probability proportional to their weight w . By modifying κ the algorithm can be steered towards

discriminating between views within the same objects. Setting $\alpha = 1$ and $\beta = 0$ only takes into account views that lie on different objects. This is useful when the task consists less of identifying objects but rather of determining the specific object pose. However, even when the object pose is not of interest it is beneficial to set β to a positive value. When the recognition is approaching saturation, most particles will then be located on one object.

The exploration process is illustrated in Figure 41. It shows individual recognition iterations for a rendered mug on the iCub simulator. In the first sequence at the top, motion planning is enabled. In the second sequence, the iCub looks at random viewpoints. With motion planning, we see that in the first iteration there is a high information gain for actions that lead to viewpoints in which the blue dot is visible (bottom left area). As soon as the probabilities for the two objects that contain a blue dot on the side is increased, we receive a high information gain for views inside the cup (bottom right area). Looking at these two viewpoints allows the robot to quickly recognize the correct object. In contrast, if random viewpoints are explored, we see that the object cannot be identified correctly until an inside view is achieved in the 12th iteration.

Planned exploration



Random exploration

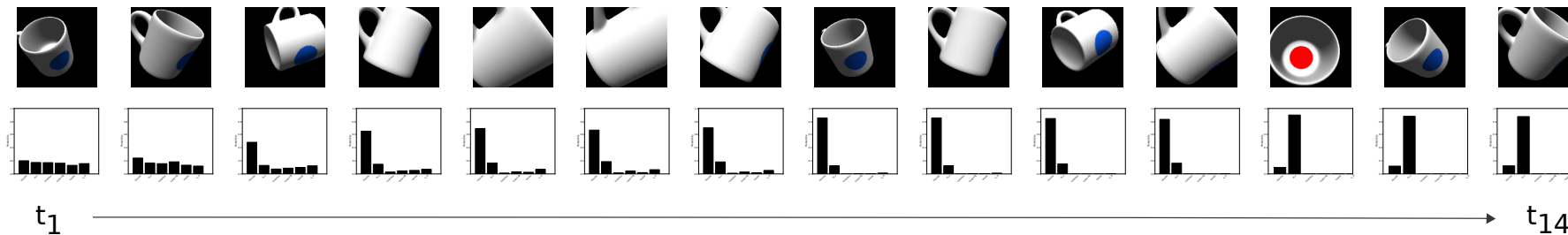


Figure 41: Recognition sequences for a white cup marked with a blue and a red dot. Top left: exploration using motion planning. Bottom: recognition of the same cup but without planning. Top right: the six objects available in the dataset. For planned exploration, heatmaps visualize the expected utility of actions leading to respective positions on the viewsphere. Red color indicates regions with high expected information gain. Bar plots below the object views show the joint probability distribution at time t .

3.5.3 INCORPORATING GLOBAL OBJECT INFORMATION

Recognition speed can be boosted even further by exploiting global information about the object. For example, if we see an object that contains blue areas we might not know what it is, however, we can assume that it is most likely not a banana or an apple. This reasoning is incorporated into the recognition process by unsupervised clustering of all keyframes across all objects into the set of clusters $\{c\}_K$. The clusters are computed by running the K-means clustering algorithm [Steinhaus 56, MacQueen 67] on the feature vectors. The object probability given a certain cluster c is then calculated using a Bayesian estimator:

$$P(o|c) = \frac{P(c|o)P(o)}{P(c)} \cdot P(c|z). \quad (33)$$

The likelihood of a given cluster is equivalent to the distribution of keyframes of this object across the set of clusters: $P(c|o) = |KF|_c/|KF|_o$. $P(c)$ is determined by the relative size of cluster c , $P(c) = |c|/|KF|$. As no object is expected to occur more frequently than any other, $P(o)$ is set to a constant value. The probability of a cluster c given an observation z is defined based on the Euclidean distance to the cluster center:

$$P(c|z) = \exp^{-\|c-z\|_{L2}}. \quad (34)$$

The total object probability given feature z ,

$$P(o|z) = \frac{\sum_i^K P(o|c_i)}{\sum_i^K P(c_i|z)}, \quad (35)$$

is then combined with the particle weight of all particles assigned to object o . Thus, if a view is observed that was prevalent during learning, the probability of this object is increased and vice versa. We expect the particle filter to reject some hypotheses quickly and narrow down the choice of object candidates to the remaining very similar and possible ones.

3.6 EVALUATION

As the actual grasping is not part of this study, all experiments are started with the object already grasped and located in the hand of the robot. To demonstrate the ability of the system to distinguish even between highly similar objects, evaluations are conducted in simulation and on the real robot on objects that share many similar views.

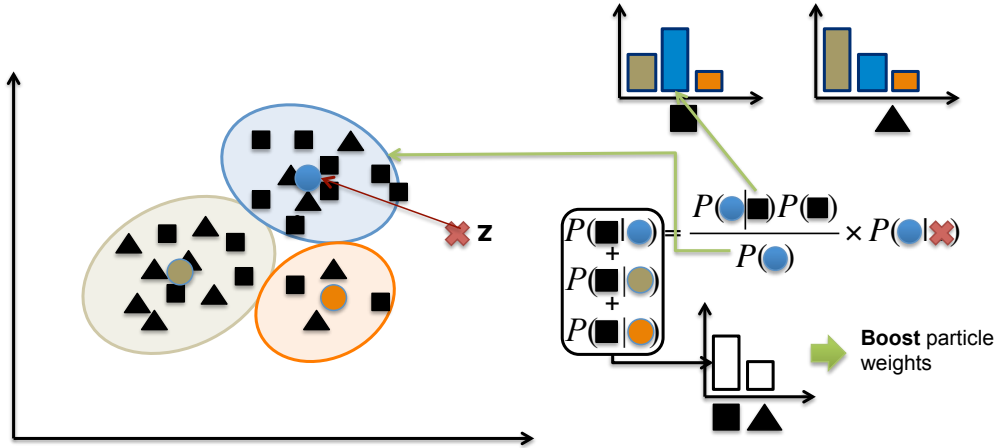


Figure 42: Particle boosting based on global image statistics across learned objects. In this example, features from two different objects (triangles and squares) are marked in two-dimensional space. The three ellipses represent three clusters among these features. For an unknown feature z the probability of being a triangle or a square is calculated based on the evaluation of the clusters.

Objects were learned as described in Section 3.4. Since looking at the back of the hand is hardly desirable, the exploration was restricted to the upper hemisphere. Furthermore, as noted earlier, the robot is not able to bring the hand into all orientations a human can. For example, it is not possible to have the robot look directly onto its fingertips from the front. Therefore, the accessible area of the viewsphere was limited for exploration as well as for recognition to $[0^\circ, 90^\circ]$ elevation and $[160^\circ, 270^\circ]$ azimuth. During the exploration phase, between 50 and 130 keyframes were recorded per object in the automatic exploration mode.

For both experiments 40 recognition trials were conducted for each object. 10 trials with planned actions using the variance maximization scheme discussed in Section 3.5 and 10 trials with viewpoints randomly selected on the previously defined part of the view sphere. It should be noted that picking random positions on the view sphere does not lead to views that are as random as setting random joint angles within a certain range. By selecting view sphere positions, the same poses will be available that were potentially obtained during learning and can be set by the motion planning algorithm. Using completely random joint or task space positions would make object recognition even more difficult and would not represent a viable baseline. After each trial, a random position was set to ensure independence between individual trials. The remaining 20 trials consisted of repeating both conditions, motion planning and random movement, this time with taking into account global appearance information (boosting).

The recognition process was stopped after 15 iterations. The particle filter was initialized with 150 particles per object—uniformly sampled on the view sphere of each object. Motion planning was performed taking into account 300 particles sampled with probability proportional to their weight. Actions are distinguished up to a resolution of $\pm 1^\circ$ on the view sphere and computation time was reduced by performing a grid search on a recursive raster of 9×14 elevation and azimuth cells. The total planning time was approximately 0.5s per iteration.

To compare the new approach against pure visual recognition (i.e., not taking into account proprioceptive information), k-nearest neighbor matching was performed with the recorded keyframes. This was done for all trials in parallel to the particle filtering. Matchings using one, three, and five neighbors were evaluated. As the resulting differences are marginal, we only report results for three nearest neighbors.

3.6.1 EVALUATION IN SIMULATION

Experiments were conducted on the iCub simulator [Tikhanoff 08] using the same 3D models as test objects as previously in Section 3.2.5 for the View-Transition-Maps. We measured the ratio of correctly recognized objects as well as studied how changes in the object orientation due to different grasp position affect the results.

3.6.1.1 *Recognition performance*

Results are shown in Figure 43. We see that planned motion leads to more correct results and more confident predictions. The latter is indicated by the much faster decreasing entropy in Figure 44. If we look further, we see that particle filtering significantly outperforms k-nearest neighbor matching. In all conditions we observe a significantly higher accuracy at much faster recognition rates. Interestingly, conditions with tighter motion planning or particle filtering enabled show a very similar accuracy characteristic. This may be seen as further evidence that in order to fully leverage the benefit of planned object exploration knowledge about the executed motion and manipulations needs to be fed into the recognition process.

As pointed out in Section 3.5.3, we expect to be able to speed up the recognition process even further if we take into account global object information. This way we are able to exploit statistics in the learned object database. Characteristic views are implicitly assigned a higher weight. In Figure 43 and Figure 44 this condition is indicated by square markers. With planned particle filtering the performance gain

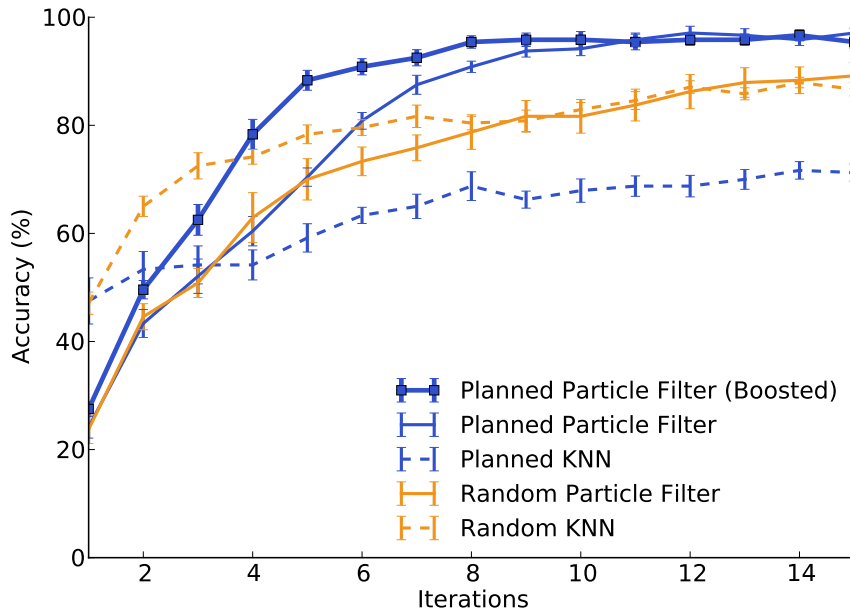


Figure 43: Iteration accuracy in simulation.

through this boosting technique is very well visible. Objects are recognized much faster—especially in the first few iterations—while even increasing the precision. However, for all other conditions there are no measurable benefits. This, in fact, does make sense as any gain comes at a cost. If we want to speed up the recognition process, thus make predictions based on less input data, it is crucial that the information content in this input data is accordingly higher.

3.6.1.2 Sensitivity to grasp changes

As we work with viewpoint changes instead of absolute viewpoints and absolute joint configurations, we should be able to recognize objects even if grasped in a different way, as long as the overlap between the learned object parts and the now visible ones are large enough. To test this, the object was placed in the robot hand rotated around the vertical axis and ran the same evaluation procedure as before. It is important to bear in mind that the accessible area of the view sphere ranges between 160° and 270° in this dimension (azimuth). The default orientation that was used during learning is equivalent to an azimuth location of 180° . This leaves 20° of exploration space in one direction and 90° in the other. If the object is now placed in the hand rotated by -20° (equivalent to an azimuth of 200° during learning) the overlap with the previously seen area is 70° . However, if it is placed at $+20^\circ$ (azimuth of 160° during learning) the robot has the chance of inspecting 110° on the learned viewsphere. Not being able to see parts of the object is only one part of

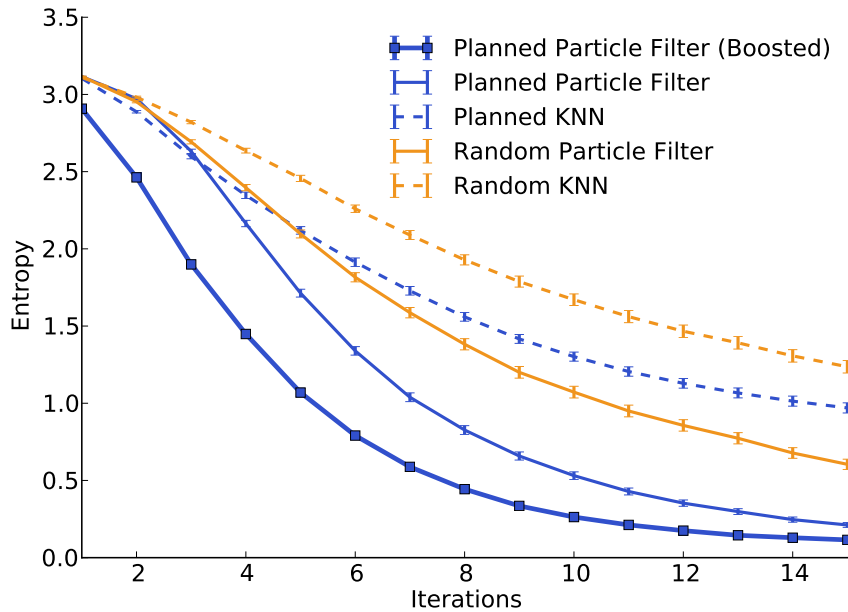


Figure 44: Iteration entropy in simulation.

the problem. The robot will instead be able to see completely new views that can look very different from the known ones and might lead to wrong interpretations. As a consequence, objects rotated in negative direction are more difficult to recognize than rotations in the positive direction. This becomes particularly noticeable as the rotation amount increases. Figure 45 shows the accuracy after 15 iterations for rotations between -30° and $+60^\circ$. The figure shows a much smaller drop in performance for planned motion in contrast to random exploration. Especially for rotations of more than $+15^\circ$ we observe a considerable performance decrease if the robot cannot actively bring the object into favorable orientations. Furthermore, for increased rotations we see that motion planning becomes the dominant factor as the differences between particle filtering and kNN become smaller. In contrast to kNN matchings, particle filtering suffers from the fact that often particles cannot be updated correctly because they are shifted to unknown areas of the viewing sphere.

3.6.2 REAL-WORLD EVALUATION

On the iCub, recognition experiments were conducted with three sets of objects (Figure 46 varying in shape and degree of similarity). Evaluations were carried out with the same parameters for the particle filter and the same experiment setup as in the simulation. However, the number of trials and iterations in the three experi-

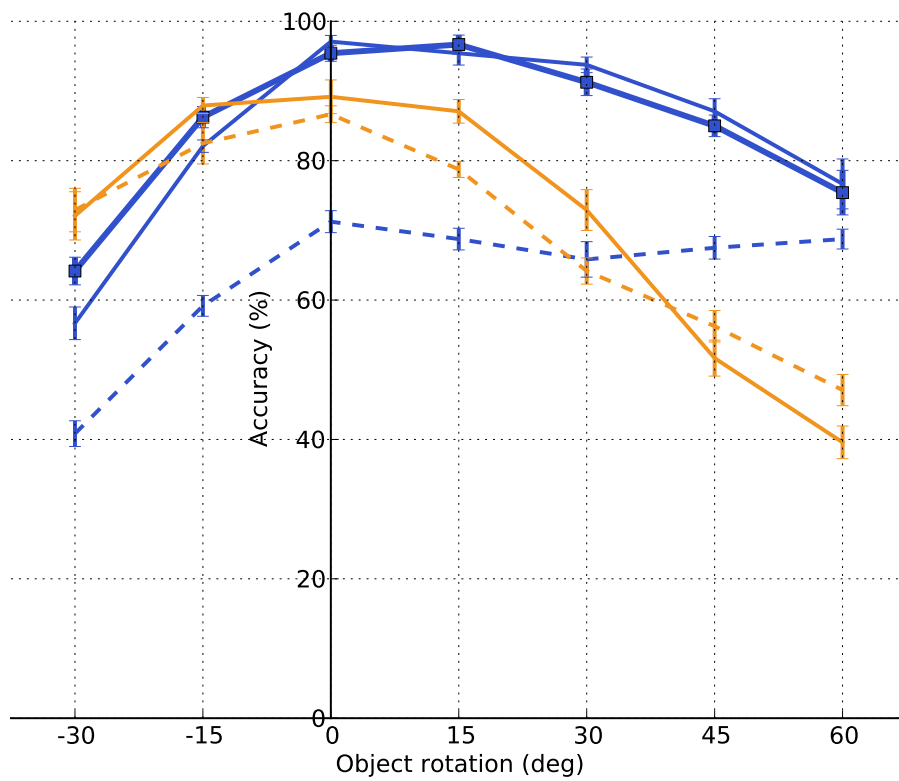


Figure 45: Accuracy after 15 iterations for different amounts of object rotation.

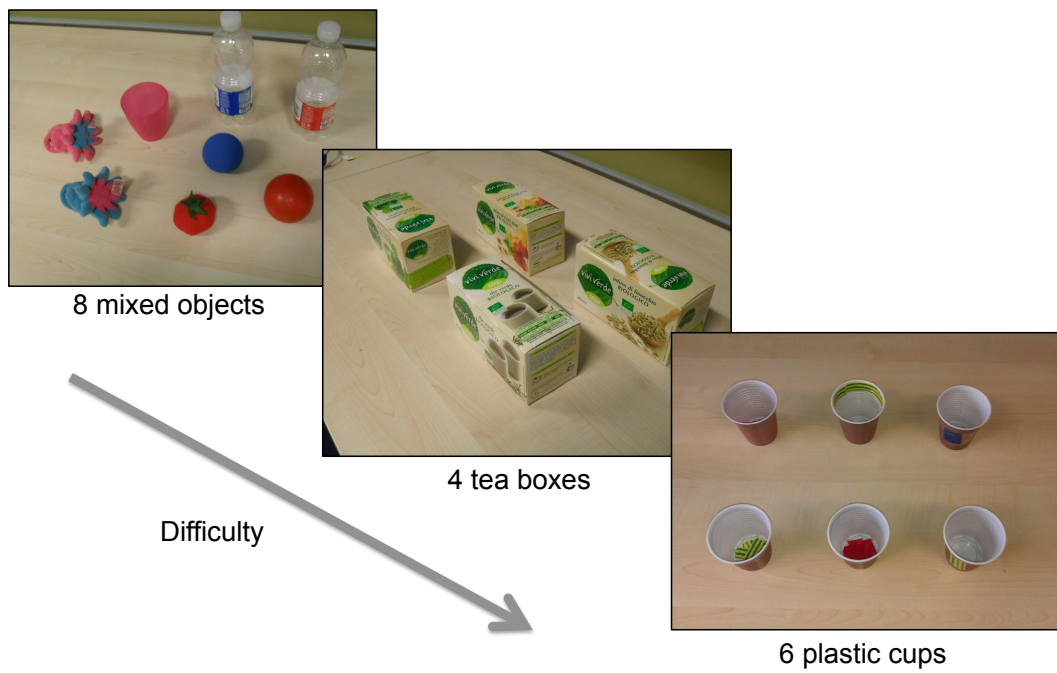


Figure 46: Objects used for evaluation on the real iCub robot.

ments was changed. Another difference was necessary as a result of modifications in the robot hardware. Due to thicker covers on the arm, the maximum range of the shoulder joints to restrict by a few degrees. This reduced the explorable part of the viewsphere to $[20^\circ, 90^\circ]$ elevation and $[160^\circ, 270^\circ]$ azimuth.

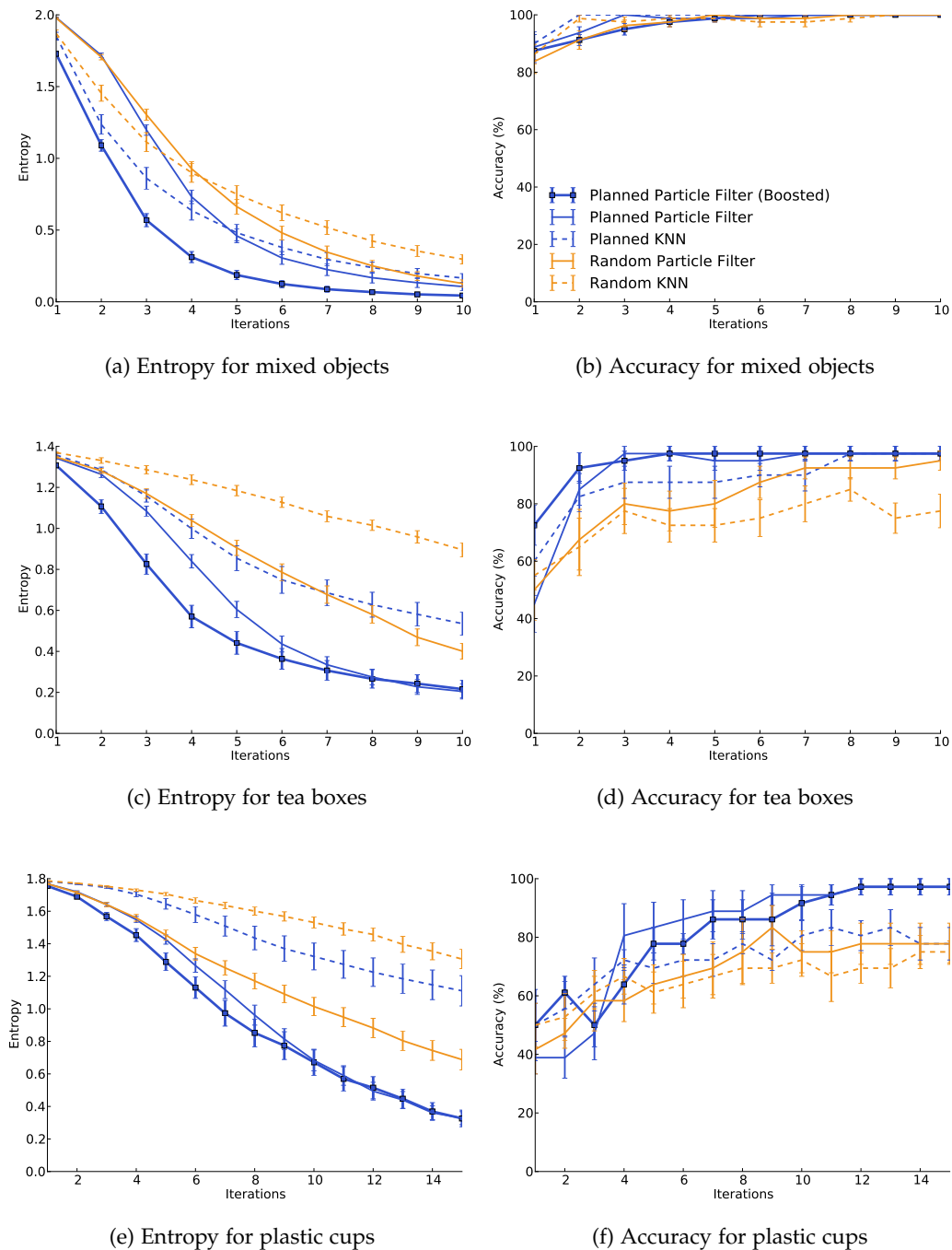


Figure 47: Results of object recognition experiments on the iCub.

3.6.2.1 *Evaluation with mixed set of objects*

In the first experiment eight objects were used that were found in the office. These were eight common everyday objects or toys often used for experiments and demos with the iCub. As only color histograms are used and shape is not important, objects were selected based on their color. Each object is a member of a subgroup of objects that shares similar views. The objects are depicted in Figure 46 (upper left). 10 trials were run for each condition with 10 iterations. In Figure 47b the recognition accuracy for all eight objects is plotted for the 10 iterations. Even with our very basic features distinguishing these eight objects seems to be an easy task. After a few iterations all objects were recognized in all trials. For real-world application, however, a vision system needs to have an acceptance threshold at which the process is stopped and one of the candidates is selected as final result (or stopped without a result). To determine this threshold, entropy can be considered as a suitable choice. By looking at Figure 47a we see that kNN and particle filtering show similar performance here. On enabling boosting, however, we gain a significant speed-up. Entropy drops below 0.5 after only three iterations. This can be seen as a good indication that the recognition process has terminated and the current hypothesis is unlikely to be changed by executing more iterations (i.e. adding more information).

3.6.2.2 *Evaluation with set of similar tea boxes*

In the next experiment the same tea boxes are used as previously in Section 3.2.6 (Figure 46, middle) for the View-Transition-Map. As in the first experiment with the mixed set of objects, 10 trials were conducted with 10 iterations each. In this more challenging experiment we see a difference in recognition rates in Figure 47d. Adding boosting, motion planning and taking into account actions between observations (using particle filtering) clearly results in a performance gain. Figure 47c shows again the improvements over random exploration exploiting only visual information indicated by the much faster dropping entropy.

3.6.2.3 *Evaluation with set of highly similar plastic cups*

In the last and most difficult experiment the six marked plastic cups were used again (Figure 46, bottom right). This experiment consisted of six trials with 15 iterations each. In Figure 47e we see the same pattern as in the previous experiments. Likewise, the plots in Figure 47f lay out the benefit of motion planning. In this experiment it was very important to look at the right position of the objects. This

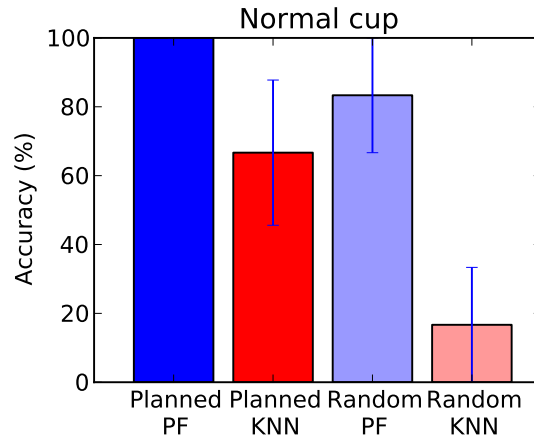


Figure 48: Recognition performance for the unmarked brown plastic cup after 10 iterations.

may explain why there is no significant difference between particle filtering and kNN. For most objects it is sufficient to find a good viewpoint and less critical how to combine the current view with previous observations. For objects without a characteristic viewpoint, such as the unmodified, normal cup this does not hold as will be shown in the next subsection.

3.6.2.4 Recognition by rejecting alternatives

If we look more closely at the results of individual objects, we find that for certain objects motion planning and particle filtering are particularly important. In Figure 48, for example, the accuracy of the unmarked brown cup ('Normal cup') after 10 iterations is shown. With random, vision only exploration performance ranges around chance level which is 16.7% for six objects. Adding more cues raises the performance significantly (Planned: 67%, PF: 83%, Both: 100%). This effect is also present in the simulation experiments for the object (white box) that bares no apparent mark by which it could be identified. From the way the experiment is set up, it becomes clear that unmarked objects can only be recognized by *rejecting alternatives*. This, however, can only be achieved by reasoning across multiple observations and regarding subsequent views in relation to each other.

3.6.2.5 Observed viewpoints

It is very interesting to investigate which viewpoints were visited during the recognition of individual objects. In Figure 49 view sphere positions are marked for each object during the six planned recognition trials. Dots are scaled according to iteration number with the first iterations represented by the smallest dots. We see that the distribution is different for all objects. The upper middle and bottom left area

in the plots corresponds to positions in which a side view of the object is obtained (here, for example, the side stickers could be seen) the bottom right-hand area contains the viewpoints that allow to view into the cups. The regions inbetween mostly indicate poses in which the bottom of cup could no longer be seen, but the inside rim was well visible. The patterns look similar for the normal cup and for the two cups with modified inside bottoms. This means that these objects were mostly inspected by looking inside and at the side. In contrast, for the cup with the yellow-green sticker on the side, the robot only looked inside a few times during the first iterations (indicated by small dots). The cup with the modified rim was most of the time held in a position that allowed to see the rim, but would ensure that in this viewpoint the marker of the other two objects with similar yellow-green stickers would be not visible.

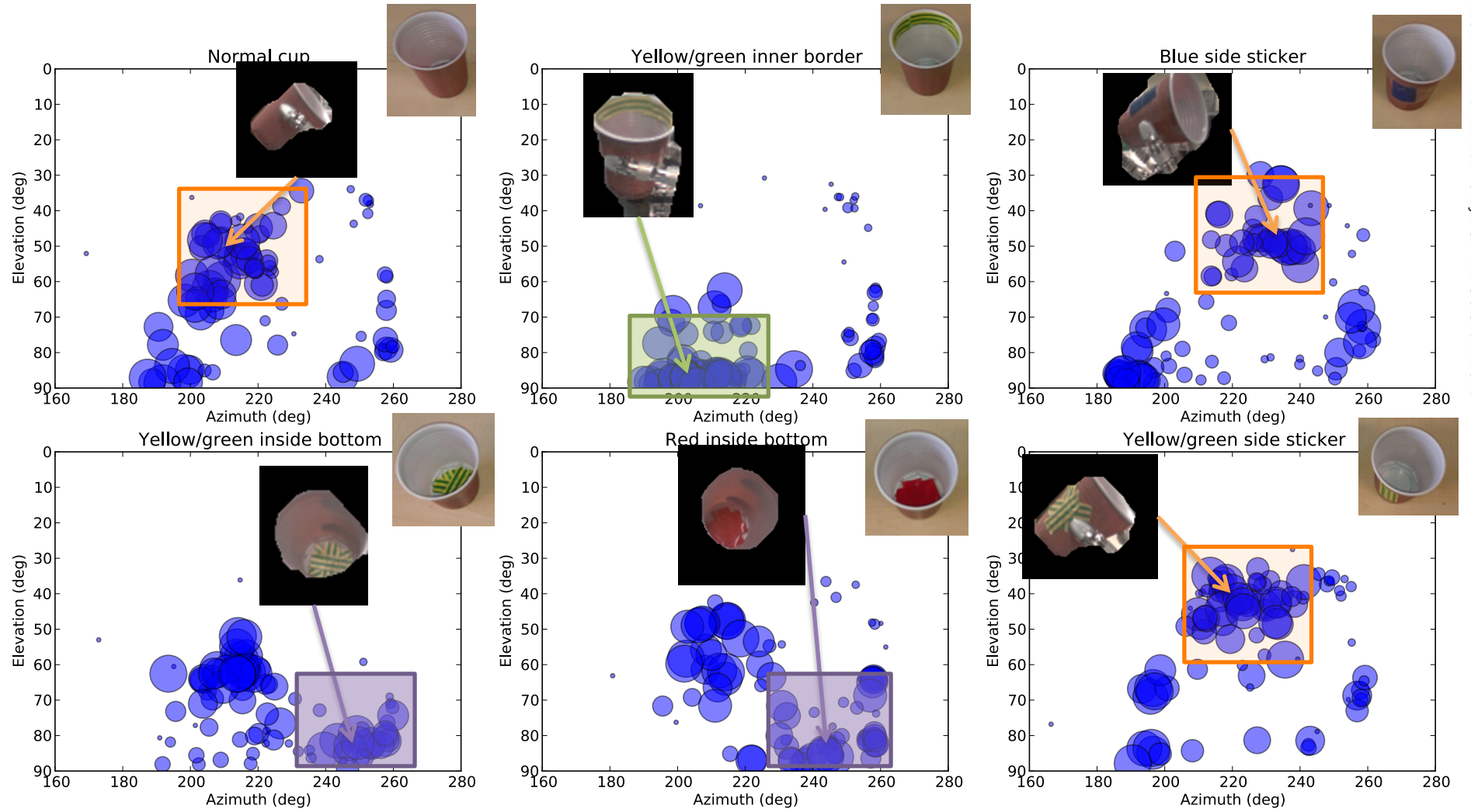


Figure 49: Observers viewpoints during 10 recognition trials. Dot sizes correspond to iteration number.

3.7 CONCLUSION

This chapter detailed a perception-driven object recognition process that allows a humanoid robot to recognize even highly similar objects by actively resolving ambiguities. Simulation and real-life experiments demonstrate that by predicting optimal viewpoints, objects can be identified much faster and more reliably. In addition, it shows that some difficult objects can only be recognized by rejecting all possible alternate hypotheses—this would not be achievable without active viewpoint planning. The incorporation of proprioceptive information (that is, that we can work in the joint angle space of the robot) also results in a significant improvement over visual-only comparisons by speeding up the recognition process considerably. In future work, this framework could be optimized to deal with a large number of concurrent hypotheses—after all, in typical use cases, the robot will be expected to deal with a large number of objects (and object categories). Finally, it might be possible to extend this framework to a more abstract search space, in which objects are disambiguated not by viewpoints but by performing certain actions (such as taking an object and trying to fit it into one of several differently-shaped slots). Perception-action skills like this will allow the robot to become an active explorer and to learn about its environment by interacting with it—similarly to the stages in infant development.

DISCUSSION

*All truths are easy to understand once they are discovered;
the point is to discover them.*

— Galileo Galilei

In this work I studied the integration of multimodal cues and active methods for robotic object recognition. In Chapter 2 I first reviewed current technology to capture and fuse color with depth information. In particular, I made use of Time-Of-Flight range sensing and fused the resulting depth map with RGB data from 2D color cameras. Thus, a high-resolution scene representation joining color and shape data was obtained. I then classified common everyday household objects based on this method. However, to conduct a systematic evaluation it was required to create a new testbed for 2D and 3D object classification. Since no suitable database was available at the time of writing a new, large-scale object dataset had to be created. This recorded dataset contains 18 object classes with 3-14 exemplars per class. The entire collection includes 154 objects with 5544 views. The dataset was made publicly available. The download link can be found in Section 2.3.

I proposed a composite classifier (cf. Section 2.5) that combines single classification results from multiple feature types. These features differ in the modality of the input data (2D vs. 3D) and how information is represented (local vs. global). The combination of individual predictions to a joint estimate is learned from the data and is different for each class. One type of objects may be more sensitive to feature A and C, whereas for another class feature B and D might be important. The results of the evaluations show that incorporating both 2D and 3D cues increases classification accuracy. Especially for certain classes, performance gains are substantial. This is not surprising as instances of many object classes rather share a common shape and outline than a similar color or texture appearance (e.g., *books, drink cartons*). Combination of different cues from the same modality (e.g., only 2D features), however, does not lead to a significant improvement. This is not very surprising as there is a large amount of redundancy within the features of

one modality. Performance gains through sensor or feature combination can only be expected if heterogeneous information is fed into the classification algorithm. This should be kept in mind when designing a recognition system—rather choose fewer but more diverse inputs.

In Chapter 3 I shifted my attention towards the incorporation of object manipulation into the recognition process. This chapter discussed object recognition in a closed loop, directly linking perception and action. I implemented modules for the iCub humanoid robot that enable it to explore objects in order to learn their appearance and recognize them even among very similar other objects. For this, the iCub holds an object in its hand and is able to move and rotate it. In Section 3.2 I presented an approach that executes these motions based on View-Transition-Maps. VTMs encode object views and their relations to each other in terms of joint configuration differences. Results from experiments in simulation as well as on the real robot show the merit of being able to verify object hypotheses by regarding a combination of viewpoints. Possible viewpoints, however, are limited to the trajectory that was executed during object learning. Furthermore, this approach lacks action planning capabilities that are needed to fully exploit the potential of active processes.

In Section 3.3 I proposed a method that allows the robot to choose the next viewpoint based on the information it has gathered so far during the current recognition run and the object data it has learned beforehand. The application of particle filtering makes it possible to maintain multiple hypotheses about the object identity and object viewpoint at the same time. The next viewpoint is chosen by calculating the action that is expected to lead to the highest information gain given the current knowledge. This computation takes into account the entire probability distribution across objects and viewpoints. In consequence, the algorithm is never forced to make hard decisions or follow heuristics. I demonstrated in multiple experiments in simulation and on the actual robot that this probabilistic approach is very powerful at discriminating between highly similar objects. Information about characteristic object views is exploited to steer the exploration sequence directly to obtain these highly informative views. In other words, using this method the robot finds out where to look and knows how to turn the object to see what it is interested in.

4.1 FUTURE WORK

There are many possible ways to continue and extend this work. In the following, I will mention some interesting directions for future research.

The MPI-IPA dataset presented in Section 2.3 should be extended in number of object classes as well as number exemplars per class. As shown in the results in Section 2.6.3 classification performance is sensitive to the amount of training objects. Plus, a higher number of classes would allow to study methods that leverage information shared across categories. Work on hierarchical (2D) object classification [Fidler 08, Bosch 08] has demonstrated the virtues of this approach. The dataset is also lacking capturings in which objects are presented in a cluttered scene. This would be necessary to evaluate segmentation algorithms that find and separate the object from the background.

The results from both studies, multisensory classification on the Care-O-bot and active recognition on the iCub could be combined to tackle object categorization by exploiting exploration capabilities. A system could join the probabilistic filtering technique from Chapter 3.3 with range sensing devices from Chapter 2.

Finally, one of the most important subjects when talking about multimodal object perception has not yet been dealt with. So far, haptic information is not incorporated into the system. In fact, already at the beginning of this work it was intended to add haptic cues. I started by experimenting with tactile fingertip sensors on the iCub. Tests and results are discussed in the next section. Unfortunately, I could not complete my work in this direction. The hardware was experimental and technical problems regarding the durability of the sensors were encountered. The output quality degraded too quickly to be employed permanently. In the course of this work the sensors were modified and repaired multiple times and finally discontinued. However, initial results and conclusions drawn from the following experiments are encouraging.

4.1.1 TACTILE FEEDBACK

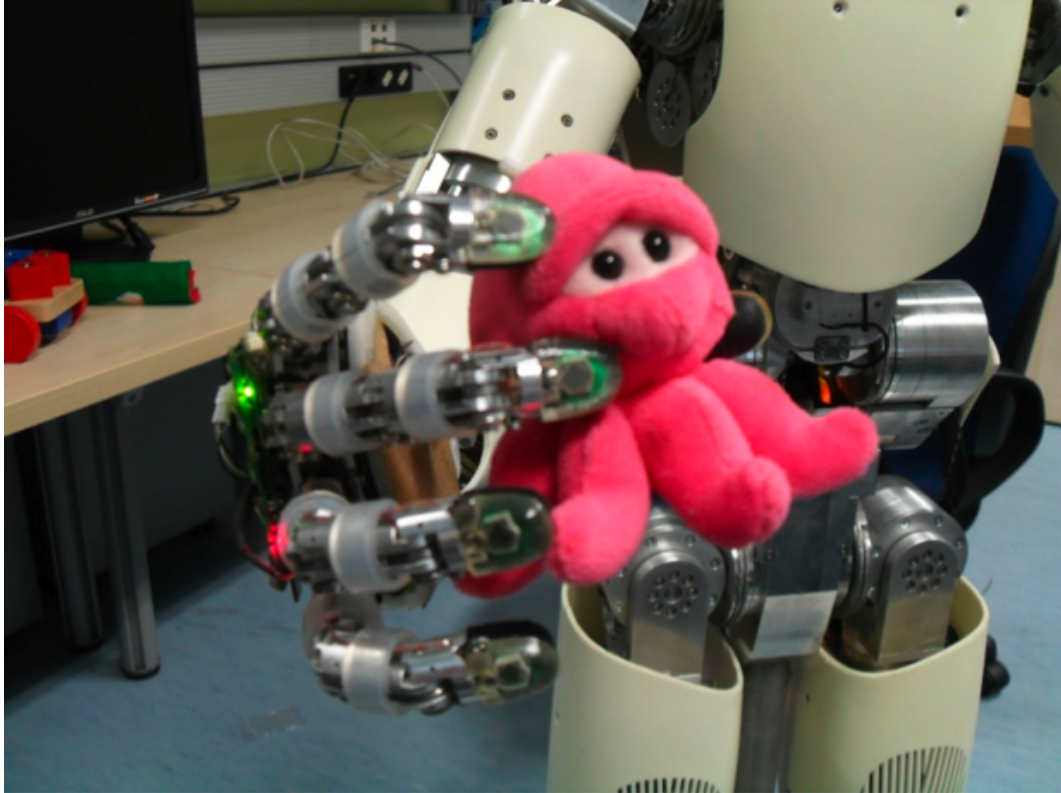
The iCub is equipped with a set of experimental tactile sensors. See Figure 50b and 50c. The fingertips are attached to the iCub enabling object grasping with the same degree of freedom as a human hand. They function as a capacitive pressure sensor. A printable circuit board (PCB) is mounted on an inner support element. This is surrounded by soft dielectric silicone foam and covered by a conductive silicone rubber layer. The electric charge of the capacitor is dependent on the distance

between the two conductive layers. The dielectric foam deforms under pressure altering the distance of the conductors. The foam also adds compliance to the fingertips. For more details please refer to [Schmitz 10].

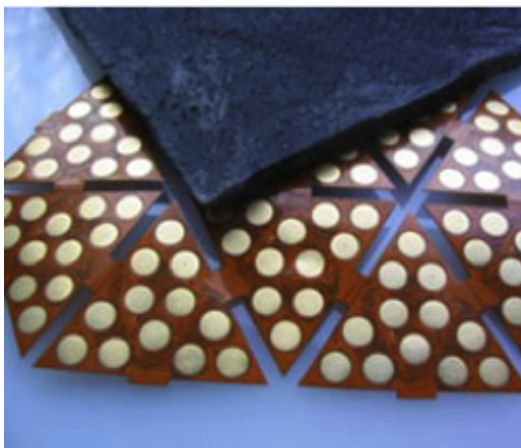
I was interested in how these sensors could be used to acquire data for object recognition. To get an insight into the data to expect and to be able to conceive representations and algorithms that exploit this data, a grasping experiment was conducted with a set of different objects. The objects varied in shape and compliance. The objects were manually placed in the (open) robot hand to execute multiple grasps. Pressure feedback was used to control these grasps so that each finger could be moved independently until a certain threshold was reached. Outputs of the tactile sensors as well as joint configurations were recorded. Due to the size of the objects, only index and middle fingers were regarded. Thumb, ring and middle fingertips did not touch the objects consistently. In Figure 50a a typical grasp is depicted.

Four objects were tested: a soft foam ball, a hard plastic ball, a porcelain cup, and a metal can. The objects were chosen so that there are two sets of objects with the same or a similar shape but one being compliant and other one being stiff. Figure 51 shows a visualization of the sensor outputs during grasping for the four objects. Each row in the plots on the right shows the measurements of one sensor. The y-axis represents time (in frames) and color corresponds to pressure intensities. Reading these plots from left to right we see at what point in time the fingers get in contact with the object. The sensors are distributed across the fingertips, with some being located at the sides and the tip as well. Since these sensors are unlikely to contact the object and should not return any values, only a subset of rows in the plots is expected to highlight. When looking at the results of the two balls, we see that during grasping the hard ball (green) higher pressure intensities were measured than for the soft ball (blue). This is caused by the abrupt stop of the finger on the rigid surface. The results of the can and the cup are not as clear since the can is made from conductive metal which leads to higher sensor responses in general. This is the case with all conductive materials.

Another interesting finding is that similar object shapes result in similar sensor activation patterns. We see that the finger configuration when grasping the balls must have been very similar since almost exactly the same sensors fire in both cases. For the two cylindrical objects, however, the pattern looks very different. It can easily be distinguished from the one of the balls. Even though the shape of the can and the cup do differ to some extent, they are still grasped the same way (i.e., same contact of index and middle finger).



(a) The iCub grasping a toy octopus

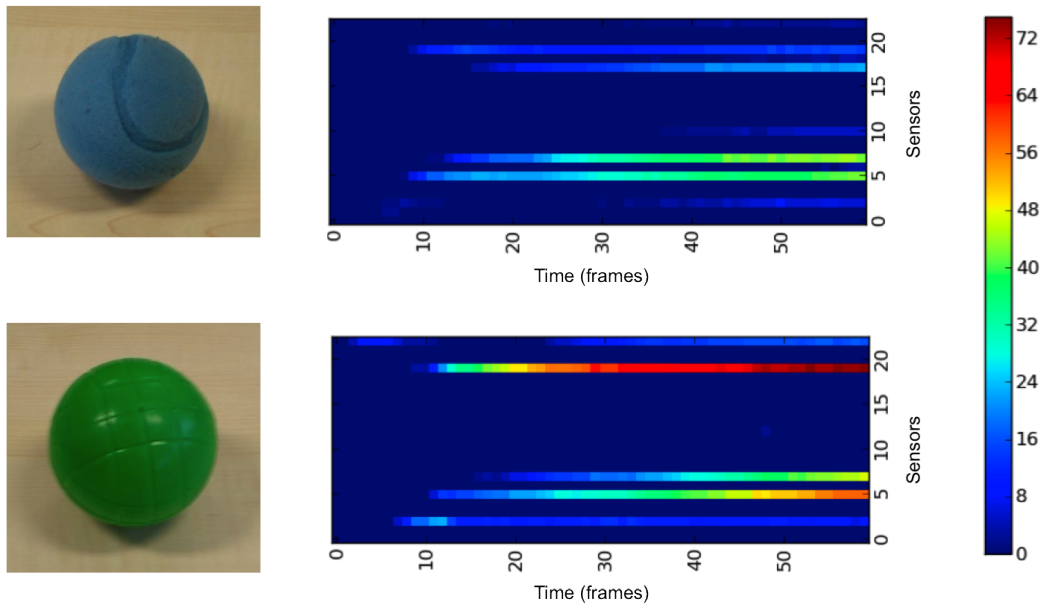


(b) Skin

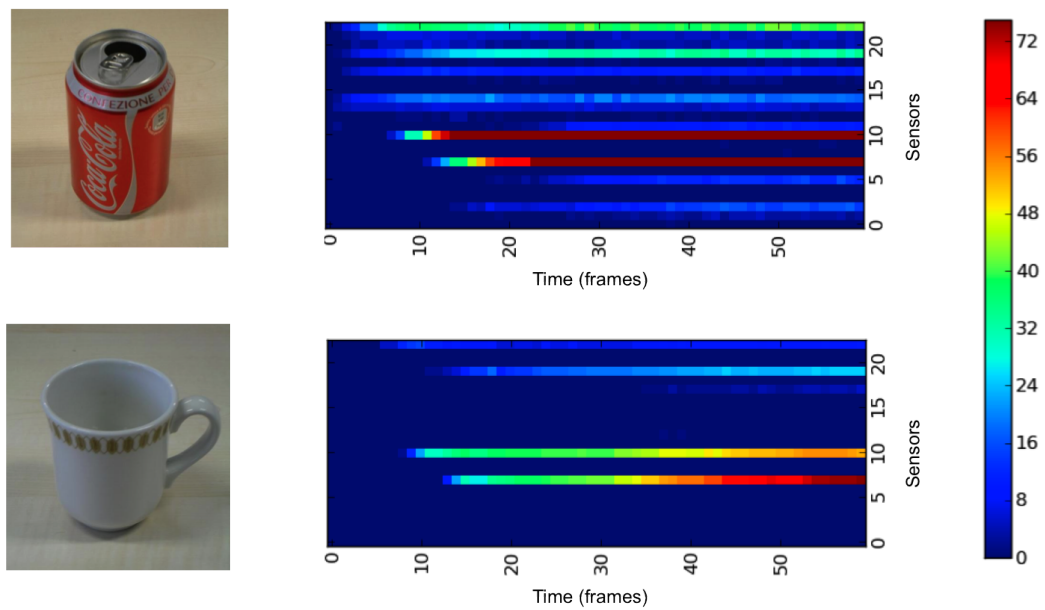


(c) Fingertips

Figure 50: Experimental tactile fingertips on the iCub.



(a) Soft vs. hard ball



(b) Compliant can vs. stiff cup

Figure 51: Time series showing intensity values read from tactile sensors during object grasping. Each row represents one sensor (12 sensors per finger).

Joint angles were recorded during grasping. Soft objects should lead to smooth trajectories as fingers do not stop abruptly but come to a gradual halt as the pressure increases. Furthermore, compliant objects should also lead to a tighter grip than rigid ones. Recorded joint angle values are plotted in Figure 52. For each finger, the angles of the first, second and third joint were summed to obtain one value corresponding to the position of the end-effector (i.e., the fingertip). Compare, for example, the red curve (toy octopus) with the green curve (hard ball). The red curve smoothly approaches the maximum value, whereas the other one first increases constantly and then stops abruptly.

The experiments presented here are, as mentioned above, intended only to point into further directions and to study if these cues could be exploited in an object recognition framework. The results are preliminary, yet they are very encouraging. I draw the following conclusions:

- Compliant objects can be distinguished from rigid object. It might be difficult to precisely measure compliance, however, discriminating between hard and soft is certainly feasible.
- The basic shape can be inferred and size can be estimated.

Recognizing specific objects as well as categorization is challenging if only based on haptic cues. However, haptic information can be incorporated into a more general recognition framework—as for example the one presented in Section 3.3. Here, this information can function as a prior narrowing down the list of possible object candidates. By weighting object hypotheses regarding these haptic inputs, probabilities can be shifted in favor of objects that agree with the data received.

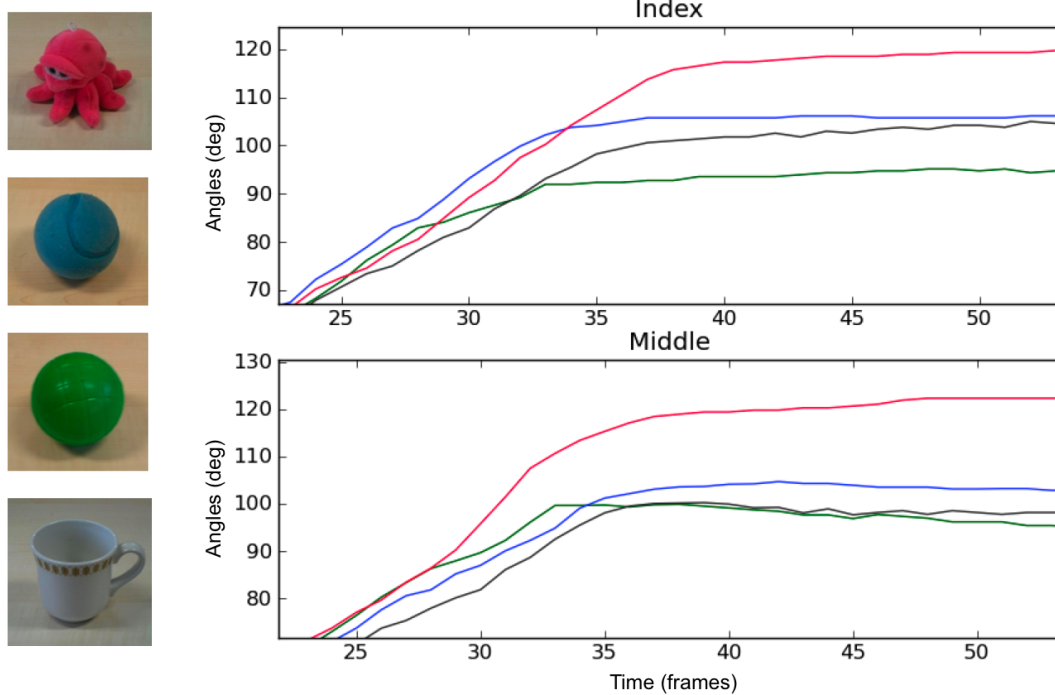


Figure 52: Joint angles for the index and middle finger during grasping. For each finger, all three joints are summed up. Line colors correspond with object colors. Octopus: pink, soft ball: blue, hard ball: green, cup: grey.

4.2 GENERAL CONCLUSIONS

To conclude the presented work, I will highlight some of the most important findings and results:

- For recognition and classification the most important ingredient is the quality and diversity of the input data. There is no 'right' sensor or modality. Different objects and different object attributes are perceived with different sensors.
- Availability of evaluation frameworks for multimodal data is still very low. Hopefully, the contributed dataset provides the community with a useful tool to extend research in multimodal object perception.
- Sensor fusion needs to be done in a probabilistic way. Hard choices and heuristics are difficult to specify and optimize. Parameters for sensor combination need to be inferred from data.
- Combining multiple predictions by incorporating proprioception lead to more robust and reliable results. Linking observations can resolve ambiguity that could not be resolved by regarding observations only individually.
- Motion planning enables an agent to directly seek out relevant information. This shortens the recognition sequence. In many cases this is even the only way to acquire the relevant information. Very similar objects can only be distinguished by observing the discriminating element. This may not be discovered with random exploration in adequate time.
- Haptic robot perception is still a largely unresolved topic. Tactile hardware is still at an experimental stage. I believe that haptic cues are a valuable source of object information and hope that future robotic systems will be able to exploit these.

The final goal of creating cognitive robots remains an open task for generations of research. I presented ideas and solutions how objects can be perceived by a robotic system exploiting multimodal information. As great a challenge building an intelligent agent may be, I hope that this work has contributed to moving us a small step closer to this goal.

BIBLIOGRAPHY

- [Agarwal 04] Shivani Agarwal, Aatif Awan & Dan Roth. *Learning to detect objects in images via a sparse, part-based representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 11, pages 1475–90, November 2004. (Cited on page 19.)
- [Alexe 10] Bogdan Alexe, Thomas Deselaers & Vittorio Ferrari. *What is an object?* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 73–80, June 2010. (Cited on page 6.)
- [Aloimonos 88] J Aloimonos, Isaac Weiss & A Bandyopadhyay. *Active vision*. International Journal of Computer Vision, vol. 356, pages 333–356, 1988. (Cited on pages 8 and 45.)
- [Andreopoulos 11] Alexander Andreopoulos, Stephan Hasler, Heiko Wersing, Herbert Janssen, John K. Tsotsos & Edgar Korner. *Active 3D Object Localization Using a Humanoid Robot*. IEEE Transactions on Robotics, vol. 27, no. 1, pages 47–64, February 2011. (Cited on page 46.)
- [Bailey 06] Tim Bailey & Hugh Durrant-whyte. *Simultaneous Localization and Mapping (SLAM): PartII*. IEEE Robotics & Automation Magazine, no. September, pages 108–117, 2006. (Cited on page 18.)
- [Bajcsy 88] R. Bajcsy. *Active perception*. Proceedings of the IEEE, vol. 76, no. 8, pages 966–1005, May 1988. (Cited on page 45.)
- [Ballard 91] D.H. Ballard. *Animate vision*. Artificial intelligence, vol. 48, no. 1, pages 57–86, 1991. (Cited on page 45.)
- [Bay 08] H. Bay, A. Ess, T. Tuytelaars & L. Van Gool. *Speeded-up robust features (SURF)*. Computer Vision and Image Understanding, vol. 110, no. 3, pages 346–359, 2008. (Cited on pages 18, 25, 26, and 27.)
- [Belongie 02] S. Belongie, J. Malik & J. Puzicha. *Shape matching and object recognition using shape contexts*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pages 509–522, April 2002. (Cited on page 30.)
- [Besl 85] P J Besl & R Jain. *Three-dimensional object recognition*. In ACM Computing Surveys (CSUR), volume 17, pages 75—145, 1985. (Cited on page 13.)

- [Biederman 87] I Biederman. *Recognition-by-components: a theory of human image understanding*. *Psychological review*, vol. 94, no. 2, pages 115–47, April 1987. (Cited on page 6.)
- [Bilmes 98] J.A. Bilmes. *A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models*. *International Computer Science Institute*, vol. 4, no. 510, page 126, 1998. (Cited on page 49.)
- [Bishop 06] CM Bishop. *Pattern recognition and machine learning*, volume 1. Springer New York, 2006. (Cited on page 7.)
- [Boiman 08] O. Boiman, E. Shechtman & M. Irani. *In defense of nearest-neighbor based image classification*. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. (Cited on page 33.)
- [Borenstein 97] J. Borenstein, H. R. Everett, L. Feng & D. Wehe. *Mobile robot positioning: Sensors and techniques*. *Journal of Robotic Systems*, vol. 14, no. 4, pages 231–249, April 1997. (Cited on page 13.)
- [Bosch 07] Anna Bosch, Andrew Zisserman & X. Munoz. *Representing shape with a spatial pyramid kernel*. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, pages 401–408. ACM, 2007. (Cited on pages 25, 27, and 50.)
- [Bosch 08] Anna Bosch, Andrew Zisserman & Xavier Munoz. *Image classification using rois and multiple kernel learning*. *International Journal of Computer Vision*, vol. 2008, pages 1–25, 2008. (Cited on page 89.)
- [Boykov 04] Yuri Boykov & Vladimir Kolmogorov. *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pages 1124–37, September 2004. (Cited on page 50.)
- [Bradski 00] G Bradski. *The OpenCV Library*. *Dr. Dobb's Journal of Software Tools*, 2000. (Cited on pages 9 and 50.)
- [Broz 09] Frank Broz, H. Kose-Bagci, C.L. Nehaniv & Kerstin Dautenhahn. *Learning behavior for a social interaction game with a child-like humanoid robot*. In *Social Learning in Interactive Scenarios Workshop, Humanoids 2009*, 2009. (Cited on page 48.)
- [Bülthoff 92] H H Bülthoff & S Edelman. *Psychophysical support for a two-dimensional view interpolation theory of object recognition*. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, no. 1, pages 60–4, January 1992. (Cited on page 6.)

- [Bülthoff 03] Isabelle Bülthoff & Heinrich H Bülthoff. *Image-based recognition of biological motion, scenes, and objects*. Analytic and holistic processes in the perception of faces, objects, and scenes, pages 146–176, 2003. (Cited on pages vii and 5.)
- [Burgard 97] W Burgard, Dieter Fox & S Thrun. *Active mobile robot localization by entropy minimization*. In Proceedings Second EUROMICRO Workshop on Advanced Mobile Robots, pages 155–162. IEEE Computer Society, 1997. (Cited on page 8.)
- [Bustos 06] Benjamin Bustos, Daniel Keim, Dietmar Saupe, Tobias Schreck & Dejan Vranić. *An experimental effectiveness comparison of methods for 3D similarity search*. International Journal on Digital Libraries, vol. 6, no. 1, pages 39–54, February 2006. (Cited on page 14.)
- [Callari 01] FG Callari & FP Ferrie. *Active object recognition: Looking for differences*. International Journal of Computer Vision, vol. 43, no. 3, pages 189–204, 2001. (Cited on page 46.)
- [Campbell 01] R Campbell. *A Survey Of Free-Form Object Representation and Recognition Techniques*. Computer Vision and Image Understanding, vol. 81, no. 2, pages 166–210, February 2001. (Cited on page 14.)
- [Cheng 11] MM Cheng & GX Zhang. *Global contrast based salient region detection*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 409–416. IEEE, June 2011. (Cited on page 6.)
- [Dalal 05] N. Dalal & B. Triggs. *Histograms of Oriented Gradients for Human Detection*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 886–893. Ieee, 2005. (Cited on page 27.)
- [Dempster 77] Arthur P Dempster, Nan M Laird & Donald B Rubin. *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society. Series B (Methodological), pages 1–38, 1977. (Cited on page 50.)
- [Denzler 02] Joachim Denzler & Christopher M Brown. *Information theoretic sensor data selection for active object recognition and state estimation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pages 1–13, 2002. (Cited on page 46.)
- [Dickinson 97] Sven J Dickinson, Henrik I Christensen, John K Tsotsos & Göran Olofsson. *Active Object Recognition Integrating Attention and Viewpoint Control*. Computer Vision and Image Un-

derstanding, vol. 67, no. 3, pages 239–260, September 1997. (Cited on page 45.)

- [Duda 72] Richard O. Duda & Peter E. Hart. *Use of the Hough transformation to detect lines and curves in pictures*. Communications of the ACM, vol. 15, no. 1, pages 11–15, January 1972. (Cited on page 42.)
- [Durrant-Whyte 06] H Durrant-Whyte & T Bailey. *Simultaneous localization and mapping: part I*. IEEE Robotics & Automation Magazine, 2006. (Cited on page 18.)
- [Dutta Roy 04] Sumantra Dutta Roy, Santanu Chaudhury & Subhashis Banerjee. *Active recognition through next view planning: a survey*. Pattern Recognition, vol. 37, no. 3, pages 429–446, March 2004. (Cited on page 8.)
- [Everingham 09] Mark Everingham, Luc Gool, Christopher K. I. Williams, John Winn & Andrew Zisserman. *The Pascal Visual Object Classes (VOC) Challenge*. International Journal of Computer Vision, vol. 88, no. 2, pages 303–338, September 2009. (Cited on page 19.)
- [Fairfield 08] N. Fairfield & D. Wettergreen. *Active localization on the ocean floor with multibeam sonar*. In OCEANS 2008, pages 1–10. IEEE, 2008. (Cited on pages 8 and 71.)
- [Faugeras 86] O.D. Faugeras & M. Hebert. *The Representation, Recognition, and Locating of 3-D Objects*. The International Journal of Robotics Research, vol. 5, no. 3, pages 27–52, September 1986. (Cited on page 14.)
- [Fei-Fei 04] Li Fei-Fei, R Fergus & P Perona. *Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories*. In IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04., volume 106, pages 59–70, April 2004. (Cited on page 19.)
- [Fei-Fei 05] L Fei-Fei & P. Perona. *A bayesian hierarchical model for learning natural scene categories*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 524–531. IEEE, 2005. (Cited on page 26.)
- [Ferrari 06] Vittorio Ferrari, Tinne Tuytelaars & L. Van Gool. *Object detection by contour segment networks*. In Computer Vision–ECCV 2006, volume 3953, pages 14–28. Springer, 2006. (Cited on page 19.)

- [Ferrari 08] V Ferrari, L Fevrier, F Jurie & C Schmid. *Groups of adjacent contour segments for object detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 1, pages 36–51, January 2008. (Cited on page 19.)
- [Fidler 08] Sanja Fidler, Marko Boben & Ales Leonardis. *Similarity-based cross-layered hierarchical representation for object categorization*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8. Ieee, June 2008. (Cited on page 89.)
- [Fischer 10] Jan Fischer, Daniel Seitz & Alexander Verl. *Face Detection using 3-D Time-of-Flight and Colour Cameras*. In 1st International Symposium on Robotics (ISR) and 2010 6th German Conference on Robotics (ROBOTIK), pages 112–116. VDE VERLAG GmbH, 2010. (Cited on page 24.)
- [Fischler 81] Martin a. Fischler & Robert C. Bolles. *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. Communications of the ACM, vol. 24, no. 6, pages 381–395, June 1981. (Cited on page 42.)
- [Fitzpatrick 03] Paul Fitzpatrick & Giorgio Metta. *Grounding vision through experimental manipulation*. Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, vol. 361, no. 1811, pages 2165—2185, 2003. (Cited on page 6.)
- [Foissotte 08] T. Foissotte, O. Stasse, a. Escande & a. Kheddar. *A next-best-view algorithm for autonomous 3D object modeling by a humanoid robot*. Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots, pages 333–338, December 2008. (Cited on page 46.)
- [Forsyth 96] DA Forsyth, Jitendra Malik, MM Fleck & Hayit Greenspan. *Finding pictures of objects in large collections of images*. Springer Berlin Heidelberg, 1996. (Cited on page 6.)
- [Frome 04] A. Frome, D. Huber & R. Kolluri. *Recognizing objects in range data using regional point descriptors*. In Computer Vision-ECCV 2004, pages 224–237. Springer, 2004. (Cited on page 14.)
- [Funkhouser 03] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin & David Jacobs. *A search engine for 3D models*. ACM Transactions on Graphics, vol. 22, no. 1, pages 83–105, January 2003. (Cited on page 14.)
- [Gao 07] Dashan Gao & Nuno Vasconcelos. *Bottom-up saliency is a discriminant process*. 2007 IEEE 11th International Conference on Computer Vision, pages 1–6, 2007. (Cited on page 6.)

- [Gibson 66] J.J. Gibson. The senses considered as perceptual systems. Houghton Mifflin, 1966. (Cited on page 5.)
- [Gordon 93] N.J. Gordon, D.J. Salmond & A.F.M. Smith. *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*. IEE Proceedings F (Radar and Signal Processing), vol. 140, no. 2, pages 107–113, 1993. (Cited on pages 67 and 68.)
- [Griffin 07] Gregory Griffin, Alex Holub & Pietro Perona. *Caltech-256 object category dataset*. Rapport technique, California Institute of Technology, March 2007. (Cited on page 19.)
- [Gross 05] R. Gross, S Baker, I Matthews & T Kanade. *Face recognition across pose and illumination*. Handbook of Face Recognition, pages 197–222, 2005. (Cited on page 19.)
- [Guennebaud 10] Gaël Guennebaud, Benoît Jacob & Others. *Eigen v3*. <http://eigen.tuxfamily.org>, 2010. (Cited on page 9.)
- [Haar 10] Alfred Haar. *Zur Theorie der Orthogonalen Funktionensysteme*. Mathematische Annalen, vol. 69, no. 3, pages 331–371, 1910. (Cited on page 27.)
- [Heczko 02] M Heczko, D Keim & D Saupe. *Methods for similarity search on 3D databases*. Datenbank-Spektrum, 2002. (Cited on pages 25 and 30.)
- [Heitz 08] Jeremy Heitz & Daphne Koller. *Learning spatial context: Using stuff to find things*. In Computer Vision–ECCV 2008, pages 30–43. Springer, 2008. (Cited on page 6.)
- [Hetzl 01] G. Hetzel, B. Leibe, P. Levi & B. Schiele. *3D object recognition from range images using local feature histograms*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 394–399. IEEE, 2001. (Cited on page 14.)
- [Horn 84] Berthold K P Horn. *Extended Gaussian Images*. Proceedings of the IEEE 72, pages 1671–1686, 1984. (Cited on page 14.)
- [Hou 07] X. Hou & L. Zhang. *Saliency detection: A spectral residual approach*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2007. (Cited on page 6.)
- [Huber 04] D Huber, A. Kapuria, R. Donamukkala & M. Hebert. *Parts-based 3d object classification*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2. IEEE, 2004. (Cited on page 14.)
- [Itti 98] L Itti, C Koch & E Niebur. *A model of saliency-based visual attention for rapid scene analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 11, pages 1254–1259, 1998. (Cited on page 6.)

- [Johnson 99] A. E Johnson & M. Hebert. *Using spin images for efficient object recognition in cluttered 3D scenes*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, page 433, 1999. (Cited on page 14.)
- [Kazhdan 03] Michael Kazhdan, Thomas Funkhouser & Szymon Rusinkiewicz. *Rotation invariant spherical harmonic representation of 3D shape descriptors*. In Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, pages 156–164. Eurographics Association, 2003. (Cited on page 14.)
- [Koenderink 92] Jan J Koenderink & Andrea J van Doorn. *Surface shape and curvature scales*. Image and Vision Computing, vol. 10, no. 8, pages 557–564, October 1992. (Cited on pages 25 and 29.)
- [Körtgen 03] M Körtgen, GJ Park, M. Novotni & R. Klein. *3D shape matching with 3D shape contexts*. In The 7th Central European Seminar on Computer Graphics, volume 3. Citeseer, 2003. (Cited on pages 25 and 30.)
- [Kraft 08] Dirk Kraft, Nicolas Pugeault, Emre Başeski, Mila Popović, Danica Kragić, Sinan Kalkan, Florentin Wörgötter & Norbert Krüger. *Birth of the Object: Detection of Objectness and Extraction of Object Shape Through Object–Action Complexes*. International Journal of Humanoid Robotics, vol. 05, no. 02, page 247, 2008. (Cited on page 6.)
- [Krainin 11] Michael Krainin, Brian Curless & Dieter Fox. *Autonomous generation of complete 3D object models using next best view manipulation planning*. 2011 IEEE International Conference on Robotics and Automation, pages 5031–5037, May 2011. (Cited on page 46.)
- [Kümmerle 08] Rainer Kümmerle & Rudolph Triebel. *Monte Carlo localization in outdoor terrains using multilevel surface maps*. Journal of Field Robotics, vol. 25, no. 6-7, pages 346—359, 2008. (Cited on pages 8 and 18.)
- [Lai 11] Kevin Lai, Liefeng Bo, Xiaofeng Ren & Dieter Fox. *A Large-Scale Hierarchical Multi-View RGB-D Object Dataset*. In IEEE International Conference on Robotics and Automation (ICRA), pages 1817—1824, 2011. (Cited on pages 21, 22, and 33.)
- [Lederman 09] SJ Lederman & RL Klatzky. *Haptic perception: A tutorial*. Attention, Perception, & Psychophysics, vol. 71, no. 7, pages 1439–1459, 2009. (Cited on page 7.)
- [Leibe 04] Bastian Leibe, Ales Leonardis & Bernt Schiele. *Combined object categorization and segmentation with an implicit shape model*.

In ECCV'04 Workshop on Statistical Learning in Computer Vision, número May, pages 1–16, 2004. (Cited on page 6.)

[Li 09] LJ Li, Richard Socher & L Fei-Fei. *Towards total scene understanding: Classification, annotation and segmentation in an automatic framework*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2036—2043. IEEE, 2009. (Cited on page 6.)

[Lindeberg 98] Tony Lindeberg. *Feature detection with automatic scale selection*. International Journal of Computer Vision, vol. 30, no. 2, pages 79–116, 1998. (Cited on page 26.)

[Liu 07] T Liu, J Sun & NN Zheng. *Learning to detect a salient object*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1—8, 2007. (Cited on page 6.)

[Lowe 99] D.G. Lowe. *Object recognition from local scale-invariant features*. In Proceedings of the Seventh IEEE International Conference on Computer Vision, pages 1150–1157 vol.2. IEEE, 1999. (Cited on pages 18, 26, 27, and 50.)

[MacQueen 67] J MacQueen. *Some methods for classification and analysis of multivariate observations*. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1, pages 281–297, 1967. (Cited on page 74.)

[Metta 06] Giorgio Metta, Paul Fitzpatrick & Lorenzo Natale. *YARP: Yet Another Robot Platform*. International Journal of Advanced Robotic Systems, 2006. (Cited on page 9.)

[Mikolajczyk 01] K Mikolajczyk & Cordelia Schmid. *Indexing based on scale invariant interest points*. In Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, volume 1, pages 525—531, 2001. (Cited on page 26.)

[Montesano 08] Luis Montesano, Manuel Lopes, Alexandre Bernardino & José Santos-Victor. *Learning Object Affordances: From Sensory–Motor Coordination to Imitation*. IEEE Transactions on Robotics, vol. 24, no. 1, pages 15–26, February 2008. (Cited on page 6.)

[Murase 95] Hiroshi Murase & SK Nayar. *Visual learning and recognition of 3-D objects from appearance*. International Journal of Computer Vision, vol. 24, pages 5–24, 1995. (Cited on page 46.)

[Omrcen 07] Damir Omrcen, Ales Ude, Kai Welke, Tamim Asfour & Rüdiger Dillmann. *Sensorimotor processes for learning object representations*. In 7th IEEE-RAS International Conference on Humanoid Robots, pages 143–150. IEEE, 2007. (Cited on page 46.)

- [Opelt 04] A Opelt & M Fussenegger. *Weak hypotheses and boosting for generic object detection and recognition*. In *Computer Vision-ECCV 2004*, 2004. (Cited on page 19.)
- [Osada 01] RobertR Osada, ThomasT Funkhouser, Bernard Chazelle & David Dobkin. *Matching 3D models with shape distributions*. In *Proc. SMI 2001 International Conference on Shape Modeling and Applications*, pages 154–166, May 2001. (Cited on page 29.)
- [Osada 02] Robert Osada, Thomas Funkhouser, Bernard Chazelle & David Dobkin. *Shape distributions*. *ACM Transactions on Graphics*, vol. 21, no. 4, pages 807–832, October 2002. (Cited on pages 14, 25, and 29.)
- [Paletta 00] Lucas Paletta & Axel Pinz. *Active object recognition by view integration and reinforcement learning*. *Robotics and Autonomous Systems*, vol. 31, no. 1, pages 71—86, 2000. (Cited on page 46.)
- [Parlitz 08] Christopher Parlitz, M. Hägele, P. Klein, J. Seifert & K. Dautenhahn. *Care-O-bot 3-Rationale for human-robot interaction design*. In *Proceedings of 39th International Symposium on Robotics (ISR)*, pages 275–280, 2008. (Cited on page 13.)
- [Pattacini 10a] Ugo Pattacini. *Modular cartesian controllers for humanoid robots: Design and implementation on the iCub*. PhD thesis, 2010. (Cited on page 47.)
- [Pattacini 10b] Ugo Pattacini, Francesco Nori, Lorenzo Natale, Giorgio Metta & Giulio Sandini. *An Experimental Evaluation of a Novel Minimum-Jerk Cartesian Controller for Humanoid Robots*. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1668–1674, 2010. (Cited on page 64.)
- [Quigley 09] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler & Andrew Y Ng. *ROS: an open-source Robot Operating System*. In *ICRA Workshop on Open Source Software*, 2009. (Cited on page 9.)
- [Reiser 09] Ulrich Reiser, Christian Connette, Jan Fischer, Jens Kubacki, Alexander Bubeck, Florian Weisshardt, Theo Jacobs, Christopher Parlitz, M. Hägele & Alexander Verl. *Care-O-bot® 3: creating a product vision for service robot applications by integrating design and technology*. In *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, pages 1992–1998. IEEE Press, 2009. (Cited on page 13.)
- [Ruiz-Correa 01] Salvador Ruiz-Correa, L.G. Shapiro & Marina Melia. *A new signature-based method for efficient 3-d object recognition*. In

- IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, page 769, 2001. (Cited on page 14.)
- [Ruiz-correa 03] Salvador Ruiz-correa, Linda G Shapiro & Marina Meil. *A new paradigm for recognizing 3-D objects from range data*. In Proceedings of the Ninth IEEE International Conference on Computer Vision, pages 1126–1133 vol.2. Ieee, 2003. (Cited on page 14.)
- [Rusu 11] Radu Bogdan Rusu & Steve Cousins. *3D is here: Point Cloud Library (PCL)*. In IEEE International Conference on Robotics and Automation (ICRA), pages 1–4. Ieee, May 2011. (Cited on page 9.)
- [Saidi 07] Francois Saidi & Olivier Stasse. *Online object search with a humanoid robot*. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1677—1682, 2007. (Cited on page 46.)
- [Sandini 07] G. Sandini, G. Metta & D. Vernon. *The iCub cognitive humanoid robot: An open-system research platform for enactive cognition*. In 50 years of artificial intelligence, pages 358–369. Springer, 2007. (Cited on page 48.)
- [Schaal 01] Stefan Schaal. *The SL simulation and real-time control software package*. Rapport technique, University of Southern California, 2001. (Cited on page 51.)
- [Schaal 07] Stefan Schaal. *The New Robotics-towards human-centered machines*. HFSP journal, vol. 1, no. 2, pages 115–26, July 2007. (Cited on pages vii and 2.)
- [Schiebener 11] David Schiebener, Ales Ude, Jun Morimoto, Tamim Asfour & Rüdiger Dillmann. *Segmentation and learning of unknown objects through physical interaction*. 2011 11th IEEE-RAS International Conference on Humanoid Robots, pages 500–506, October 2011. (Cited on page 6.)
- [Schmitz 10] A Schmitz, M Maggiali, L Natale, B Bonino & G Metta. *A tactile sensor for the fingertips of the humanoid robot iCub*. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2212–2217. Ieee, October 2010. (Cited on page 90.)
- [Schölkopf 02] Bernhard Schölkopf & Alex Smola. *Learning with kernels*, 2002. (Cited on page 7.)
- [Shechtman 07] Eli Shechtman & Michal Irani. *Matching local self-similarities across images and videos*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8. IEEE, June 2007. (Cited on pages 25 and 27.)

- [Shilane 05] P. Shilane, P. Min, M. Kazhdan & T. Funkhouser. *The princeton shape benchmark*. In *Shape Modeling Applications*, 2004. Proceedings, volume 08540, pages 167–178. IEEE, 2005. (Cited on pages 14 and 21.)
- [Siciliano 08] Bruno Siciliano & Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer, 2008. (Cited on page 46.)
- [Steinhaus 56] H Steinhaus. *Sur la division des corp materiels en parties*. Bull. Acad. Polon. Sci, vol. 1, pages 801–804, 1956. (Cited on page 74.)
- [Sun 10] Min Sun, Gary Bradski, BX Xu & Silvio Savarese. *Depth-encoded hough voting for joint object detection and shape recovery*. In *ECCV 2010*, pages 1–14, 2010. (Cited on page 21.)
- [Thrun 05] Sebastian Thrun, Wolfram Burgard & Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. (Cited on pages 8 and 13.)
- [Tikhanoff 08] V Tikhanoff, P Fitzpatrick & Francesco Nori. *The icub humanoid robot simulator*. In *IROS Workshop on Robot Simulators*, volume 1, 2008. (Cited on pages 57 and 76.)
- [Tikhanoff 11] V. Tikhanoff, A. Cangelosi & G. Metta. *Integration of Speech and Action in Humanoid Robots: iCub Simulation Experiments*. *Autonomous Mental Development*, IEEE Transactions on, vol. 3, no. 1, pages 17–29, 2011. (Cited on page 48.)
- [Torralba 04] A. Torralba, K.P. Murphy & W.T. Freeman. *Sharing features: efficient boosting procedures for multiclass object detection*. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 762–769. IEEE, 2004. (Cited on page 19.)
- [Triebel 07] Rudolph Triebel, Richard Schmidt, Ó.M. Mozos & W. Burgard. *Instance-based AMN classification for improved object recognition in 2D and 3D laser range data*. In *Proceedings of the 20th international joint conference on Artificial intelligence*, pages 2225–2230. Morgan Kaufmann Publishers Inc., 2007. (Cited on page 18.)
- [Viola 01] P. Viola & M. Jones. *Rapid object detection using a boosted cascade of simple features*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pages 1–511–1–518, 2001. (Cited on page 26.)
- [Wallraven 07] C Wallraven & H H Bülthoff. *Object Recognition in Humans and Machines*, chapitre Object Rec, pages 89–104. Springer, Tokyo, Japan, 2007. (Cited on page 51.)

- [Welke 08] Kai Welke, Tamim Asfour & Rüdiger Dillmann. *Object separation using active methods and multi-view representations*. In 2008 IEEE International Conference on Robotics and Automation, pages 949–955. IEEE, May 2008. (Cited on page 46.)
- [Welke 09] Kai Welke, Tamim Asfour & Rüdiger Dillmann. *Active multi-view object search on a humanoid head*. In 2009 IEEE International Conference on Robotics and Automation, pages 417–423. IEEE, May 2009. (Cited on page 46.)
- [Welke 10] Kai Welke, Jan Issac, David Schiebener, Tamim Asfour & Rüdiger Dillmann. *Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot*. 2010 IEEE International Conference on Robotics and Automation, pages 2012–2019, May 2010. (Cited on page 46.)
- [Wilkes 94] D. Wilkes & J.K. Tsotsos. *Active object recognition*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 136–141. IEEE, 1994. (Cited on page 45.)
- [Wyszecki 68] Günter Wyszecki, VS Stiles & Kenneth L Kelly. *Color Science: Concepts and Methods, Quantitative Data and Formulas*. Physics Today, vol. 21, no. 6, pages 83—84, 1968. (Cited on page 29.)

PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

B. Browatzki, V. Tikhanoff, G. Metta, H.H. Bühlhoff, and C. Wallraven, "Active Object Recognition on a Humanoid Robot", *IEEE International Conference on Robotics and Automation (ICRA2012)*, May 2012

B. Browatzki, J. Fischer, B. Graf, H.H. Bühlhoff, and C. Wallraven, "Going into depth: Evaluating 2D and 3D cues for object classification on a new, large-scale object dataset", *1st ICCV Workshop on Consumer Depth Cameras in Computer Vision (CD4CV2011)*, November 2011

B. Browatzki, "Lernen und Erkennen von 3D-Objekten durch Kombination visueller und propriozeptiver Information", Januar 2009

ACKNOWLEDGMENTS

First, I want to thank my supervisor Christian Wallraven. We have been working together since I started my diploma thesis at the MPI. In all this time he continuously supported me and my research. He provided many valuable ideas, insightful discussions and answered countless questions. This thesis would not have been possible without him.

I want to thank Prof. Heinrich H. Bülthoff who gave me the opportunity to carry out this PhD project in his department at the Max Planck Institute for Biological Cybernetics. I am grateful that I could spend time in this international and multidisciplinary scientific environment in the company of so many bright and friendly people. I am also very thankful to Prof. Alexander Verl for supervising this thesis at the University of Stuttgart. Further, I would like to thank the Bernstein Center for Integrative Neuroscience and the Poeticon EU-project for funding parts of my work.

I would like to thank my collaborators on the iCub project in Italy, Vadim Tikhanoff and Giorgio Metta. Grazie mille! They always took care of me in Italy and made sure I would not get lost in Genoa or the IIT. I also thank my collaborators on the Care-O-bot project in Stuttgart, Jan Fischer and Birgit Graf. I always enjoyed visiting the Fraunhofer IPA.

I am glad I could share an office with Janina Esins and Christian Herdtweck. They created a very friendly working atmosphere and were a source of constant help and motivation.

Finally, I thank my parents for their never ceasing support throughout the last three decades.

CURRICULUM VITAE

PERSONAL INFORMATION

Name Björn Browatzki
Born 30th of June 1983, Herrenberg, Germany
Email bjoern@browatzki.net

EDUCATION

Since March 2009 Ph.D. thesis at Max Planck Institute for Biological Cybernetics in Tübingen, Germany
June 2008 – Feb. 2009 Diploma thesis at Max Planck Institute for Biological Cybernetics in Tübingen, Germany
2002 – 2009 Software Engineering (Dipl. Inf.), University of Stuttgart
1993 – 2002 Abitur at Schickhardt Gymnasium Herrenberg, Germany
1989 – 1993 Grund- und Hauptschule Bondorf, Germany