

Video based Object Retrieval and Recognition using Lightweight Devices

DOI:10.18136/PE.2018.680

A Thesis Submitted for the Degree of Doctor
of Philosophy in Computer Science

Metwally Rashad Metwally Omar Alseedy

Supervisor: Dr. Laszló Czúni



Department of Electrical Engineering and Information Systems

Doctoral School of Information Science and Technology

University of Pannonia

Veszprém, Hungary

2018

VIDEO BASED OBJECT RETRIEVAL AND RECOGNITION USING
LIGHTWEIGHT DEVICES

Thesis for obtaining a PhD degree in
the Doctoral School of Information Science and Technology
of the University of Pannonia.

Written by:
Metwally Rashad Metwally Omar Alseedy

Written in the Doctoral School of Information Science and Technology of the University
of Pannonia.

Supervisor: **Dr. László Czúni**
propose acceptance (yes / no)

.....
(signature)

The candidate has achieved % in the comprehensive exam,
Veszprém,

.....
Chairman of the Examination Committee

As reviewer, I propose acceptance of the thesis:

Name of reviewer: (yes / no)

.....
(signature)

Name of reviewer: (yes / no)

.....
(signature)

The candidate has achieved % at the public discussion.

Veszprém,

.....
Chairman of the Committee

Grade of the PhD diploma

.....
President of UCDH

Acknowledgments

First of all, All gratitude and thankfulness to ALLAH for guiding and aiding me to bring this work out to light. Also, it is impossible to give sufficient thanks to the people who gave help and advice during the writing of this thesis.

I am deeply indebted to my supervisor Dr. *László Czúni* for his kindness and emotional support throughout the period it took to prepare this work in its final form. I am particularly grateful for his genuine concern and his prompt replies to all the questions I addressed to him, for giving so generously of his time in revising this work and, in the process, pointing out some relevant and interesting ideas.

I would like to thank the Director of the Doctoral School Prof. *Katalin Hangos* for her help and support, also I wish to express my deep gratitude to the staff of the Department of Electrical Engineering and Information Systems, Faculty of Information Technology University of Pannonia for their guidance and moral support during the making of this work.

I acknowledge the financial support of this work by the NKFI-6 fund through project K120369 (OTKA) and by Széchenyi 2020 under EFOP-3.6.1-16-2016-00015.

Many thanks for all my friends and colleagues in Veszprém, they played a very important role in my life.

The last, but the most important thanks to my family: my father, my mother, my wife and my daughters for their love, also for keeping me in their prayers.

Kivonat

Igen sok vizuális feladat lényegileg azon a képességen alapul, hogyan ismerjük fel konkrét tárgyakat, helyszíneket vagy objektum kategóriákat. A vizuális felismerésnek igen sokfajta lehetséges alkalmazása van - érintve a mesterséges intelligencia és információ visszakeresés területeit - mint a tartalom alapú képviszakeresés, videó adatbányászat, vagy objektum azonosítás mobile robotok által. A disszertáció számítógépes látás algoritmusokat mutat be a videó alapú specifikus objektum visszakeresés és felismerés területén.

Az érdeklődés központjában azok az alkalmazások állnak, ahol kis erőforrásigényű eszközöket használunk: olyan látórendszerek, ahol limitált a számítási kapacitás, az energiaforrás és az elérhető memória. A javasolt módszerek a specifikus objektumfelismerésre ún. gyenge osztályozókat használnak: mivel a 3D tárgyaknak nagyon eltérő lehet a kinézete különböző irányokból, a különböző nézeteket először egymástól függetlenül dolgozzuk fel, majd az előzetes eredményeket kombináljuk.

Olyan új eljárásokat mutatunk be, amelyek kamerák mellett inerciális mérőegységet (Inertial Measurement Unit - IMU) használnak a 3D objektumok visszakeresésére. A disszertációban gyors és robusztus kompakt képi leírók és relatív irány információk segítségével készítünk ún. "multi-view" központú objektum modelleket. Három megoldást mutatunk be a 3D tárgyak visszakeresésére és felismerésére.

Először egy olyan videó alapú visszakereső eljárást, amely kompakt képi leírókat használ KD-Fa indexeléssel, és a Hough transzformáció szolgál az egyes videó kockák-

on futtatott lekérdezések kiértékelésére.

Ezek után az IMU-k használatát ismertetjük, megmutatva, hogy ez előző eljárás jósági függvényét egy orientációs taggal kibővítve - érdemi komplexitás növekedés nélkül - javíthatjuk a találati arányt.

Végül egy új, Rejtett Markov Model alapú keretrendszert mutatunk be, ahol a 2D nézetek állapotoknak felelnek meg, a megfigyeléseket a kompakt él és szín érzékeny képi leírók jelentik, az orientációs szenzorok pedig az állapotok közti átmeneti valószínűségeket adják meg. A kis erőforrásigényű megközelítéseket különböző adathalmazokon teszteljük több ezer lekérdezés által, többféle összehasonlítás során. A kiértékelések eredményei azt mutatják, hogy a bemutatott módszerek magas találati arányt érnek el kicsi memória és számítási igény mellett.

Abstract

Many visual tasks fundamentally rely on the ability to recognize specific objects, scenes, or object categories. Visual recognition itself has a variety of potential applications that touch many areas of artificial intelligence and information retrieval, such as content-based image search, video data mining, or object identification for mobile robots. This dissertation presents computer vision algorithms for video-based object specific retrieval and recognition problems.

Our focus of interest is the application area where lightweight devices are used: computer vision systems that has limited computing power, energy resources, and memory. The proposed methods attack the problem of recognition of specific objects by the idea of using several weak classifiers: since 3D objects can have very different appearance from the different directions the combination of these views are to be processed independently and the results are combined to have the decision results. We introduce new object retrieval approaches where besides cameras, Inertial Measurement Unit (IMU) sensors are used for the retrieval of 3D objects. In the dissertation we use fast and robust compact image descriptors and the relative orientation of the camera to build multi-view-centered retrieval object models. The dissertation introduces three solutions for the 3D object retrieval and recognition problem.

First, a video-based 3D object retrieval method is presented, based on fast retrieval mechanisms with weak classifiers using compact image descriptors and KD-Tree in-

dexing, where the Hough transformation paradigm is used to evaluate the results of queries applied on several frames of a video.

Next, the dissertation introduces the usage of IMUs, showing that adding the orientation term to the fitness function of the previous method increases the retrieval rate without the cost of (significant) extra computations.

Finally, we present a new retrieval model using Markovian estimation mechanisms. We built a Hidden Markov Model (HMM) framework where 2D object views correspond to states, observations are coded by compact edge and color sensitive descriptors, and orientation sensors are used to secure temporal inference by estimating transition probabilities between states.

We analyze the performance of our lightweight approaches on several datasets and thousands of queries with different comparisons. Our evaluation results show that the presented approaches achieve high hit-rates with low memory and computation requirements.

Contents

Acknowledgments	iii
Kivonat	iv
Abstract	vi
List of Figures	xv
List of Tables	xvi
1 Introduction	1
1.1 The Need for New Recognition Methods	1
1.2 Coding Information for Multiple-views with Orientation Sensors	3
1.3 Object Model-centered or Appearance-based Approaches	4
1.4 Contributions	6
1.5 Thesis Organization	7
2 Problem Formulation and Preliminaries	9
2.1 Previous Works	9
2.2 General Problem Formulation	14
2.3 View-Centered Retrieval and Recognition	17
2.4 Visual Features	18
2.5 Feature Indexing Using KD-Tree	20

2.6	Image Similarity Measure	22
2.7	The Hough Transformation Paradigm	23
2.8	Hidden Markov Models	24
2.8.1	Discrete Density HMMs (DHMM)	26
2.8.2	Continuous Density HMMs (CHMM)	26
2.8.3	HMM Problems	27
2.9	Object Tracking	30
2.10	Summary	30
3	Datasets and Test Issues	32
3.1	Object Model Datasets	32
3.1.1	SUP-16 Dataset	33
3.1.2	SUP-25 Dataset	33
3.1.3	SMO Dataset	35
3.1.4	COIL-100 Dataset	35
3.1.5	ALOI Dataset	36
3.2	Query Datasets	36
3.3	Hardware and Software Issues	38
4	Weak Classifiers for Object Retrieval with the Hough Paradigm	40
4.1	EIS (Extensive Image Search) Approach	41
4.2	VCI Best Match Approach	42
4.2.1	Voting of Candidate using Indexing (VCI)	43
4.3	Experimental Evaluation	45
4.3.1	Retrieval Performance on SUP-16, COIL-100 and ALOI Datasets	45
4.4	Conclusion	46

5	Orientation Sensors for Object Retrieval and Recognition	50
5.1	IMU Sensors	50
5.2	Object Retrieval with IMU	52
5.2.1	SIIS (Selected Image and IMU Search) Approach	52
5.2.2	EIIS (Extensive Image and IMU Search) Approach	53
5.2.3	VCI Approach with IMU	54
5.2.4	Zero Weight Case	56
5.3	Object Recognition with IMU	56
5.4	Active Vision in the Hough Framework	57
5.4.1	Active Retrieval to Minimize Ambiguity	59
5.5	Experimental Evaluation	60
5.5.1	Retrieval Performance on SUP-16, COIL-100 and ALOI Datasets	61
5.5.2	Retrieval Performance on the SMO Dataset with Various Back- grounds	62
5.5.3	Recognition Performance with the Extended SUP-16 Dataset .	67
5.5.4	Retrieval Performance with Automatic Segmentation on the SUP-25 Dataset	68
5.5.5	Retrieval Performance with Active VCI on the COIL-100 Dataset	69
5.5.6	Running Times	70
5.6	Conclusion	71
6	View Centered Object Models using Hidden Markov	
	Model	74
6.1	Object Retrieval with HMM	75
6.1.1	Object Views as States in a Markov Model	75
6.1.2	State Transitions	76

6.1.3	Hidden States Approximated by Observations with Compact Descriptors	78
6.1.4	Decoding for Retrieval	79
6.2	Experimental Evaluation	79
6.2.1	Retrieval Performance on COIL-100, ALOI and SMO with Various Backgrounds	79
6.2.2	Running Time and Memory Requirements	80
6.3	Conclusion and Future Works	80
7	Conclusion	86
7.1	New Scientific Results	87
7.2	Publications	89
7.2.1	Publications related to this Thesis	89
7.2.2	Publication not related to this Thesis	90
	Bibliography	91

List of Figures

1.1	Real-life recognition examples.	3
2.1	Basic object recognition flowchart.	10
2.2	Model generation setup with target object in the centre.	18
3.1	Test model object examples from the SUP-16 dataset.	33
3.2	Test model object examples from the SUP-25 dataset.	34
3.3	Top row: examples for the results of tracking. Bottom row: objects and their environment.	34
3.4	Test object examples from the SMO dataset with uniform background.	35
3.5	Test model object examples from the COIL-100 dataset.	36
3.6	Test model object examples from the ALOI dataset.	37
3.7	Noisy and blurred query examples from the SUP-16 dataset.	38
3.8	Noisy and blurred query examples from the COIL-100 dataset.	38
3.9	Noisy and blurred query examples from the ALOI dataset.	38
3.10	Noisy and blurred query examples from the SMO dataset with uni- form background.	39
3.11	Noisy and blurred query examples from the SMO dataset with tex- tured background.	39

4.1	Illustration of the EIS approach: only the images of the query are compared to the images of the candidates independently. The similarity of the query sequence and candidates are based on the sum of Tanimoto Coefficients.	42
4.2	Illustration of the VCI approach with three retrieval lists ($N_f^q=3$). Since C_3 was not on L_2 but on L_1 and L_3 , $C_{3,4}$ was added based on $TC(q_i, c_j)$	44
4.3	The pseudo code for VCI search algorithm.	45
4.4	Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the SUP-16 database.	47
4.5	Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the COIL-100 database.	48
4.6	Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the ALOI database.	49
5.1	IMU measurement error distribution.	52
5.2	Illustration of the SIIS approach: after finding the best matching frame of a query and a candidate, other frames are also compared selected on the bases of their similar relative positions as the frames of the query.	54
5.3	Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the SUP-16 database and for VCI $\omega = 0.5$	63

5.4	Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the COIL-100 database and for VCI $\omega = 0.5$	64
5.5	Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the ALOI database and for VCI $\omega = 0.5$	65
5.6	Average hit-rate of the VCI method for motion blur (top) and additive Gaussian noise (bottom) at different query views (N) and w settings for the ALOI dataset.	66
5.7	Examples for untrained objects, taken from the SMO dataset, to test recognition performance.	68
5.8	Average hit-rate for the SUP-16 dataset with 9 extra untrained queries with different thresholds.	68
5.9	Average hit-rate obtained over the COIL-100 dataset with the VCI_Best_Match, VCI_W, VCI, AVCI, and with the “Best 10”: queries with motion blur (top); queries with additive Gaussian noise (bottom).	71
5.10	Average running time for EIS, SIIS, EIIS, VC and VCI approaches on a dataset of 100 objects.	72
6.1	Geometrical interpretation of transition probabilities.	77
6.2	Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the COIL-100 database.	81
6.3	Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the ALOI database.	82

-
- 6.4 Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the SMO database with uniform backgrounds. 83
- 6.5 Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the SMO database with textured backgrounds. 84

List of Tables

5.1	Complexity of different approaches at $N_c = 16$, $N_f^c = 50$, $N_f^{leaf} = 14$, and different N_f^q	56
5.2	Comparison of experimental setups of [12] and VCI approach.	66
5.3	Average hit-rate for distortion free (DF), motion blur (MB) and ad- ditive Gaussian noise (GN) with uniform (UB) and textured back- grounds (TB) on the SMO dataset.	67
5.4	Hit-rate of VCI with object tracking for SUP-25.	69
6.1	Running times in seconds for the retrieval of one object from 100.	80

Chapter 1

Introduction

1.1 The Need for New Recognition Methods

The rapid advances in computer techniques, graphics hardware, and networks have led to the wide application of 3D models in various domains, such as 3D graphics, computer aided design (CAD), the medical industry, robot systems, architecture design, and the civil engineering community. Therefore, efficient 3D object retrieval and recognition technologies are of great importance for many applications.

The humans can recognize objects easily without great efforts; on contrary the recognition by machines is one of the fundamental problems of computer vision. Since childhood, 3D object recognition is one of the most fascinating abilities that humans easily possess. Humans are often able to tell the identity or category despite of the variations of appearance due to change in pose, texture, illumination, deformation, or occlusion with a simple glance on a 3D object. Furthermore, humans can easily generalize from observing a set of objects to recognizing objects that have never been seen before.

Machine vision based 3D object retrieval and recognition are basically attempts

to mimic the human capability to distinguish different 3D objects. Very similarly to the human brain, deep learning neural networks allow computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. Convolutional Neural Networks (CNNs) [30] [53] are very often used in image processing giving breakthrough in some visual recognition tasks however, the memory and computational requirements of the application (and training) of such networks is quite large. As the number of embedded and autonomous systems evolves there is a need for new recognition systems, with low power consumption, as alternatives for CNNs. Thus 3D object retrieval and recognition techniques need to be developed which are less complex computationally, use less energy, and need limited amount of memory. This gives a motivation for our research.

Optical 3D object retrieval and recognition have many problems in general such as scaling, illumination changes, partial occlusion, and background clutter. In case of capturing 3D objects with mobile devices viewpoint variation and image noise (e.g. motion blur due to hand shaking in poor lighting conditions) can decrease the retrieval and recognition rate significantly. Numerous retrieval and recognition algorithms have been developed, most of them apply single image-based retrieval and recognition. Single view methods may easily fail when there is strong similarity between the captured images or when the background clutter or partial occlusion masks distinctive features. Video based approaches can use more views but suffer from the increased complexity. To avoid big data/cloud processing type of solutions rises a need for efficient lightweight but robust techniques that could run in low cost embedded systems without a high performance back-end support. I. e. our motivation is to develop methods which can retrieve/recognize specific objects viewed from multiple directions. The methods should be relatively fast, require small memory

and easy to extend with new objects (easy to train).



Figure 1.1: Real-life recognition examples.

1.2 Coding Information for Multiple-views with Orientation Sensors

While optical information is crucial for the recognition of 3D objects of the real world, other information, such as depth and audio data (e.g. [38] and [42]) are also often utilized. But besides these, new sensors appeared in the last few years in handheld devices: Inertial Measurement Units (IMUs), which sense either gravity, acceleration or orientation, are being embedded into devices more frequently as their cost continues to drop. These devices hold a number of advantages over other sensing technologies: they directly measure important parameters for human interaction e.g., the orientation, position, and motion of cameras and they can easily be embedded into mobile platforms giving low dimensional data.

However, the utilization of these sensors, to help the retrieval and recognition process, is not investigated yet. In this work we build up multi-view object models composed of 2D images with corresponding orientation information. Thus instead

of depth and texture data we use 2D image information and orientation from several viewpoints.

It is obvious that video gives much more visual information about 3D objects than simply 2D projections. As will be discussed later, not only the different views of the objects can be recorded but the 3D structure can be reconstructed by direct [46] or indirect [75] structure from motion techniques. However, these object centered approaches often require high quality images and camera calibration with relatively large computational power still far from most of the mobile computing platforms and intelligent sensor nodes.

1.3 Object Model-centered or Appearance-based Approaches

The 3D object retrieval and recognition methods can be divided into two categories: (object) model-centered methods and appearance-based or view-centered methods. Object-centered 3D object retrieval aims to retrieve 3D objects which are represented by 3D models. While these methods compare 3D object information during the search, it is still not straightforward to obtain satisfactory 3D information in several real-world applications, especially when passive optical sensors are used. Structure from Motion (SfM) [77, 55] is a remote sensing technique that uses multiple views of an object to create a three-dimensional set of points, corresponding to the surface, often with associated RGB color values. SfM is often used to reconstruct large spaces ([79]); after georeferencing the point cloud with ground control points, taken with a GPS unit defining the position of recognizable features, the data can be converted to a digital elevation model (DEM). Moreover SfM can be extended with appearance features for recognition purposes [10] and appearance features can be

also used for 3D reconstruction and then for recognition as in [43]. SfM is often used for Simultaneous Localization and Mapping (SLAM) where being real-time can be critical, especially for robotics and unmanned vehicles. In [70] and [25] we have seen that direct SfM can be carried out in real-time with dense data. However, please note that 3D model reconstruction is still computationally expensive and can be unnecessary if our goal is (only) object retrieval and recognition from 2D images. That is we propose that the for the recognition of objects textural and color information, seen from several views, is satisfactory in general. Extending our work with 3D features (such as viewpoint feature histograms - VFHs) or 3D reconstruction could increase the hit-rate. Besides, with the widely spreading active depth acquisition devices (such as Kinect and lidars), it becomes feasible to record color and/or depth information for real objects. These special sensors and the SfM techniques are out of the focus of our research.

The appearance/view-based approach represents objects as collection of 2D views, sometimes called aspects or characteristic views [51], [35]. The advantage of a such approach is that it avoids having to construct a 3D model of an object as well as having to make 3D inferences from 2D features. Many approaches to view-based modeling represent each view as a collection of extracted features, such as extracted line segments, curves, corners, line groups, regions, or surfaces. The success of these view-based recognition systems depends on the extent to which they can extract their requisited features. With real images of real objects in unconstrained environments, the extraction of such features can be unreliable while representing of 3D information in 2D views is not straightforward.

There are several psychophysical supports for two-dimensional view interpolation theory for object recognition in the human visual system. In [13] it is suggested

that the human visual system can be described by recognizing 3D objects by 2D view interpolation. In [27] viewpoint aftereffects also prove that object-selective neurons can be tuned to specific viewing angles in the human visual system. The spatial properties of objects were always considered as significant information in the retrieval and representation of images [15]. View-centered recognition methods can be considered as early machine vision attempts for the recognition of 3D objects. The idea of storing only a limited number of views of 3D objects and then applying transformations to find correspondence with other views already appear for example in [6] where novel views are generated by the linear combination of stored ones. Rigid objects with smooth surfaces and articulated objects could also be represented this way.

1.4 Contributions

The main contribution of this work is introducing different lightweight algorithms for the retrieval and recognition of a limited number (100-1000) of 3D objects. It is an advantage that the proposed techniques don't require camera calibration and can be implemented in embedded systems with limited resources (memory and processing power). Moreover, the proposed methods are efficient if the quality of the queries is low, have a built-in mechanism to amend, based on IMU data, possible missing visual information by the insertion of candidates to the evaluation set.

Contrary, it is not the purpose of our work to find the most appropriate visual feature extractors and descriptors. While we apply the Color and Edge Directivity Descriptor (CEDD) [16] [17] as an efficient, fast and low dimensional descriptor our proposed model also could use other popular descriptors (such as SIFT, FAST, GHOST, etc.).

To prove the efficiency of our work several test-beds were used and in order to obtain realistic test scenarios significant motion blur and heavy Gaussian noise was applied on the query images. Besides these simulations we carried out real-life tests where the queries were generated with the help of a semi-automatic process: first the target object was marked by the user then, in the consecutive frames, a mean-shift based algorithm was tracking the object and generating the further query images.

1.5 Thesis Organization

After giving an overview of related papers we define object retrieval and recognition problems and some structures that are used for solving these problems in Chapter 2. In Chapter 3, we describe the various object model datasets and query datasets used in our experimental evaluations. Chapter 4 describes a 3D object retrieval mechanism without using IMU data. This method, using KD-Tree indexing and the Hough transform paradigm, is compared to the approach when all frames of the query are used in an extensive retrieval process.

In Chapter 5, we will show the utilization of IMU data in the Hough paradigm resulting in the increase of hit-rates at relatively fast operation. Besides retrieval, object recognition is presented. Moreover, an application for active perception is also introduced.

Chapter 6 introduces a new 3D object retrieval model by the following mechanisms: viewer-centric recognition, Markovian estimations, and the fusion of information originating from the visual and orientation subsystems. We have built a Hidden Markov Model (HMM) framework where 2D object views correspond to states, observations are coded by compact edge and color sensitive descriptors, and orientation sensors are used to secure temporal inference by estimating transition probabilities

between states.

Finally, the three theses about the solution of the 3D object retrieval and recognition problems, and the list of publications are presented in Chapter 7.

Chapter 2

Problem Formulation and Preliminaries

The aim of this chapter, after making an overview of similar techniques, is to define the targeted 3D object retrieval and recognition problems and describe general structures used to solve these problems in later chapters.

2.1 Previous Works

Object recognition is a fundamental task in image processing, still object of research and development for devising automatic solutions. A good broad overview of this topic is well-written in [40]. The general main idea is to find a set of features to describe and discriminate objects of interest from others. A general overview of the process is in Figure 2.1, where first visual features (e.g. lines, corners, edges, color, object shape) are first detected, described, then compared to previously learnt patterns.

Beside this general model Deep Neural Networks (DNN) represent a different approach where feature detection filters and comparison mechanisms are not directly

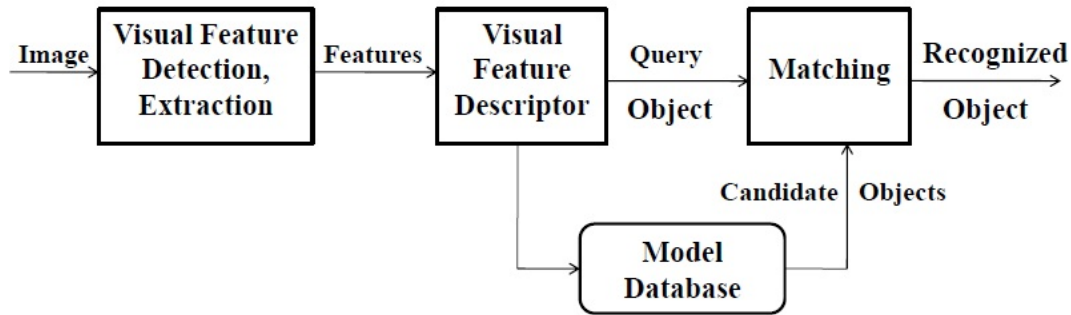


Figure 2.1: Basic object recognition flowchart.

separated [73], [52]. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. DNNs discover intricate structure in large data sets by using the back propagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation of the previous layer. While DNNs have very high performance for several recognition tasks, they are prone to overfitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data. A key questions with DNNs is training since it needs lots of examples and it is not possible to explicitly define positive or negative rules. To increase its performance the preprocessing of images might be necessary [54]. Also there are alternatives such as Cerebellar Model Articulation Controllers (CMAC) where training can be guaranteed to converge relatively fast [64]. In short: DNNs, and especially CNNs, can be considered as hierarchical, mass feature detectors, which have the power by the combination of large number of various filters. In contrast, we will show that relatively simple features can be used if multiple views are applied and supported with orientation information. Since the feature representation of our approach is much more simple it is very easy to extend our system with new objects. Now we introduce earlier and recent 3D object recognition systems in terms of the types of information they encode and how this

information is organized and used for retrieval by multiple views. Since our approach greatly differs from DNNs we don't analyze such techniques.

In an early paper of [48] recognition was achieved from video sequences by employing a multiple hypothesis approach. Appearance similarity, and pose transition smoothness constraints were used to estimate the probability of the measurement being generated from a certain model hypothesis at each time instant. A smooth gradient direction feature was used to represent the appearance of objects while the pose was modeled as a von Mises-Fisher distribution. Recognition was achieved by choosing the hypothesis set that has accumulated the maximum evidence at the end of the sequence. Unfortunately, the testing of the method was carried out on four objects only.

In [12] authors created object models, for video object recognition, with the help of SIFT points gathered from images taken by rotating around the object. Feature points were tracked from frame to frame and video matching was achieved by the comparison of every view of the query with all components of the optimized models of candidates. While the accuracy was about 80% in case of 25 objects, the complexity was too high to be implemented on mobile platforms. Also no explicit technique was utilized to discover the intrinsic structure of sequentially recorded query images and IMUs were not used.

In [61] also SIFT points were used as visual features. The underlying topological structure of an image dataset was generated as a neighborhood graph of features. Graph pruning was used to get simplified structures and motion continuity in the query video was exploited to demonstrate that the results, obtained using a video sequence, are more robust than using a single image. The ratio of correct retrieval increased to 80% with the method from only 20% of single image queries in case of 100 objects while the complexity was not discussed. Besides using computationally

intensive feature extraction only visual sensors were used in the recognition process.

In [34] in addition to the camera they used the accelerometer and the magnetic sensor to recognize the landscape. Clustered SURF (Speeded Up Robust Features) features were quantized using a vocabulary of visual words, learnt by k-means clustering. For tracking objects the FAST corner detector was combined with sensor tracking. Because of the small storage capacity of the mobile device a server-side service was needed to store the large number of images.

Gao et al. [36] proposed a hypergraph learning method for 3D object retrieval, in which the relevance among 3D objects is formulated in a hypergraph structure. In [31] the bag-of-visual-words approach was applied to view-based 3D model retrieval. Each 3D model was rendered into a group of depth images, and SIFT features were extracted from these depth maps to generate bag-of-features determining the distance between 3D models.

The authors in [4] introduced an Adaptive Views Clustering (AVC) method. In AVC, there are 320 initial views which are captured and representative views are optimally selected by adaptive view clustering with Bayesian information criteria. A probabilistic method is then employed to calculate the similarity between two 3D models, and those objects with high probability are selected as the retrieval results. There are two parameters in the method, which are used to modulate the probabilities of objects and views, respectively.

In [22] authors proposed a Compact Multi-View Descriptor (CMVD) method, in which 18 characteristic views of each 3D model are first selected through 18 vertices of the corresponding bounding 32-hedron. In CMVD, both the binary images and the depth images are taken to represent the views. Then the comparison between 3D models was based on the feature matching between selected views using 2D features, such as 2D Polar-Fourier Transform, 2D Zernike Moments, and 2D

Krawtchouk Moments. For the query object, the testing object rotated and found the best matched direction for the query object. The minimal sum of distance from the selected rotation direction was calculated to measure the distance between two objects.

HMMs are often used in different recognition problems such as speech, musical sound, or human activity recognition but we relatively rarely meet them in the recognition of 2D or 3D visual objects. This is natural since ordered sequences of features are needed to construct HMM models. In [45] affine invariant image features are built on the contours of objects, and the sequence of such features are fed to the HMM. This approach is interesting but seemed to be too unnatural to have later followers. Another approach was proposed in [41], where range images are modeled using HMMs and Neural Networks, using 3D features such as surface type, moments and others.

In [24] authors presented an approach for face recognition using Singular Values Decomposition (SVD) to extract relevant face features, and HMMs as classifier. In order to create the HMM model the 2 dimensional face images had to be transformed into 1 dimensional observation sequences. For this purpose each face image was divided into overlapping blocks with the same width as the original image and a given height, and the singular values of these blocks were used as descriptors. A face image was divided into seven distinct horizontal regions: hair, forehead, eyebrows, eyes, nose, mouth and chin forming seven hidden states in the Markov model. While the algorithm was tested on two standard databases, the advantage of the HMM model over other approaches was not discussed.

The method of Torralba et al. [74] seems to be more close to a real-life temporal sequence: HMM was used for place recognition based on the sequences of visual observations of the environment created by a low-resolution camera. It was also

investigated how the visual context affects the recognition of objects in complex scenes.

In [23] a HMM was used to model the sequence of 2D views gathered from a moving camera, where each view was described using contour-based features. It seems to be a drawback that the contour of the object, often difficult to detect, was employed to compute features, discarding important information such as texture and colors. A thorough experimentation with a standard database is missing, and occlusions were not considered.

The most similar viewer-centered HMM based 3D object retrieval method to ours (described in Chapter 6) was published by Jain et al. [47]. However, there are many differences to our work and there are many ambiguous details in [47]: it is not clear how the crucial emission and transition probabilities were estimated and also the dimension of the applied image descriptor seems to be too small for real-life applications. The dataset in their tests included only gray-scale computer generated images without texture and no orientation sensor was used during the recognition.

Generally speaking the problem of using HMMs in 3D object recognition (especially with handheld devices) is that the transition between states is greatly determined by the user and can not be easily put into the HMM models of objects. To avoid this problem we use orientation sensors to estimate the transition probabilities.

2.2 General Problem Formulation

3D objects form an important type of data with many application domains such as CAD engineering, robotics, visualization, cultural heritage, and entertainment. Technological progress in acquisition, modeling, and dissemination of 3D geometry leads to the accumulation of large repositories of 3D objects. Let us now introduce

the definition of the 3D object retrieval and recognition problems in the field of computer vision. Please note, that our purpose is to recognize specific objects in contrast to the generic recognition of object categories ([40]).

The 3D object recognition problem can be defined as a labeling problem based on (multiple) views and on known specific models of given rigid 3D objects. An image or images containing one 3D object of interest (and background) and a set of labels, corresponding to a set of models known to the system, is given. We should assign the correct label to regions, or a set of regions, in the images. The 3D object recognition problem is basically accomplished by matching visual features of a query image or query images and of models of possible objects.

The architecture of our 3D object recognition system (as shown previously in Figure 2.1) must have the following components to perform the task:

- **Visual Feature Detection and Extraction.** Visual feature is some visually observable attribute of the 3D object that will be used in describing and recognizing in relation to other objects. Size, color, texture, and shape are some commonly used features. The feature detection and extraction applies operators to identify locations and measure features that help in forming 3D object hypotheses. The following properties are important for utilizing a good feature detector in computer vision applications:
 - Robustness and repeatability: the feature detection algorithm should be able to detect the same feature locations independent of scaling, rotation, shifting, radiometric deformations, compression artifacts, and noise.
 - Distinctiveness: the features should be able to show difference between different objects.
 - Efficiency: the feature detection algorithm should be able to detect fea-

tures quickly (for the sake of real-time applications) and should use limited amount of memory.

- Quantity: the feature detection algorithm should be able to detect all or most of the important features in the image. The cover of the object is not always satisfactory, especially occlusion can cause considerable problems.

- **Visual Feature Descriptor.** Feature descriptors encode feature information into a series of numbers and act as a sort of numerical “fingerprint” that can be used to differentiate one feature from another. Ideally this information would be invariant under image transformation, so we can find the feature again even if the image is transformed in some way. The efficient feature descriptor must have several advantages, e.g., small size, low computational cost, easy to compare, suitable for use in large image datasets and insensitive to blur, color distortion and geometrical distortions.
- **Visual Feature Matching.** Once we have extracted features and generated their descriptors from two or more images, the next step is to establish feature matching between these images. Some transformation invariance can be achieved also with the matching process itself. The complexity of matching can have a great effect on the running time, that is why indexing techniques are often used.
- **Model Database.** Object representations must be suitable for use in object recognition under various conditions, which means that there must be some potential correspondence between the trained representations and the features that can be extracted from a query. In general the conditions under which the models are built are better than those of making queries.

In our approach 3D object retrieval is the process where a query is initiated by a

user within a collection of 3D objects and a retrieval system responds to the query by presenting a list of retrieved objects sorted in decreasing similarity to the query, according to a predefined measure of object similarity. Each 3D model is represented by a group of 2D views, in several cases also the view's relative orientation is also given. How to perform multiple view matching is the key topic in view-based 3D model retrieval tasks.

2.3 View-Centered Retrieval and Recognition

Our view-centered representations model the outlook of objects as captured from different viewpoints. Since objects can look very differently from different viewing directions image feature descriptors, the storing database, the search mechanism, and the feature similarity measure should be carefully designed to minimize the amount of data space, retrieval time and to maximize the hit-rate of recognition and retrieval.

Figure 2.2 illustrates the view-centered approach where multi-view object models are built up from 2D images taken from different orientations and being the object of interest in the focus of the camera. To get a complete object model a larger number of different Azimuth and Elevation angles are required. However, for average applications the elevation can be limited (in our tests we used only one elevation angle typical for an object placed on the table).

Why the term recognition is often used generally in our context we can find the difference between recognition and retrieval based on the output of the process: in case of retrieval we know that the target object must be one of the candidates while in case of recognition a previously unknown object can also be targeted. I.e. in the later case we should have the possibility to reject the query as none of those known

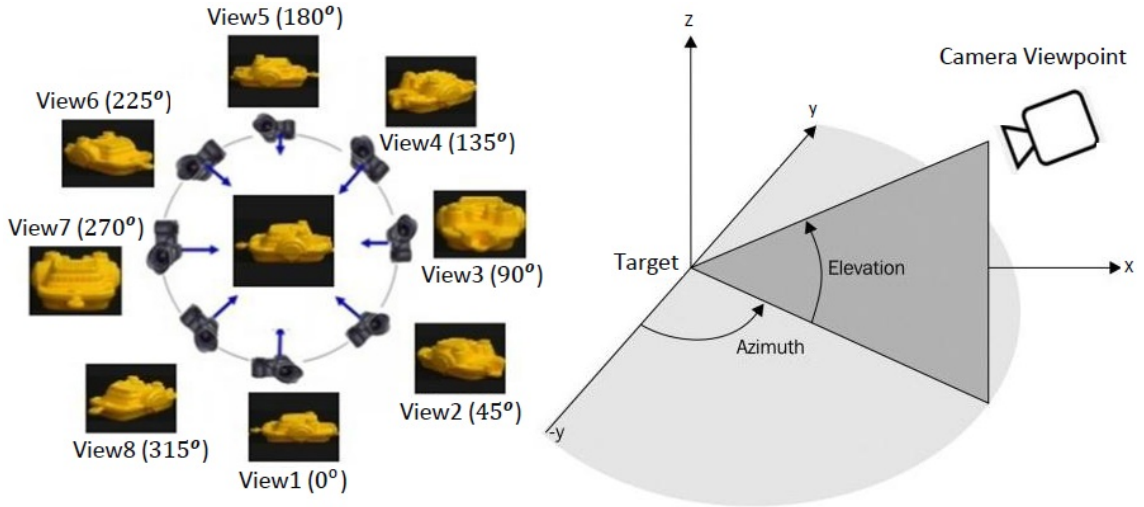


Figure 2.2: Model generation setup with target object in the centre.

from previous training.

2.4 Visual Features

In this section we discuss the visual features used in our framework. As mentioned above, there are four main aspects of choosing the right features for a specific image retrieval task: to carry enough information to distinguish images; to be invariant to possible distortions; to cover enough area; to be subject of fast and robust comparisons. More precisely there are the following aspects to consider: robustness, invariance, repeatability, efficiency, accuracy, quantity, locality, distinctiveness. Previously (see [21] and [20]) we investigated different types of descriptors in real-life circumstances: MPEG-7 based methods (MPEG7 CLD, MPEG7 EHD, MPEG7 SCD, MPEG7 Fusion); local feature based methods (SURF, SURFVW [16], SIFT); Compact Composite Descriptors [16] [17] (Compact CEDD, CEDD, Compact FCTH, FCTH, JCD, CCD Fusion, Compact VW); and others (Tamura texture descriptor, Color Correlogram and Correlation (ACCC) [76], MPEG7-CCD Fusion

[17]). We found two main effects that could seriously degrade the performance of image descriptors in real-life conditions: the change in appearance of colours under different lighting conditions and colour balance settings, and the loss of contrast due to motion blur typically occurring when the image is taken under low lighting conditions with a handheld camera. Detailed descriptions and results of those tests are available in [21] and [20]. Contrary to the popularity of SIFT (and similar descriptors in its family such as SURF) in image retrieval we found serious drawbacks such as running time, touchiness to blur and high dimensionality. CEDD was found one of the most robust, fast and compact among those. In [21] and in [20] it is showed that CEDD is quite tolerant for different noises and can be computed fast in today's mobile platforms. Since then new descriptors were developed, [57] and [69] give good overviews and analysis about many of them. However, since our purpose was to develop lightweight methods we did not change our decisions to test new alternatives, the very compact CEDD was used in all of our tests. However, we emphasize that other descriptors are also subject to be used since the framework defined in Chapter 4, 5 and 6 could be used with other visual features.

CEDD [16] is a block-based approach where each image block is classified into one of 6 texture classes (non-edge, vertical, horizontal, 45-degree diagonal, 135-degree diagonal, and nondirectional edges) with the help of MPEG7 EHD (Edge Histogram Descriptor). Then for each texture class a 24 bin color histogram is generated where each bin represents colors obtained by the division of the HSV color space. The values of the generated histogram of length 6x24 are then normalized and quantized to 8 bits. Besides its robustness and compactness, however, we should note that there are two disadvantages of CEDD compared to some other popular local point descriptors:

- in its original form it is not rotation invariant;

- as it is a global (area-based) descriptor it needs proper area selection for the target.

The first issue can be handled with a proper similarity measure (see Section 2.6), the second can be fulfilled with manual or automatic segmentation methods. While in our recent application and tests a bounding rectangle around the target was designated manually (or rather the camera was moved to have the object within a bounding box), a good choice for an automatic method could be the Grabcut algorithm known from [67]. It is clear that there are always newer and better global and local descriptors (for overviews see [57] and [69]), the selection of the most appropriate one is out of focus of this thesis.

2.5 Feature Indexing Using KD-Tree

To support the fast search of candidates there are two class of approaches: tree indexing and hashing algorithms. Index structures, containing pointers to descriptor data, are structures to speed up information retrieval and are typically built up off-line. KD-Tree is an efficient data structure, established by Friedman, Bentley, and Finkel [29], [71] and is often used for fast indexing and retrieval. KD-Tree is defined as a binary tree in which every node is a k-dimensional point. Every non-leaf node can be thought of as implicitly generating a splitting hyperplane that divides the space into two parts, known as half-spaces. Points to the left of this hyperplane are represented by the left subtree of that node and points right of the hyperplane are represented by the right subtree. Given a set of N dimensional vectors to be represented, KD-Tree is constructed as follows. First, compute the mean and variance at each dimension of the set, then find the coordinate with the biggest variance and compute the mean at this dimension. Non-leaf nodes divide the set

into two parts known as left subtree (LS) and right subtree (RS). LS contains only the vectors which are smaller or equal to M at the coordinate of highest variance and RS otherwise. Leaf nodes contain those sets which are not divided further. The search is sequential among the elements of leaf nodes.

In [3] authors improved the KD-Tree for a specific usage: indexing a large number of SIFT and other types of image descriptors. They also extended priority search to search among multiple trees in a simultaneously way. In [2] parallel KD-Trees were explored for ultra large scale image retrieval in databases containing dozens of millions of images. In this thesis we also use KD-Tree, however, the number of candidate views (typically below 100,000) does not require the use of such multiple tree solutions.

Recently, product quantization (PQ) [49] and its extensions are popular and successful approximated nearest neighbor search (ANN) methods for handling large-scale data. Each database vector is quantized into a short code, which we call a PQ-code. The search is conducted over the PQ-codes efficiently using lookup tables. The essence of product quantization is to decompose the original high-dimensional space into the Cartesian product of a finite number of low-dimensional subspaces that are then quantized separately. Optimal space decomposition is important for the performance of ANN search, but still re-mains unaddressed. In general, PQ based approaches consist of the following three main steps: (1) a robust proposal mechanism is used to identify a list of nearest neighbor candidates in the database, (2) a re-ranking step then sorts these candidates according to their ascending approximate distances to the query vector. Finally, the approximated k -nearest vectors after re-ranking are further sorted using an exact distance calculation. In [78] the authors presented an extension to the family of PQ methods called the Product Quantization Tree (PQT). The main contributions of this approach are:

a two level product and quantization tree that requires significantly fewer exact distance tests than prior work; a relaxation of the algorithm for an effective bin traversal order; a fast, re-ranking step to approximate exact distances between query and database points in constant time $O(P)$, where a query vector is split into P parts; and a highly optimized GPU based open-source implementation.

2.6 Image Similarity Measure

The similarity between two CEDD descriptor vectors is efficiently given with the help of the Tanimoto Coefficient [16]. Let q_i be the descriptor of the i th view from the query and c_j be the descriptor of the j th view of a candidate. The Tanimoto coefficient is then:

$$TC(q_i, c_j) = \frac{q_i^T c_j}{q_i^T q_i + c_j^T c_j - q_i^T c_j} \quad (2.1)$$

where q_i^T is the transpose vector of the descriptor q_i . In case of absolute congruence of the vectors, the Tanimoto coefficient takes the value 1, while in case of maximum deviation the coefficient tends to 0. The Tanimoto coefficient was found to be preferable to the similarity functions L_1 (Absolute), L_2 (Euclidean Distance) [63] because of better performance in retrieval tasks. The difference of CEDD vectors is:

$$T(q_i, c_j) = 1 - TC(q_i, c_j) \quad (2.2)$$

We need a modified Tanimoto coefficient to achieve rough rotation invariance:

$$TC^R(q_i, c_j) = \max_{roll} TC(q_{i,roll}, c_j) \quad (2.3)$$

where $roll \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ and $q_{i,roll}$ means that orientation specific texture

class positions are shifted within the CEDD vector of the query. (Please note, that the extraction of CEDD descriptor values should not be changed, only comparisons take more time to fit the actual candidate c_j best).

Please note, that since objects have different appearances from different directions, they will be represented with several frames and the corresponding descriptors. These will be denoted such $c_{j,f}$ (that is descriptor of frame f of object j). Reasonably for queries we use only one index to denote the view.

2.7 The Hough Transformation Paradigm

In this section we discuss the Hough transform method since it will be used as the framework for the combination of weak classifiers. Originally, the Hough transform is a technique which can be used to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. but it is also known to be used for object detection [32], object tracking [33] and action recognition [14]. The authors in [5] generalized the classical Hough transform in fuzzy set theoretic framework, called fuzzy Hough transform, in order to handle the imprecise shape description. In [56] a discriminative Hough transform based object detector is presented where each local part casts a weighted vote for the possible locations of the object center. The weights can be learned in a max-margin framework which directly optimizes the classification performance and improved the Hough detector. In [66] the circular Hough transform is used to detect the presence of circular shapes and in [72] authors also used the Hough transform to improve the detection of low-contrast circular objects. Considering the basic approach for detection, the main advantage

of the Hough transform technique is that it is tolerant for gaps in feature boundary descriptions, it is not strongly affected by occlusion in the image and it is also relatively unaffected by image noise.

In general there are three main steps of retrieval in this framework:

1. Feature extraction: visual, depth or other observation data can be used to generate descriptors ($d_i \in D$) to gather information about a space/time event or object.
2. Voting: each occurrence of a descriptor votes for a candidate c_i with a weight $\Theta(c_i, d_i)$. In many cases the weights are determined by training and their value can also depend on different factors (such as the observation's distance from the candidate or reliability).
3. The Hough Score for a candidate is given:

$$H(c_i) = \sum_{d_i} \Theta(c_i, d_i), \quad (2.4)$$

and the retrieval is done by selecting the object with the highest score:

$$\hat{c} = \arg \max_i H(c_i). \quad (2.5)$$

In our approaches d_i will be the CEDD descriptors and IMU data (the role of IMUs will be described in details in Chapter 5) .

2.8 Hidden Markov Models

Hidden Markov Models (HMMs) are statistical frameworks in which the system being modeled is assumed to behave as a Markov process with directly unobservable

(hidden) states. An HMM can be considered as the simplest dynamic Bayesian network. The logic behind HMMs was already introduced in the late 1960s and early 1970s in the works of [7], [8]. A HMM is a probabilistic model, which originates from discrete first-order Markov processes.

In this section we briefly overview the theoretical background of HMMs and some related problems. Let $S = \{S_1, \dots, S_N\}$ denote the set of N hidden states of the model. In each t index step this model is described as being in one $q_t \in S$ state, where $t = 1, \dots, T$. Between two steps the model undergoes a change of state according to a set of transition probabilities associated with each state. The transition probabilities have first-order Markov property, *i.e.*

$$P(q_t = S_i | q_{t-1}, q_{t-2} = S_k, \dots) = P(q_t = S_i | q_{t-1} = S_j) \quad (2.6)$$

Furthermore, we only consider the processes, where the transitions of Equation 2.6 are independent of time. Thus we can define the set of transition probabilities in the form

$$a_{ij} = P(q_t = S_i | q_{t-1} = S_j) \quad (2.7)$$

where i and j indices refer to states of HMM, $a_{ij} \geq 0$, and for a given state $\sum_{j=1}^N a_{ij} = 1$ holds. The transition probability matrix is denoted by $\mathbf{A} = \{a_{ij}\}_{1 \leq i, j \leq N}$. We also define the initial state probabilities:

$$\pi_i = P(q_1 = S_i) \quad (2.8)$$

and $\pi = \{\pi_i\}_{1 \leq i \leq N}$. Now we extend this model to include the case, where the observation is a probabilistic function of each state. Let $O = \{o_1, o_2, \dots, o_T\}$ denote the set of observation sequence. The emission probability of a particular o_t observation

for state S_i is defined as

$$b_i(o_t) = P(o_t|q_t = S_i) \quad (2.9)$$

The set of all emission probabilities is denoted by $\mathbf{B} = \{b_i(\cdot)\}_{1 \leq i \leq N}$. The complete set of parameters of a given HMM is described by $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$. A more comprehensive tutorial on HMMs can be found in [65].

2.8.1 Discrete Density HMMs (DHMM)

In case of discrete HMMs we assume that the observation sequences are symbols from a discrete $V = \{v_1, \dots, v_M\}$ alphabet. The symbols correspond to the physical output of the model. In this case the emission probability for a given $o_t = v_k$ observation is

$$b_i(o_t) = P(o_t = v_k|q_t = S_i) \quad (2.10)$$

The \mathbf{B} set of emission probabilities can easily be implemented as a 2-D array of size $N \times M$.

2.8.2 Continuous Density HMMs (CHMM)

Many applications have to work with continuous signals, and Gaussian Mixture Model (GMM) is a widely used representation of the emission probability. In this work we use GMM emission probabilities only, which is defined as

$$b_i(o_t) = \sum_{m=1}^M \omega_m \mathcal{N}(o|\mu_m, \Sigma_m) \quad (2.11)$$

where M is the number of mixture components.

2.8.3 HMM Problems

There are three fundamental problems of interest that must be solved for HMM to be useful in some applications. These problems are the following:

- **Evaluation problem.** Given an HMM $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ and an observation sequence $O = \{o_1, o_2, \dots, o_N\}$, determine the probability that model λ has generated observation sequence O , *i.e.* $P(O|\lambda)$.
- **Decoding problem.** Given an HMM $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ and an observation sequence $O = \{o_1, o_2, \dots, o_N\}$, calculate the most likely sequence of hidden states that produced observation sequence O .
- **Learning problem.** Given some training observation $O = \{o_1, o_2, \dots, o_N\}$ and general structure of HMM (number of hidden and visible states), determine HMM parameters $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ that best fit training data.

2.8.3.1 Solution to Evaluation Problem using Forward-Backward Algorithm

The Forward-Backward algorithm [50] uses two auxiliary variables for the parameter estimation. First, the forward $\alpha_t(i)$ and backward $\beta_t(i)$ variables are calculated inductively as follows.

- **Forward Algorithm.** The forward variable $\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = S_i|\lambda)$ is the probability of observing the partial sequence o_1, o_2, \dots, o_t such that the state q_t is S_i .

1. Initialization:

$$\alpha_1(i) = \pi_i b_i(o_1) \tag{2.12}$$

2. Forward recursion:

$$\alpha_{t+1}(i) = b_j(o_{t+1}) \sum_{i=1}^N \alpha_t(i) a_{ij}, \quad 1 \leq j \leq N \quad (2.13)$$

From the definition of Equation 2.12 and 2.13 it is obvious that

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (2.14)$$

- **Backward Algorithm.** The backward variable $\beta_t(i) = P(o_{t+1}, o_{t+2}, \dots, o_T, q_t = S_i | \lambda)$ is defined similarly, and is the probability of observing the partial sequence from $t + 1$ to T , given state S_i at t .

1. Initialization:

$$\beta_T(i) = 1 \quad (2.15)$$

2. Backward recursion:

$$\beta_t(i) = \sum_{j=1}^N \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), \quad 1 \leq i \leq N, t = T - 1, \dots, 1 \quad (2.16)$$

From the definition of Equation 2.15 and 2.16 it is obvious that

$$P(O|\lambda) = \sum_{i=1}^N \beta_1(i). \quad (2.17)$$

2.8.3.2 Solution to the Decoding Problem using Viterbi Algorithm

The problem of many real-world applications is to efficiently determine the most probable state sequence given an observation sequence O , *i.e.* we want to find the sequence, which maximizes $P(Q, O|\lambda)$. We can use the Viterbi algorithm [28] to

find the state sequence. The variable δ_t is the maximum probability of producing observation sequence o_1, o_2, \dots, o_t when moving along any hidden state sequence q_1, q_2, \dots, q_{t-1} and getting into $q_t = S_i$, *i.e.*

$$\delta_t(i) = \max P(q_1, q_2, \dots, q_t = S_i, o_1, o_2, \dots, o_t | \lambda) \quad (2.18)$$

and can be calculated inductively as

1. Initialization:

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N \quad (2.19)$$

2. Recursion:

$$\delta_{t+1}(j) = \max_i [a_{ij} b_j(o_{t+1}) \delta_t(i)], \quad 1 \leq j \leq N \quad (2.20)$$

Finally, choose best path ending at T, *i.e.*

$$P^* = \max_i [\delta_T(i)]. \quad (2.21)$$

2.8.3.3 Solution to the Learning Problem using Baum-Welch Algorithm

The most conundrum problem of HMMs is to determine the model parameters which maximizes the probability of a given observation sequence. The Baum-Welch method [7], [8] is an iterative procedure to estimate model parameters that maximum the likelihood $P(O|\lambda)$. There are three main steps of Baum-Welch algorithm:

- Calculate $\gamma_t(i)$ which is the probability of being in state s_i at index t given the observation sequence O and the parameters λ .

- Calculate $\xi_t(i, j)$ which is the probability of being in state s_i and s_j at index t and $t + 1$, respectively given the observation sequence O and parameters λ .
- Using the above formula γ and ξ , one can re-estimate model parameters.

2.9 Object Tracking

Object tracking is one of the key tasks in the field of computer vision when processing a sequence of images. Tracking is the estimation and analysis of trajectories of objects in the plane of the image by moving through a sequence of images. Various methods of object tracking are available, we only mention one, used in our applications later, which can be used with limited resources in real-time applications.

The *Meanshift* algorithm is designed for static distributions. The method tracks targets by finding the most similar distribution pattern in a frame sequences with its sample pattern by iterative searching. It is simple in implementation, but it fails to track the object when it moves away from camera [80] due to changes in scale.

Continuous adaptive Meanshift (Camshift) easily overcomes this problem. The principle of the Camshift, based on the Meanshift algorithm, is given in [60], [81], [26]. Camshift is able to handle the dynamic distribution by adjusting the size of the search window for the next frame based on the zeroth moment of the current distribution of images. This allows the algorithm to anticipate the movement of objects and quickly track the object in the next frame.

2.10 Summary

In this chapter we discussed the general concepts and tools we use for 3D object retrieval and recognition. We proposed compact visual features and descriptors and

the method to measure the similarity between them. We also discussed the Hough transform paradigm, the concept of Hidden Markov Models, and a tracking method which will serve as the basis for the retrieval mechanisms.

In Chapter 3 we introduce the different types of the 3D object model datasets and queries that are used in our experimental evaluations.

Chapter 3

Datasets and Test Issues

The evaluation of object recognition methods greatly depends on the test data and the testing environments itself. As our purpose is to develop methods to recognize small sized 3D objects in real-life conditions, we created different datasets based on our own images or others' image collections.

3.1 Object Model Datasets

Object models are formed by images taken under good viewing conditions. Each object is captured from several views but the same elevation. These data can be considered as training information for the knowledge base of the recognition mechanisms. We introduce five separate object model datasets, with different size and visual nature. There are several reasons that besides standard datasets we used our own recordings (SUP datasets). Orientation data was precisely given for the standard datasets. Using our IMU sensor (with possible inaccuracy) is more realistic in experimental tests. To make tracking experiments was not possible with online datasets, thus new samples (in SUP-25), with several other visible objects, were necessary.

3.1.1 SUP-16 Dataset

SUP-16 is a small dataset including 16 objects, where 44-73 views per object were captured from the same elevation but from different azimuth leading to approximately 900 images. Objects were centered and a bounding box was manually defined for each image. Image sizes and side ratios varied a lot as shown in Figure 3.1. As we can see the object size, shape, color, contrast can vary from view to view. Some view of the same object can be very different from the other (see f.e. the green pencil or the white cup). The background of the objects were only roughly uniform and the surface of objects was sometimes glossy.



Figure 3.1: Test model object examples from the SUP-16 dataset.

3.1.2 SUP-25 Dataset

The SUP-25 dataset, illustrated in Figure 3.2, includes 25 different objects, where 64-77 images of each object were recorded from different views at the same elevation

leading to approximately 1800 images. This dataset was created by our tablet to test the VCI approach with automatic Camshift tracking applied for the generation of queries (described in Section 5.5.4). Figure 3.3 illustrates some tracked windows and the environment of the objects.

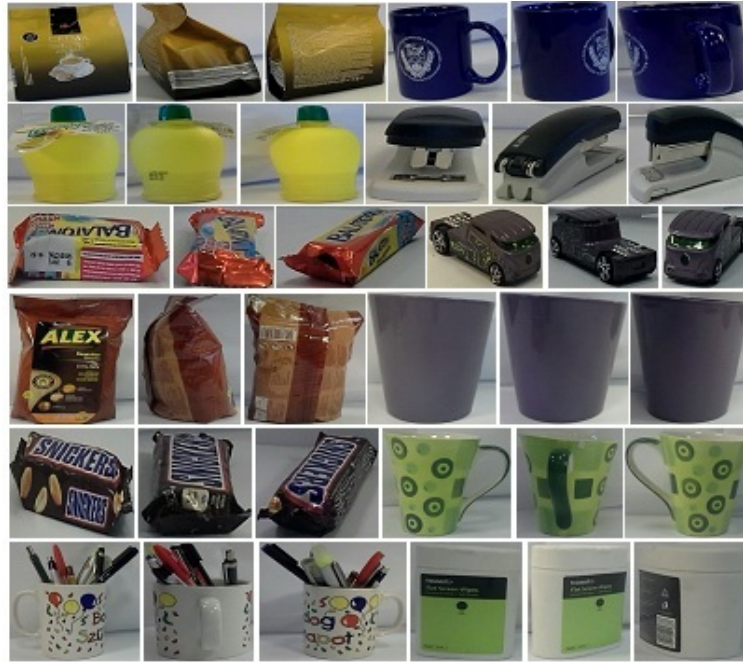


Figure 3.2: Test model object examples from the SUP-25 dataset.



Figure 3.3: Top row: examples for the results of tracking. Bottom row: objects and their environment.

3.1.3 SMO Dataset

The SMO dataset [12], illustrated in Figure 3.4, includes 25 different objects with uniform background, where 36 images of each object were recorded by rotating the object in the plane at 10° steps leading to approximately 900 images. This database is chosen for comparison with the method of [12].



Figure 3.4: Test object examples from the SMO dataset with uniform background.

3.1.4 COIL-100 Dataset

The COIL-100 dataset [59] includes 100 different objects. The objects were placed on a motorized turntable against a black background, 72 images of each object

were taken at pose intervals of 5° . Figure 3.5 shows some examples objects from COIL-100.



Figure 3.5: Test model object examples from the COIL-100 dataset.

3.1.5 ALOI Dataset

The ALOI dataset [37] including 1000 small objects against a black background, where 72 images of each object were recorded by rotating the object in the plane at 5° steps, as examples show in Figure 3.6.

3.2 Query Datasets

In this section, we introduce the methods used to generate query data from the different object model datasets. Since each object was tested 10 times, the query datasets (separate for the different datasets) are composed of 10×8 ($N = 8$)



Figure 3.6: Test model object examples from the ALOI dataset.

randomly selected images of each object, either distortion free, strongly distorted with motion blur, or by additive Gaussian noise. (Please note, that while the 8 views are set to be different, the 10 random test cases have common views at chance) We have chosen these two types of distortions since in previous evaluations [20] we found many image descriptors to be most vulnerable to these common quality degradations happening often in real life.

We used the built-in function of Matlab *imnoise* with standard deviation $sd = 0.012$ to generate additive Gaussian noise (GN) and made motion blur (MB) by *fspecial* with parameters $len = 15$, and angle $\theta = 20$ degrees. Some examples of the distorted queries are shown in Figures 3.7, 3.8, 3.9, 3.10, 3.11. Please note, that besides homogeneous backgrounds in some test we used modified backgrounds (Figure 3.11) or objects placed in changed background and semi-automatically cropped (Figure 3.3). We think that the different datasets and heavy distortions make our

tests somehow realistic.



Figure 3.7: Noisy and blurred query examples from the SUP-16 dataset.

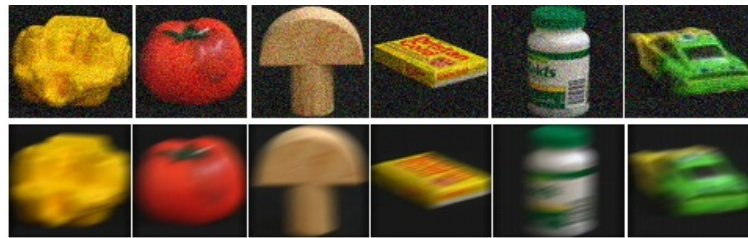


Figure 3.8: Noisy and blurred query examples from the COIL-100 dataset.

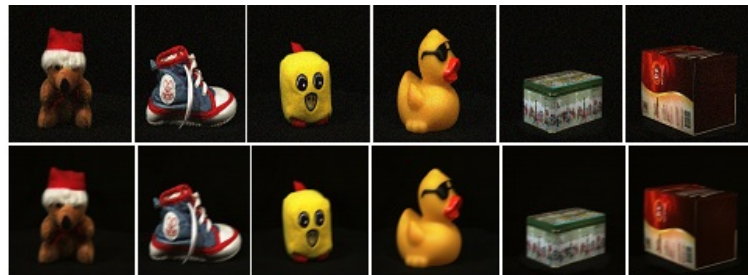


Figure 3.9: Noisy and blurred query examples from the ALOI dataset.

3.3 Hardware and Software Issues

While the development and some offline evaluations were run on a PC, time measurements and our test captures were run on a tablet. The tablet used in our tests



Figure 3.10: Noisy and blurred query examples from the SMO dataset with uniform background.



Figure 3.11: Noisy and blurred query examples from the SMO dataset with textured background.

is a Samsung SM-T311 equipped with Android 4.2.2 Jelly Bean, 1 GB RAM, and ARM Cortex A9 Dual-Core 1.5 GHz Processor. It has a built in accelerometer, digital compass, and a 5MP primary camera with auto focus, and it is capable to make 720 x 1280 HD video recording. Software development was done in Eclipse using OpenCV Library 2.4.8 .

Chapter 4

Weak Classifiers for Object Retrieval with the Hough Paradigm

To achieve efficient (fast, reliable, and memory efficient) retrieval CEDD vectors could be good information source, however, under real-life conditions it has some probability that captured image queries are so distorted that can deteriorate us from the good results. Unfortunately, the evaluation of the quality (and reliability) of the actual descriptors is not straightforward and can be time consuming. But we can assume that some of the queries are reliable and thus only those models should be considered as good candidates which highly correlate with some of the queries (but not necessarily all):

- we make several CEDD query captures and run independent queries,
- KD-Tree indexing enables fast search on CEDD descriptors,
- we only consider those candidates which show high correlation to any of the queries (i.e. not all possible models should fit well),

- we evaluate the best candidates via the Hough framework to find the best matching one.

Our approach will be called Voting of Candidates with Indexing (VCI), with different variants as VCI Best Match, Active VCI.

The rest of this chapter is organized as follows. In Section 4.1, we present extensive image based search (where the candidate object is chosen with the lowest average distance). The candidate voting mechanism with KD-Tree indexing for object retrieval will be introduced in Section 4.2. Finally in Section 4.3, we present the evaluation of the voting mechanism with KD-Tree indexing.

4.1 EIS (Extensive Image Search) Approach

In this approach only visual CEDD descriptors are used for retrieval: N_f^q frames are taken from the video as query frames to compute the average distance resulting in complexity $O(N_c * N_f^q * N_f^c)$, where N_c is the number of candidate objects, and N_f^c is the number of frames in candidates. The distance function between query q and candidate object c_i is:

$$T^{EIS}(q, c_i) = \frac{\sum_{k=1}^{N_f^q} \min_f T(q_k, c_{i,f})}{N_f^q}. \quad (4.1)$$

Note, that we did not utilize the *spatial relation* of query images, they were handled independently and with equal weight. See Figure 4.1 for the illustration of the approach, where the blurred query images are compared to all views of candidates.

The disadvantage of the EIS approach is that we could not handle outliers properly. In real-life cases it can easily happen that an image is taken accidentally, or a query frame has so poor quality (e.g. due to the shaking of the hand) that it should not be considered as a valid query. Since it is not easy to reliably evaluate the quality of a query itself, it can easily happen that a wrong query highly correlates a (wrong)

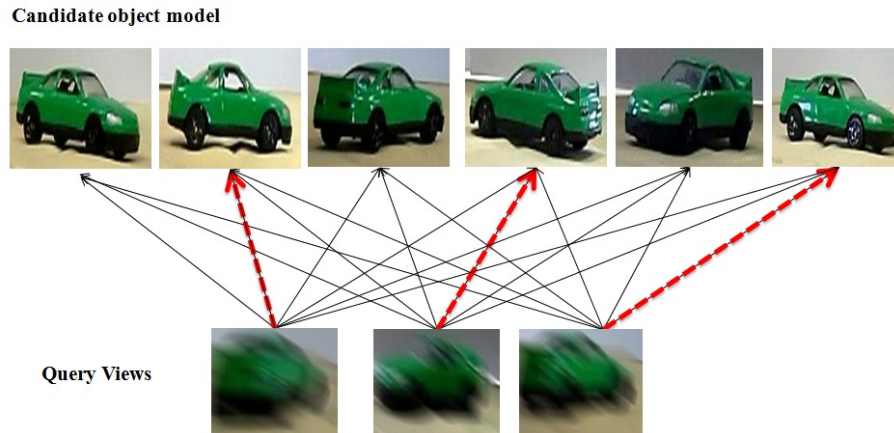


Figure 4.1: Illustration of the EIS approach: only the images of the query are compared to the images of the candidates independently. The similarity of the query sequence and candidates are based on the sum of Tanimoto Coefficients.

candidate giving it high chance to win. Contrary to this extensive search approach we will consider only short lists of candidates that could highly correlate to the sequence of queries. I.e. a possible winner should fit very well to the query (at least to one query), besides, the limited length candidate list has the advantage of fast (partial) evaluation of image similarity (Equation 2.1) by dynamic programming.

4.2 VCI Best Match Approach

An alternative of the above method is to create independent lists of candidates for each query. The elements of the lists are ranked according to visual similarity. To get rid of low badly correlated candidates the length of the lists is limited (typically less than 10). The evaluation of the sequence of these lists is done by the Hough transform approach as a voting process: elements of independent retrieval lists will vote for the most reminder object (by its view). To make the independent searches fast we use KD-Tree indexing by a single KD-Tree containing the CEDD descriptors of all objects, as the number of candidate views (typically below 100,000) does not

require the use of multiple tree solutions.

4.2.1 Voting of Candidate using Indexing (VCI)

View-centered video-based object retrieval and recognition approaches can use thousands of views and thus can easily suffer from high complexity. In our model we have not only one but several CEDD descriptors of the objects extracted from different viewing directions. Object features might overlap among the images of the same object which requires special attention to keep the size of model database at minimum. Moreover, as we have to run several queries in the VCI approach, fast searching mechanisms are required.

In our retrieval process all queries q_i (some frames of the input video sequence) generate their own retrieval list L_i with limited length (e.g. $N_L = 4$) by running independent searches in the object models. That is we have a sequence of retrieval lists, one for each query view, and all the retrieved candidates give votes based on visual similarity. The Tanimoto Coefficient measure similarity between the query and the actual candidate $TC(q_i, c_{j,f})$ will be (yet) the only one term of the vote, so for the Hough Score (Equation 2.4) we get:

$$H(c_i) = \max_{\mathbf{j} \text{ s.t. } c_{i,\mathbf{j}_k} \in L_k} \sum_{k=1}^{N_f^q} TC(q_k, c_{i,\mathbf{j}_k}), \quad (4.2)$$

where vector \mathbf{j} stores the indices of frames of object i in relevance of query k (please note, that we have separate \mathbf{j} for each candidate object). That is \mathbf{j} will contain indices of c_i frames maximizing the Hough Score for object i (\mathbf{j}_k for query q_k).

Practically, satisfying Equation 2.5 means the evaluation of Equation 4.2 when traveling through all paths connecting the views of the same objects on the different

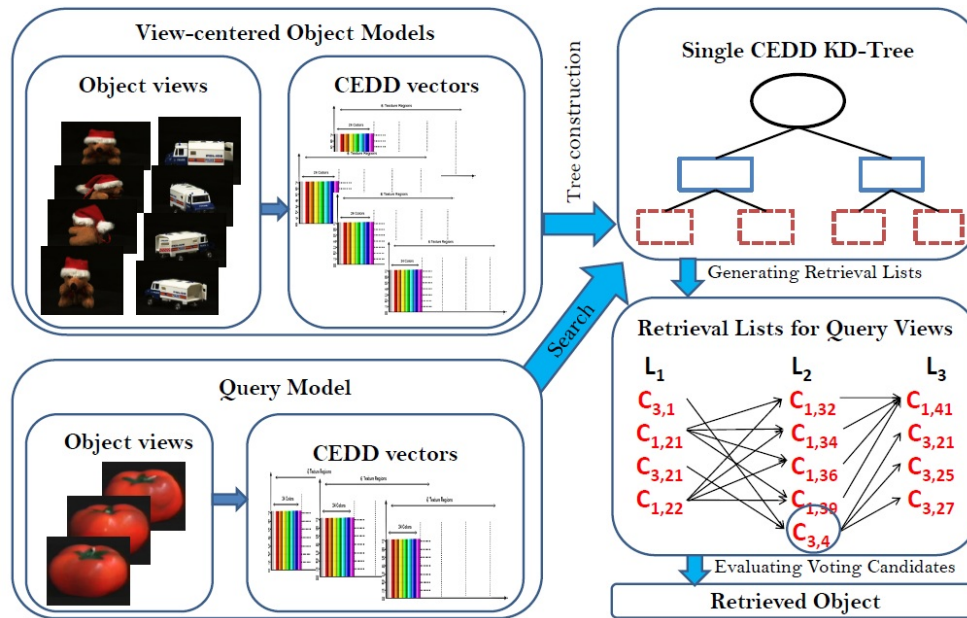


Figure 4.2: Illustration of the VCI approach with three retrieval lists ($N_f^q=3$). Since C_3 was not on L_2 but on L_1 and L_3 , $C_{3,4}$ was added based on $TC(q_i, c_j)$.

retrieval lists. That is we calculate the sum of similarity values of all paths of identical objects and choose the one that has the maximum Hough score with the winner path (see Figure 4.2 and Figure 4.3 for illustration). Since the length of the lists is limited it can easily happen that no representative of an object appears on the lists. If an object is not on any of the lists then simply it is out of focus of our search. However, if it appears on any but missing from some, then those lists are extended with a view of the object, that has the best similarity with the actual query.

Algorithm: VCI search

Input:

- 1- One binary KD-Tree (K) for all CEDD vectors (\mathbf{x}_i) for all object model views in the training database; each tree node k_i contains also two variables k_{max} (the index of the highest value of the variance vector calculated from all vectors $\mathbf{x}_i \in k_i$), and k_{mean} (the mean of vectors in node k_i at k_{max}).
- 2- The CEDD vectors (\mathbf{y}_i) for query Q , i is the index of query views.

Output: Best matching candidate object for query Q .**Operation:**

1. For each vector $\mathbf{y}_i \in Q$, traverse K from root node with \mathbf{y}_i as follows:
 - i. If traversed node k_i is leaf node:
 - (a) For each view $\mathbf{v} \in k_i$, compute distance $T(\mathbf{v}, \mathbf{y}_i)$.
 - (b) Choose the four \mathbf{v} -s with the smallest T distances.
 - (c) Generate retrieval list $L_{\mathbf{y}_i}$ for \mathbf{y}_i , consisting of object views of the \mathbf{v} -s of point (b) above.
 - ii. If traversed node k_i is not leaf node:
 - (a) If $\mathbf{y}_i[k_{max}] \leq k_{mean}$, then traverse to the left child node of k_i , else traverse to the right child node of k_i .
 2. Let R_L be the set of retrieval lists for Q , then evaluate candidate object c_i based on retrieval lists:
 - i. If c_i appears on any retrieval list but missing from some, then those lists are extended with a view of c_i s.t. it has smallest T distance from the query.
 - ii. For each c_i being on the retrieval lists R_L , compute $H(c_i)$ using Equation 4.2.
 3. The object which has highest value is the best matching candidate object for the query Q .
-

Figure 4.3: The pseudo code for VCI search algorithm.

4.3 Experimental Evaluation

4.3.1 Retrieval Performance on SUP-16, COIL-100 and ALOI Datasets

The purpose of the tests were to see the hit-rate of the VCI Best Match approach compared to the method when all models, without real outstanding similarity to any of the queries, were competing (EIS approach). Implementations were tested on previously specified lightweight device. We used three datasets: the SUP-16, the COIL-100, and the ALOI datasets both with distortion free and with strong

distortions (motion blur and Gaussian noise) and with different number of query images ($N_f^q = 1, \dots, 8$) with random selection.

There are three figures illustrating the hit-rate vs. the number of frames in the query. As shown in Figures 4.4, 4.5 and 4.6, as N_f^q goes from 1 to 8 the hit-rate increases about 5%. Comparing the results with the different datasets and distortions we can see a slight advantage of the proposed method against EIS. Especially if the distortion is significant and the dataset is large this advantage is clear and in case of 8 queries VCI is always better.

4.4 Conclusion

In this chapter we introduced a new 3D object retrieval approach, which is based on the Hough framework and using KD-Tree indexing. We compared this method with the approach when all model frames can contribute to compute the average similarity. We analyzed the performance of our approach on several test datasets. While the new approach produces better results in general, the main advantage of this approach will be the straightforward way to integrate the data of orientation sensors as introduced in Chapter 5.

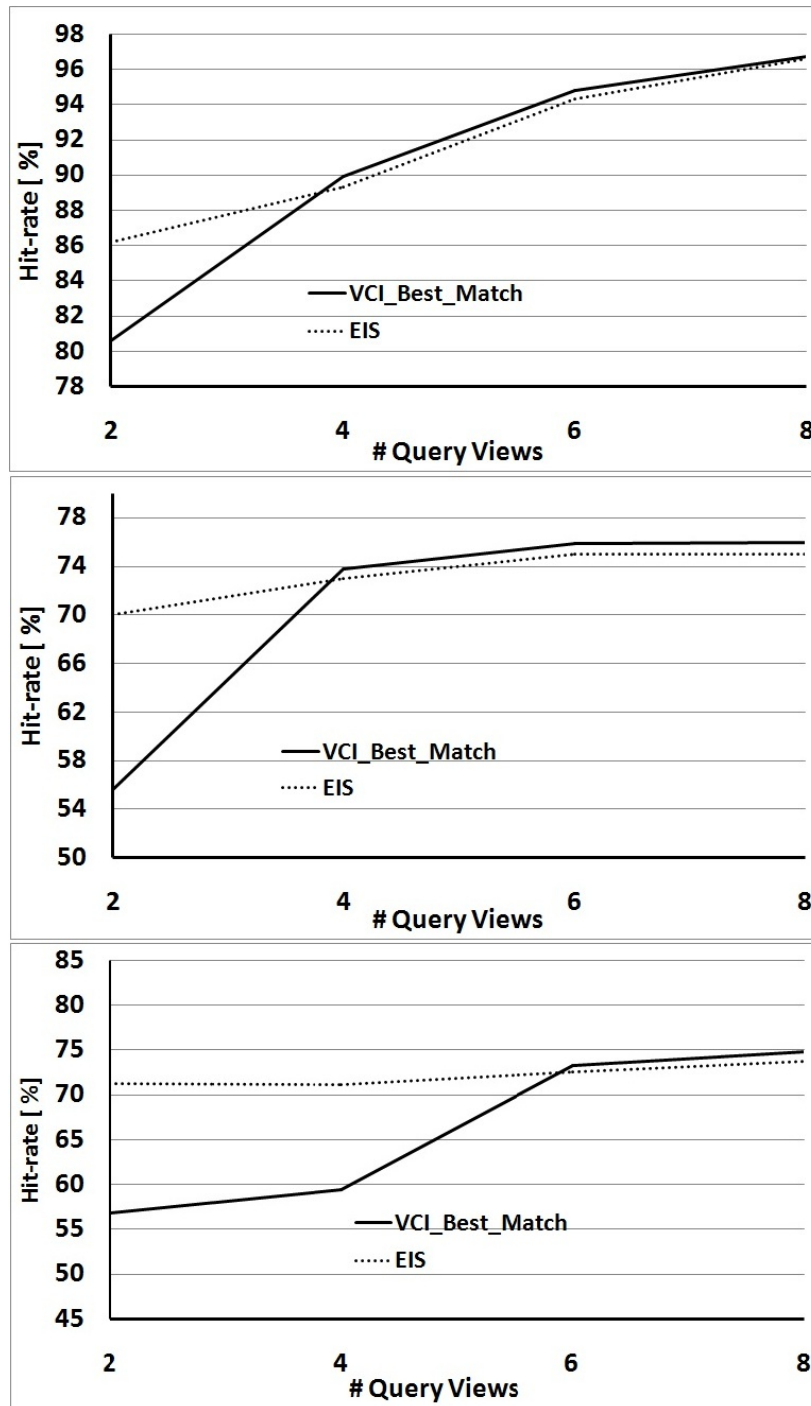


Figure 4.4: Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the SUP-16 database.

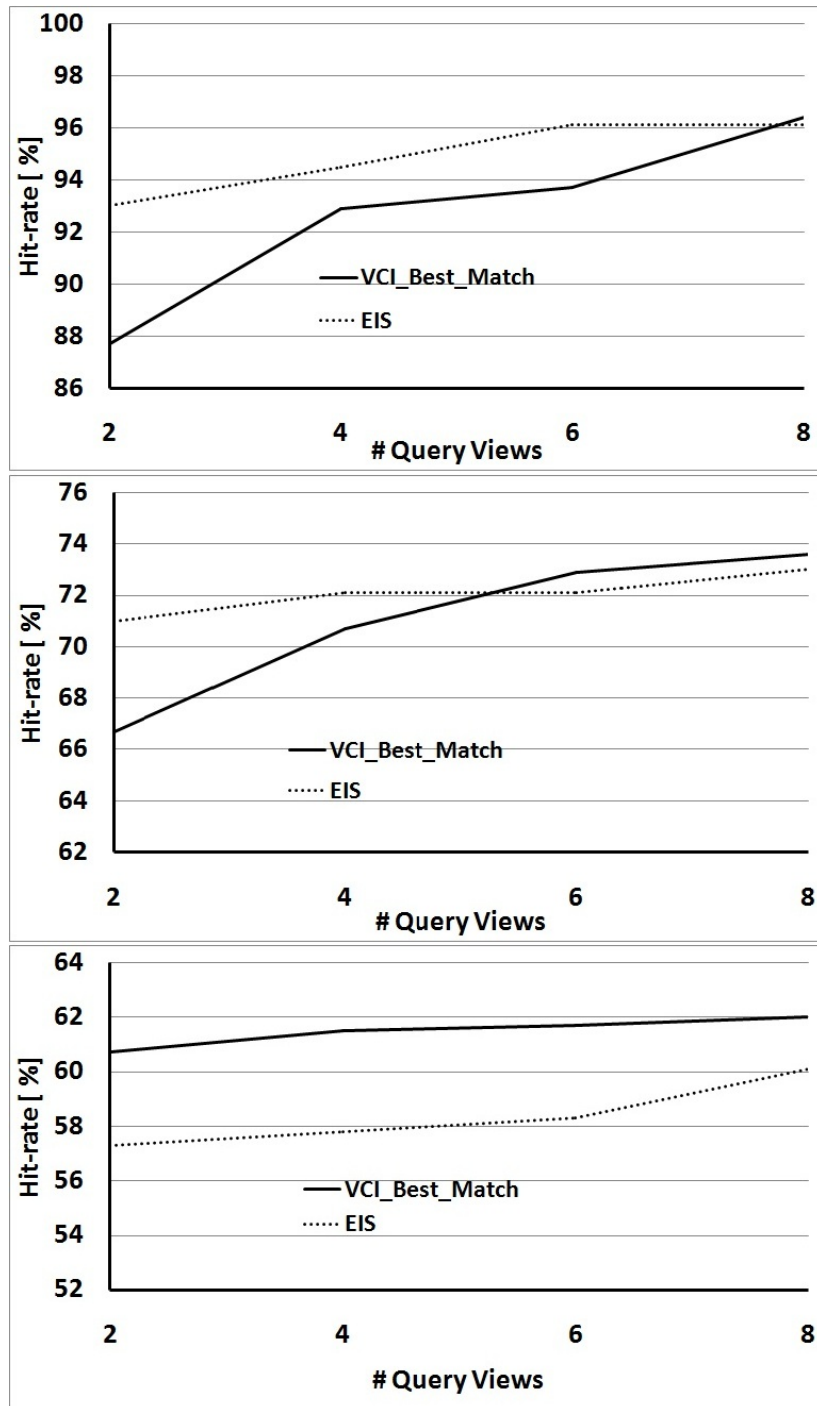


Figure 4.5: Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the COIL-100 database.

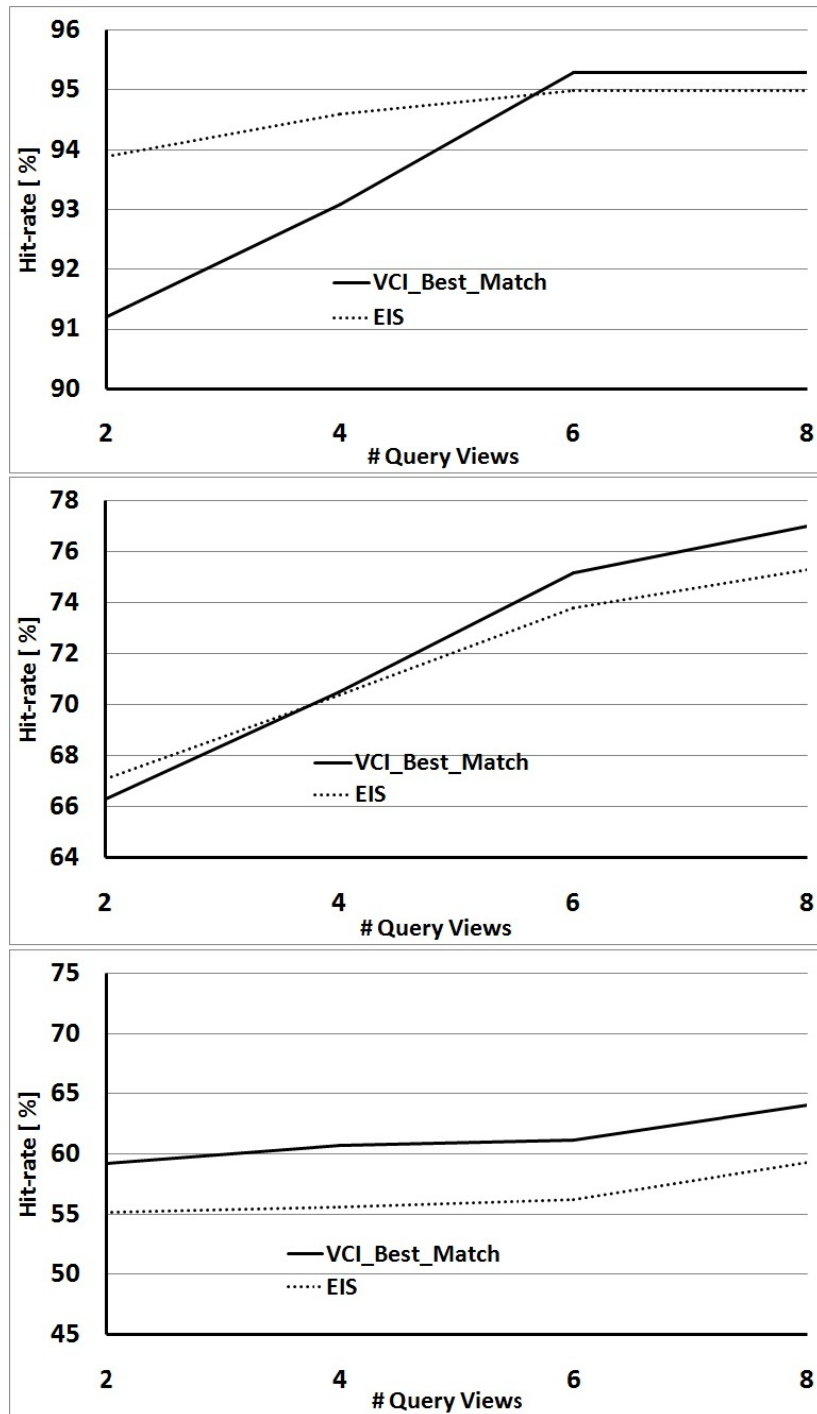


Figure 4.6: Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the ALOI database.

Chapter 5

Orientation Sensors for Object Retrieval and Recognition

In this chapter, we introduce a novel visual retrieval and recognition mechanism involving the camera's orientation sensor. We introduce a sensor-fusion model using the Hough paradigm resulting in significant increase of hit-rate while the complexity remains basically the same. By the extension of the retrieval method we can achieve recognition capabilities and an application in a tracking environment is also described.

5.1 IMU Sensors

An Inertial Measurement Unit, commonly known as IMU [44], is an electronic device that measures and reports magnetic force, velocity, acceleration, and gravitational forces through the use of accelerometers, gyroscopes, and magnetometers. IMUs are main components of inertial guidance systems used in air space, and watercrafts, including guided missiles. Nowadays, the usage has been extended to other consumer applications such as motion capture systems, gaming, visual effects, hand

held devices, gesture recognition and navigation. In virtual reality systems, sometimes IMUs are used inside the controller to perform motion tracking, sensing and capturing.

IMUs can estimate motion including the type, rate, and direction of that motion using a combination of accelerometers, digital compass, and gyroscopes. Measured data are fed into a computer, to calculate the current speed and position, given the initial speed and position.

IMUs are now available in the market in various types and with different precision. A main parameter is the degree of freedom (DOF): For 3 DOF, the sensor contains two accelerometers and a gyroscope that measures yaw. For 5 DOF, the sensor has three accelerometers and two gyroscopes that measure pitch and roll. For 6 DOF, all axes for the accelerometer and gyroscope are available.

Luckily, today's mobile computing devices often contain inertial measurements units. Moreover, these devices can be used for the calibration of cameras. However, it is still an open question how to exploit the IMUs in video retrieval and recognition without camera calibration and also without going through the structure from motion reconstruction methodology.

Since the position of the target object is not fixed (not only buildings or statues are being recognized) relative orientation (degree) information between the object views is to be estimated and to be exploited. The accuracy and precision of today's IMU sensors allow their application in many fields (for an overview of these see [1]) including our framework. Figure 5.1 shows the distribution of absolute orientation error measured in 6 cycles (8 positions checked in each) with our tablet. While in most of our experiments we used the same device, in case of experiments with the SMO, COIL-100 and ALOI datasets we used the above noise distribution to simulate IMU noise.

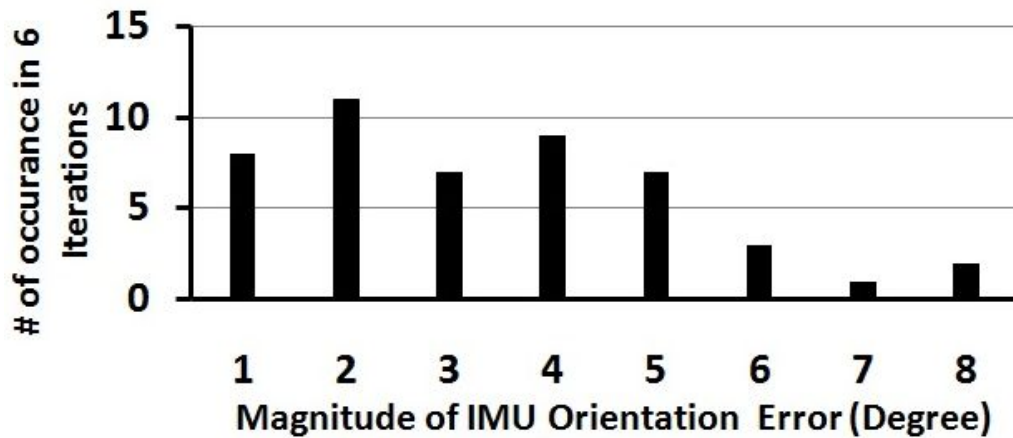


Figure 5.1: IMU measurement error distribution.

5.2 Object Retrieval with IMU

In this section we introduce various retrieval mechanisms which use orientation sensors to help the visual retrieval process. We apply a view based model of the objects and the matching of the query and candidate images is based on compact image descriptors coupled with relative orientation.

Our research is focused on view-centered models where information about the relative position of the target object and the camera is exploited. Preliminary experiments already showed (see [19] and [18]) that IMUs can help the recognition process with low computational demands. However, the problem caused by low quality query images, fast object tracking and/or segmentation still can be a problem in this framework being a subject for research.

5.2.1 SIIS (Selected Image and IMU Search) Approach

In this approach we show that testing only one (selected) frame from the query against all model frames then using the known relative orientation for the evaluation of other query frames results in much lower complexity without sacrificing retrieval

rate. That is we define the following distance function between a query q and candidate c_i :

$$T^{SIIS}(q, c_i) = \frac{\min_f T(q_k, c_{i,f}) + \sum_{\forall l, l \neq k} T(q_l, c_{i, \Delta\alpha(k)})}{N_f^q} \quad (5.1)$$

where $c_{i, \Delta\alpha(k)}$ means the candidate model frame from object i which has the same (or very close) relative orientation difference $\Delta\alpha$ from $\operatorname{argmin}_f T(q_k, c_{i,f})$ as frame l from frame k in the query. That is first we find the best matching frame of a selected query frame in candidates based on visual information then compare the visual descriptor of other c_i frames found at the same (or closest) orientation positions. The selection of the query frame, used for extensive visual search, can be based on its quality, information content, or time order; in evaluation experiments we used a randomized selection.

The complexity of this approach can be described as: $O(N_c * (N_f^c + 2 * (N_f^q - 1)))$. Since there is no guarantee that we find a frame at the exact relative position in the candidate model we use the best matching of the left and right neighbors in the closest available orientations (this explains the multiplication by 2 in the complexity equation). Please see Figure 5.2 for illustration.

5.2.2 EIIS (Extensive Image and IMU Search) Approach

While the previous approach (SIIS) can be vulnerable to the only query frame used to find the right view in each candidate, by involving all queries in the extensive search we can utilize all available information of the query. Thus we test all frames from the query against all views of the candidate but when computing the distance we keep the constraint that the frames should be in the same relative orientation in

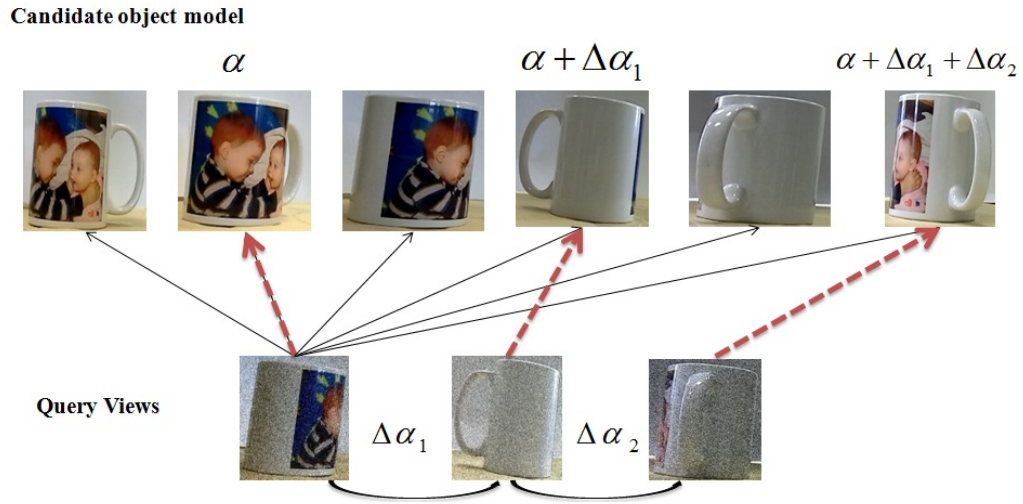


Figure 5.2: Illustration of the SIIS approach: after finding the best matching frame of a query and a candidate, other frames are also compared selected on the bases of their similar relative positions as the frames of the query.

the candidates and in the query. The distance is then:

$$T^{EIIIS}(q, c_i) = \frac{\min_{c_i, \Delta\alpha(k)} \sum_{\forall k} T(q_k, c_i, \Delta\alpha(k))}{N_f^q} \quad (5.2)$$

where $c_{i, \Delta\alpha(k)}$ denotes the candidate model frame which has the same (or very close) relative orientation $\Delta\alpha$ as k frame in the query. The complexity of the EIIIS approach can be described as $O(N_c * N_f^q * N_f^c)$.

5.2.3 VCI Approach with IMU

Now we describe how to insert relative orientation information to the VCI method described previously. VCI generates limited length retrieval list for all query views. All the retrieved candidates give votes based on visual similarity and relative orientation. (The relative orientation of subsequent queries should match the relative orientation of subsequent candidate frames). As the Tanimoto Coefficient measures similarity between the query and the actual candidate $TC(q_i, c_{j,f})$ will be one term

of the vote, while the weighted term will be responsible for the orientation difference: $|\Delta\beta_{i,k,l} - \Delta\alpha_k|$ where $\Delta\beta_{i,k,l}$ is the difference between the orientation values of two frames of candidate object i :

$$\Delta\beta_{i,k,l} = \alpha(c_{i,k}) - \alpha(c_{i,l}), \quad (5.3)$$

and $\Delta\alpha_i$ is the difference between the orientation values of consecutive queries:

$$\Delta\alpha_i = \alpha(q_i) - \alpha(q_{i+1}). \quad (5.4)$$

So for the Hough Score, we modify the Equation 4.2 as follows:

$$H(c_i) = \max_{\mathbf{j} \text{ s.t. } c_{i,\mathbf{j}_k} \in L_k} \left(TC(q_{N_f^q}, c_{i,\mathbf{j}_{N_f^q}}) + \sum_{k=1}^{N_f^q-1} (TC(q_k, c_{i,\mathbf{j}_k}) - w |\Delta\beta_{i,\mathbf{j}_k \mathbf{j}_{k+1}} - \Delta\alpha_k|) \right), \quad (5.5)$$

where w is a constant weight.

Considering Figure 4.2, we should add IMU data to object models and to queries. As C_3 was not on L_2 but on L_1 and L_3 , $C_{3,4}$ was selected to be added based on its best matching relative orientation (this means a change at point 2.i of Figure 4.3). In this approach, we compute $H(c_i)$ (2.ii of Figure 4.3) using Equation 5.5, where Equation 2.4 will be evaluated over all paths (connecting the same objects) on the subsequent retrieval lists.

The complexity of the VCI approach now changes to: $O(N_f^q * N_f^{leaf}) + O'(N_L, N_f^q)$, where N_f^{leaf} is the number of frames in the KD-Tree leaf node (typically around 14). Please note that the complexity of the combinatorical evaluation of possible paths through the retrieval list can be high (exponentially increasing for N_f^q) so it is given by O' , since not being on the core bases of comparing two CEDD descriptors. That is,

Table 5.1: Complexity of different approaches at $N_c = 16$, $N_f^c = 50$, $N_f^{leaf} = 14$, and different N_f^q .

Method	$O()$	Query Views (N_f^q)			
		2	4	8	16
EIS, EIS, VC	$O(N_c * N_f^q * N_f^c)$	1600	3200	6400	12800
SIIS	$O(N_c * (N_f^c + 2 * (N_f^q - 1)))$	832	896	1024	1280
VC + KD-Tree (VCI)	$O(N_f^q * N_f^{leaf}) + O'(N_L, N_f^q)$	$28 + O'$	$56 + O'$	$112 + O'$	$224 + O'$

besides the rough estimations summarized in Table 5.1, we need time measurements to get a better picture of time complexity as will be given in Section 5.5.6.

5.2.4 Zero Weight Case

To investigate the effect of IMUs we introduce a special case, where the value of the weight w is *zero* ($w = 0$). Orientation information was used only to extend some of the retrieval list where any object (seen on any other retrieval list) is missing (2.i of Figure 4.3). This case will be named as VCI W in the forthcoming figures.

5.3 Object Recognition with IMU

In many applications it is possible that untrained objects are targeted to be recognized. To avoid false recognition we have to evaluate the confidence of the best hypothesis found by the VCI method. If we find the confidence of the best candidate low, we should reject it and declare that the views of an unknown object were captured. For this purpose we extended the object models to store the CEDD differences (Equation 2.2) with c_j^{Noise} the views loaded with distortion (resulting in matrices of size $N_f^c * N_f^c$).

Now, we are able to estimate a view-dependent average “typical” distance of

model views and noisy approximations:

$$T^{Typ}(c_{i,(l,m,\dots)}) = \frac{1}{N_q^f} \sum_{\forall k \in (l,m,\dots)} T(c_{i,k}, c_{i,k}^{Noisy}) \quad (5.6)$$

A hypothesis can be tested by thresholding:

$$Decision = \begin{cases} \text{Accept candidate} & \text{if } T^M / T^{Typ} \leq Th \\ \text{Reject candidate} & \text{if } T^M / T^{Typ} > Th \end{cases} \quad (5.7)$$

where T^M is the average distance of the query and the candidate according to the views on the retrieval lists of the VCI method. The reader should be aware that this confidence model is orientation adaptive: *i.e.* some views of objects can be modeled with high sensitivity to noise (if T^{Typ} is relatively small) while other views of the same object with low sensitivity if (T^{Typ} is relatively large). Please note, that this post-filtering step will be evaluated only in the tests of Subsection 5.5.3.

5.4 Active Vision in the Hough Framework

Active recognition is a relatively old idea in pattern recognition, and non-active methods have been extended many times. For example, the work [9] can be considered as an extension of [58] in that a viewing position that minimizes the average entropy was chosen. Without mentioning many of such techniques, we refer to the survey in [68] and shortly discuss a few new results which aim at lightweight recognition.

In [11], hypotheses about objects in the hand of an iCub robot were created and updated as the recognition progressed. They adopted probabilistic Monte Carlo localization methods to maintain a high number of hypotheses in parallel, and they

used particle filtering, regarding hypotheses, to take into account the viewpoint changes in the form of proprioceptive information obtained from the robot arm. Unfortunately, the tests included only six real objects, so a real-life evaluation of the proposed strategy was not presented.

The method used in [39] used SIFT features for active object recognition and verification. They created an automatic viewpoint selector that uses a vocabulary tree structure to weigh the uniqueness of each feature in a viewpoint. Every viewpoint was then given a value that was obtained by summing the uniqueness measure of all its features. This mechanism is used to select the subsequent view. While the experiments, with only a few test objects, showed that the view selection mechanism is successful, the running time and other practical questions (the effects of noise, the estimation of orientation, and cases with large datasets) were not discussed.

In [62], a method for active recognition using an RGB-D (Red Green Blue - Depth) camera mounted on a quadcopter, using an object bounding box, color, SIFT, and a viewpoint feature histogram for recognition, is discussed. The authors define a utility score for a particular action, which can be computed by mutual information (MI). MI is used to reduce the uncertainty of the current object's class and its pose if a new observation is made. If a prior map of the environment is given, the change of the object class or the change of its orientation can be detected.

By contrast, our model works with relative orientation since the object can be moved and rotated. Additionally, our approach is designed to be lightweight and thus uses compact visual descriptors and the scalar orientation data often available from low-cost IMU (Inertial Measurement Unit) sensors. Noise tolerance and efficiency was found to be achieved with a series of observations (rather than with deeper processing of complex data). New viewpoints could be chosen in a way to serve with discriminating data.

The main contribution of this subsection is showing that the proposed Hough framework can be used for active perception and thus that very lightweight techniques can be efficiently used for 3D object retrieval.

5.4.1 Active Retrieval to Minimize Ambiguity

Active vision systems can be classified, according to their next view planning strategy, into two groups:

1. systems that take the next view to minimize an ambiguity function;
2. systems incorporating explicit planning algorithms.

We have chosen the first strategy and here introduce a method that is very close to the way in which humans would naturally move around an object to become acquainted with its appearance from different directions to be able to recognize it. Based on a rapid evaluation of the first observations, they hypothesize which objects have high probability to appear and plan their movement to find those views that can reduce ambiguity. Based on the preliminary models, the $N_f^{\tilde{c}}$ average views of object i from the descriptors is computed within a viewing range (each containing N^k views):

$$\tilde{c}_{i,k} = 1/N^k \sum_{l=1}^{N^k} c_{i,l} \quad (5.8)$$

It is important that $N_f^{\tilde{c}} \ll N_f^c$, which means that each object is now represented by only a few views, computed as average CEDD vectors that equally divide the circle into $N_f^{\tilde{c}}$ parts.

This is necessary to reduce the amount of computations when comparing the different candidates for next view planning. Afterward, the similarity between these average views can be computed with the Tanimoto coefficient (Equation (2.1)) and

can be stored in matrix S of size $N_f^{\tilde{c}}N_c \times N_f^{\tilde{c}}N_c$, where N_c gives the number of all possible objects.

After making the very first observations, we are to evaluate the retrieval lists L_i just as described above in subsection 5.2.3. As $\alpha(c_{j,k})$ provides the estimate of orientation for all $c_{j,k} \in L_i$, we can also compute the similarity of views to the left (and to the right accordingly):

$$S_{left} = \sum_{c_j, c_l \in L_i, j \neq l} T(\tilde{c}_{j,left}, \tilde{c}_{l,left}) \quad (5.9)$$

where $\tilde{c}_{j,left}$ is the closest \tilde{c}_j view left of $\alpha(\tilde{c}_{j,k})$.

Finally, we should move the sensor either to the left or to the right depending on the dissimilarity of views of the possible candidates:

$$Decision = \begin{cases} \text{Move to left} & \text{if } S_{left} \leq S_{right} \\ \text{Move to right} & \text{if } S_{left} > S_{right} \end{cases} \quad (5.10)$$

We call this approach Active VCI (AVCI). Its performance will be compared to the non-active version.

5.5 Experimental Evaluation

In this section, we introduce the average hit-rate of the different methods and the measured time complexity.

5.5.1 Retrieval Performance on SUP-16, COIL-100 and ALOI Datasets

The purpose of these experiments is to show the advantage of the retrieval performance of the VCI over the other approaches. To be realistic we removed not only the query images from each model during testing: we deleted every view from the models within 10° angle from a possible query. (We have chosen 10° since it is slightly above the accuracy of the IMU sensor as shown in Figure 5.1.) Since the queries were randomly selected the closest orientation angle between a query and an available model view was observed to be between 10 and 30 degrees. Thus the size of the model sets reduced to 34% for the SUP-16, 78% for the SMO, 79% for the COIL-100, and to 69% for the ALOI database.

In tests with COIL-100 and ALOI the IMU data of the queries were loaded with the noise given in Section 5.1. We tested all approaches using three datasets: the SUP-16 dataset, the COIL-100, and the ALOI dataset, all with distortion free and with strongly distorted queries (MB and GN). The SMO database was used only for the evaluation of the VCI approach (see next section).

Figure 5.3, Figure 5.4, and Figure 5.5 contains the hit-rates at different N_f^q -s. As it is easy to see the four basic methods run close to each other; VCI_Best_Match, VCLW, SIIS, EIIS, and VCI is the increasing order. That is the orientation information, included into the different models, could add valuable information to improve the hit-rate of these methods. Our proposed VCI technique overcomes all other significantly except for only very few cases. Higher number of queries results in higher hit-rates unexceptionally and GN makes more problem for the descriptor than MB.

We also evaluated the role of w in Equation 5.5. Figure 5.6 contains information for only the ALOI dataset. It seems that the optimal value for w is somewhere

between 0.1 and 0.5 and comparing with the case $w = 0$ the maximum improvement is below 4%. That is a large amount of the contribution of orientation information is not via the orientation term of Equation 5.5, but by the fact that the retrieval lists are occasionally extended by missing candidates based on the relative orientation of the actual view.

5.5.2 Retrieval Performance on the SMO Dataset with Various Backgrounds

To give comparisons of the proposed VCI approach with other methods we got access to the raw dataset of article [12] called SMO (as described in Chapter 3). This article uses SIFT and feature tracking for the recognition of objects in videos. Since the queries of the tests of [12] were not available for us and our method has no fully automatic target selection we can give only rough comparisons of the performance of the two approaches.

In order to get similar queries as in [12], the 10 randomly selected images of each object were tested with changed uniform or textured backgrounds. Moreover, besides distortion free test images, queries were strongly distorted with motion blur or additive Gaussian noise as specified previously. Figure 3.10, and Figure 3.11 in Chapter 3 illustrates the queries with the uniform and textured backgrounds respectively, and Table 5.2 contains the comparisons of the testing conditions.

The average hit-rate of the VCI method, for various settings, can be found in Table 5.3. As we experienced earlier, the increase of number of query images (N_f^q) results in significant increase of hit-rate (the positive change is the range from 4.8% to 20.4%) while the effect of placing the object on various backgrounds make difference from 2.4% to 15.6%. It is interesting, but not surprising, to see that while the hit-rate is higher for uniform backgrounds the improvement is greater for the

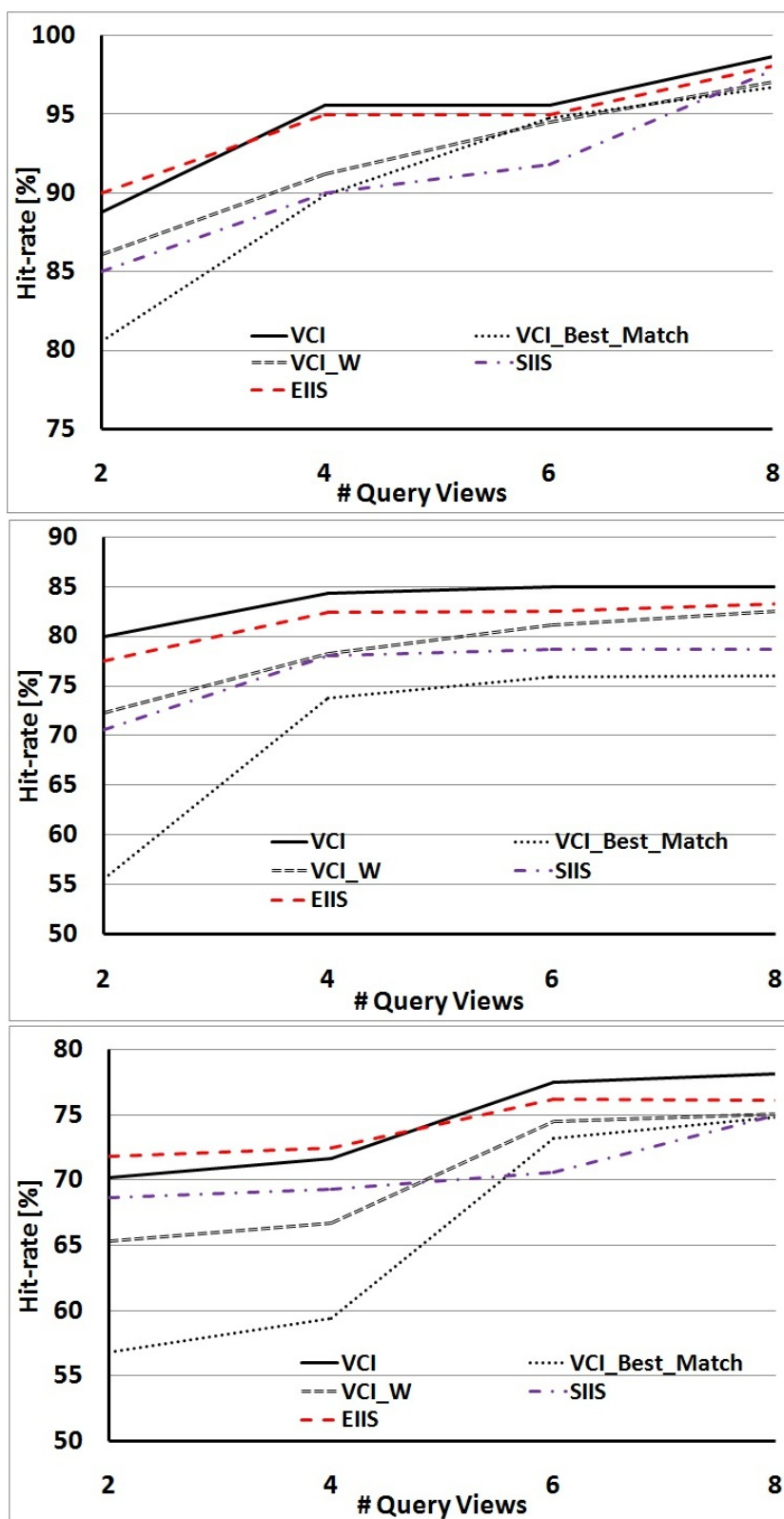


Figure 5.3: Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the SUP-16 database and for VCI $\omega = 0.5$.

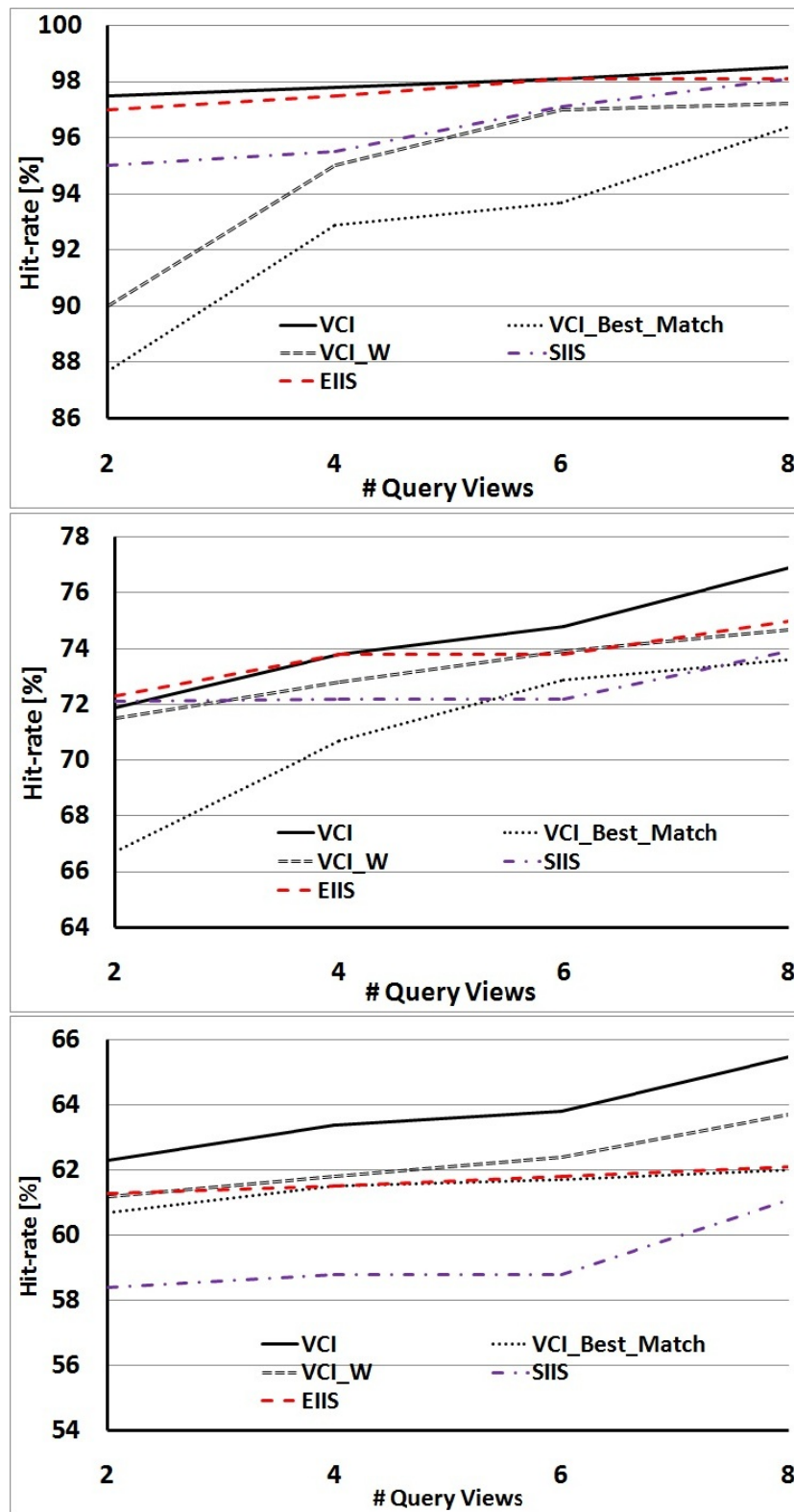


Figure 5.4: Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the COIL-100 database and for VCI $\omega = 0.5$.

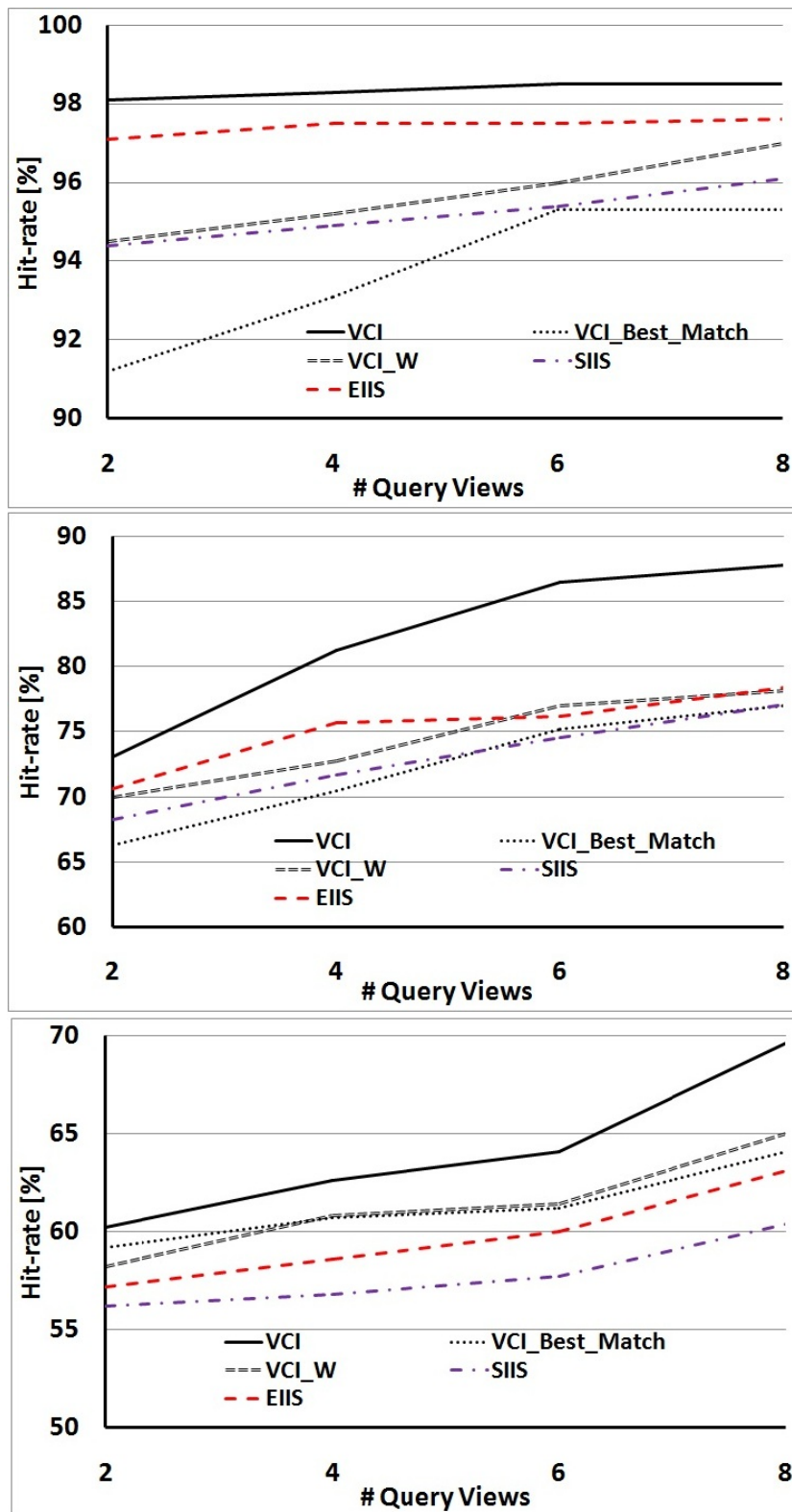


Figure 5.5: Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the ALOI database and for VCI $\omega = 0.5$.

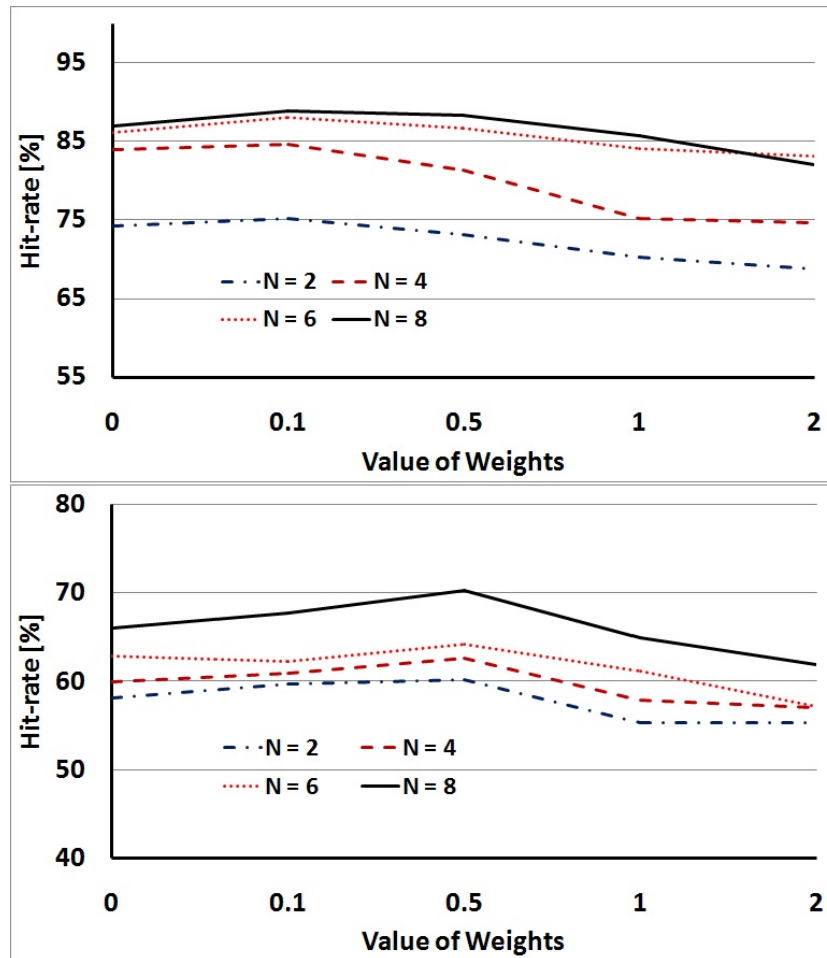


Figure 5.6: Average hit-rate of the VCI method for motion blur (top) and additive Gaussian noise (bottom) at different query views (N) and w settings for the ALOI dataset.

Table 5.2: Comparison of experimental setups of [12] and VCI approach.

	Tests of [12]	VCI approach
Candidate database	SMO (25 objects)	SMO (25 objects)
Number of objects on the query image	more than one object	only one object on image
Object environment	randomly placed object	background is changed to uniform / textured
Number of views tested	4	2 / 4 / 6 / 8
Image noise on queries	No	No / Gaussian / motion blur

Table 5.3: Average hit-rate for distortion free (DF), motion blur (MB) and additive Gaussian noise (GN) with uniform (UB) and textured backgrounds (TB) on the SMO dataset.

Dataset Variations	Query Views (N_f^q)			
	2	4	6	8
DF-UB	94.4%	98%	98.4%	99.2%
DF-TB	78.8%	90.4%	94.4%	96.8%
MB-UB	84.8%	93.6%	95.6%	96%
MB-TB	69.2%	80%	85.6%	88.8%
GN-UB	82.4%	89.6%	92.4%	93.6%
GN-TB	69.2%	80.4%	86%	89.6%

textured cases. In [12] there is data only for the 4 queries case with about 80% success. Please note, that all of our data are above this performance at $N_f^q = 4$: hit-rate is spread between 80% and 98%.

5.5.3 Recognition Performance with the Extended SUP-16 Dataset

To achieve recognition instead of retrieval we should evaluate the confidence of the best candidate and make a decision about a possible false acceptance (see Section 5.3). To test the performance of our approach from this aspect we added 9 untrained objects, taken from the SMO dataset, to SUP-16 (see some examples on Figure 5.7) and applied MB for the queries.

Figure 5.8 shows the results for different Th thresholds of the post-filtering method. As would be expected the results are worse than for retrieval with the 16 objects. Now, a hit was counted only when an object was correctly recognized or an untrained object was recognized as unknown.

In these tests the increase of the number of query views had also a positive effect on the hit-rate giving the best results at $Th = 2$.



Figure 5.7: Examples for untrained objects, taken from the SMO dataset, to test recognition performance.

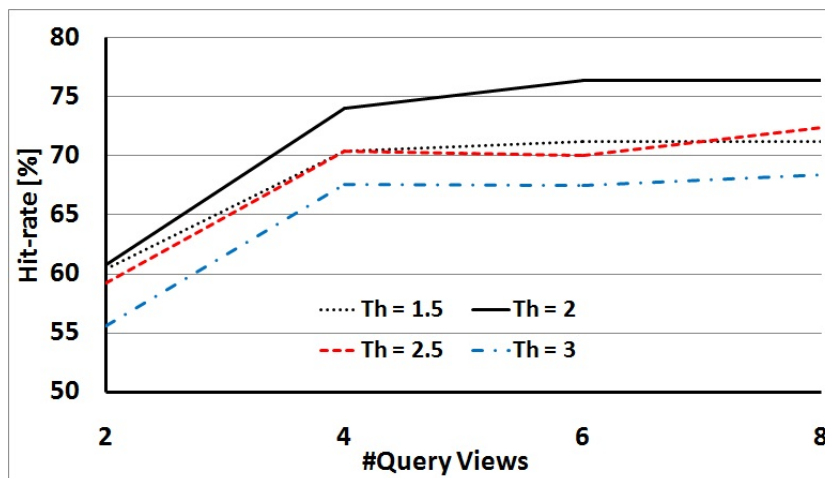


Figure 5.8: Average hit-rate for the SUP-16 dataset with 9 extra untrained queries with different thresholds.

5.5.4 Retrieval Performance with Automatic Segmentation on the SUP-25 Dataset

In some human operated real-life applications users are to select a target object then the application should make the queries automatically from different views. Queries for the experiments with SUP-25 were recorded with the tablet. The user

was asked to mark the target object with a bounding box on the live image then continuous adaptive mean-shift was to track it when moving the camera around the object (typically 90° change in viewing direction at the same elevation). While the targeted objects were not occluded they were surrounded with others and the background was different than in the models.

The automatic tracking can adjust the position and the size of the bounding window by using invariant moments, these windows are given to the query engine. Figure 3.3 illustrates some tracked windows and the environment of the objects. Table 5.4 contains the hit-rates for different number of queries telling that even a simple tracking algorithm was successful to generate queries to reach almost 100% for $N_f^q = 8$. Comparing these results to those of Figures 5.3, 5.4 and 5.5 we find that our previous estimations were not far from these real-life tests.

Table 5.4: Hit-rate of VCI with object tracking for SUP-25.

Method	Query Views (N_f^q)			
	2	4	6	8
VCI	72%	84%	92%	96%

5.5.5 Retrieval Performance with Active VCI on the COIL-100 Dataset

AVCI is a straightforward extension of VCI: since the evaluation of the weak classifiers is independent and continuous, the direction of a new viewpoint can be arbitrarily selected. In the presented experiments the first 2 queries were searched by the VCI algorithm resulting in a small set of candidates to be evaluated by AVCI, and to propose a new position. There is a high probability that this set of initial selections contains the right candidate. In Figure 5.9 the data lines called 'Best 10' are the hit-rates counted so that the right candidate is within the first best 10 elements

after the evaluation of Equation 5.5. This curve is from 5% to 14% above the best result of VCI, meaning that VCI is close to the good solution many times but the randomly selected views are not satisfactory to make the good choice. That gives us a chance that by a better selection of views we can increase the discriminating power of the method. Figure 5.9 contains the hit-rates for AVCI, which are, while below the "Best 10" as expected, significantly better than for VCI. Data can also be interpreted in a way that after 4 queries of the active method the results are as good or even better than for VCI at $N_f^q = 8$, so to achieve the same accuracy less observation is needed. Naturally, orientation measurements were loaded with noise, and all query images were also distorted in the tests; where $N_f^c = 4$.

5.5.6 Running Times

As we have seen it is not convenient to evaluate the complexity of the methods since the different parts of the algorithms react differently for the various parameters. For this reason we implemented them on a tablet and made time measurements. Tests were run on a lightweight device (specified in Subsection 3.3). Figure 5.10 contains the average running time of 10 measurements on a database of 100 objects.

Only the retrieval mechanisms are considered in this graph for comparison, the generation of CEDD descriptors (which is about 0.04 sec for a frame of size 640x480) and the tracking algorithm (0.7 sec per frame) is not included. As expected from Table 5.1 SIIS has low running time just as VCI for low N_f^q . However, while SIIS increases almost linearly with N_f^q , VCI grows exponentially due to the $(N_L)^{N_f^q}$ combinatorial evaluations of possible paths through the retrieval lists.

Finally, we can conclude that our VCI code (without special code optimization or code parallelism) can achieve slightly below 8 sec / 8 queries, for a database of 100 objects, resulting in practically real-time operation on a mobile device (for SUP-16

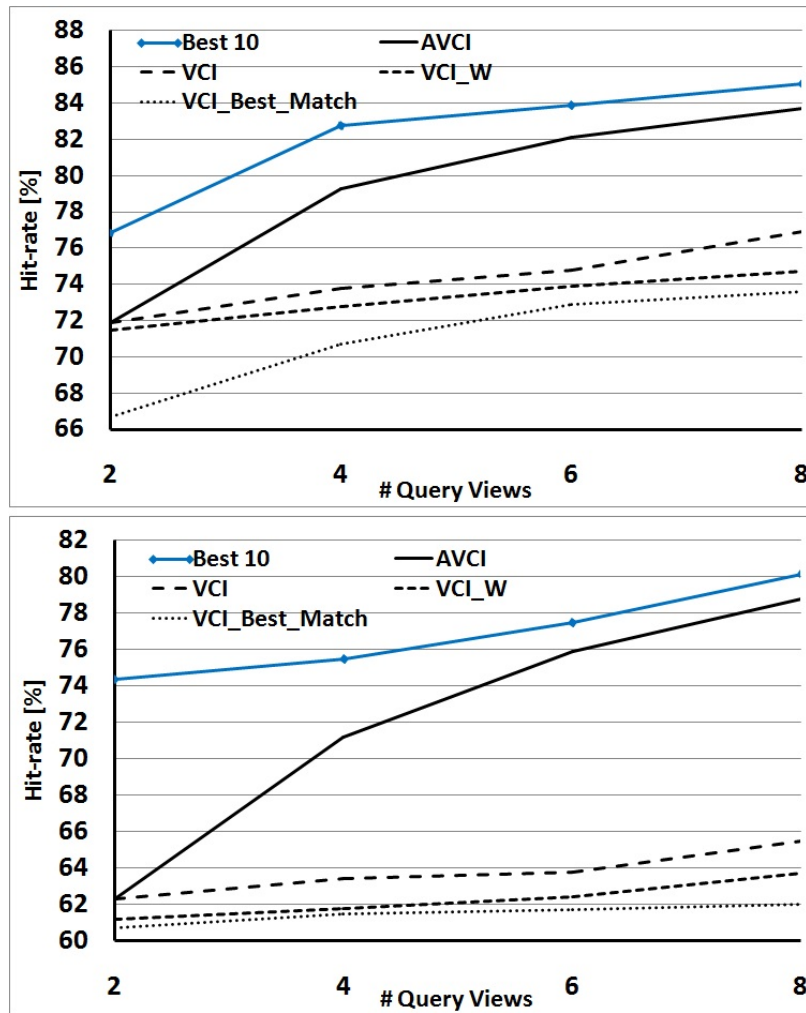


Figure 5.9: Average hit-rate obtained over the COIL-100 dataset with the VCI_Best_Match, VCI_W, VCI, AVCI, and with the “Best 10”: queries with motion blur (top); queries with additive Gaussian noise (bottom).

it is 0.34 sec / 8 queries). The memory space required for a database of 100 objects is around 1.2MB.

5.6 Conclusion

In this chapter, we introduced new 3D object retrieval and recognition approaches where besides cameras, Inertial Measurement Unit (IMU) sensors are used for the

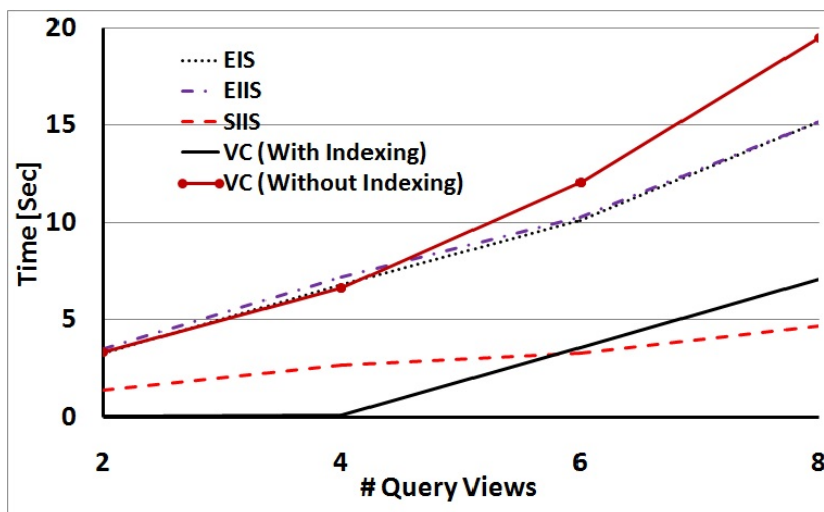


Figure 5.10: Average running time for EIS, SIIS, EIIS, VC and VCI approaches on a dataset of 100 objects.

retrieval and recognition of 3D objects. Contrary to computationally intensive deep learning recognition and retrieval solutions we focused on lightweight methods which could be utilized in handheld devices and autonomous systems equipped with moderate computing power and memory.

We used fast and robust compact image descriptors and the relative orientation of the camera to build multi-view-centered retrieval object models. As for retrieval and recognition the Hough transformation paradigm was used to evaluate the results of queries applied on several frames of a video. We analyzed the performance of our lightweight approaches on several test datasets and with different comparisons, including automatic tracking for the capturing of queries. These experiments show the advantages of our proposed techniques since retrieval and recognition rates could be significantly increased.

In Chapter 6 we will show a new 3D object retrieval model by using a Hidden Markov Model (HMM) framework where 2D object views correspond to states, observations are coded by compact edge and color sensitive descriptors, and orientation

sensors are used to secure temporal inference by estimating transition probabilities between states.

Chapter 6

View Centered Object Models using Hidden Markov Model

In this chapter, we introduce a new 3D object retrieval model incorporating the following mechanisms: viewer-centric recognition, Markovian estimations, and fusion of information originating from the visual and orientation subsystems. We have built a Hidden Markov Model (HMM) framework where 2D object views correspond to states, observations are coded by compact edge and color sensitive descriptors, and orientation sensors are used to secure temporal inference by estimating transition probabilities between states. Our evaluation results, over different databases, are very good: the fast and memory efficient new method outperformed all previous models.

6.1 Object Retrieval with HMM

To achieve object retrieval will need to build HMM models for all elements of the set of objects (M). Then, based on observations, we find the most probable state sequence for all objects models. The state sequence among these, which is the most similar to the observation sequence, will belong to the object being retrieved.

6.1.1 Object Views as States in a Markov Model

Let $S = \{S_1, \dots, S_N\}$ denote the set of N hidden states of a model. In each t index step this model is described as being in one $q_t \in S$ state, where $t = 1, \dots, T$.

In our approach the states can be considered as the 2D views (or the average of some neighboring views) of a given object model. This can be easily imagined as a camera is targeting towards and object from a relative elevation and relative azimuth. The number of possible states should be kept low, otherwise the state transition matrix (\mathbf{A}) would contain too small numbers and finding the most probable state sequence could be too unstable.

On the other hand, small number of states would mean that quite different views of objects should be represented by the similar descriptors, resulting in too much generalization. Thus it is easy to see that the generation of states should be designed carefully. Often Gaussian mixtures are used to combine the views of similar directions. Now we use static subdivision of the circle of 360° , into 2, 4, 6, and 8 uniform parts with 180° , 90° , 60° , and 45° sectors correspondingly. We define the initial state probabilities $\pi = \{\pi_i\}_{1 \leq i \leq N}$ based on the orientation range of states:

$$\pi_i = P(q_1 = S_i) = \frac{\alpha(S_i)}{360} \quad (6.1)$$

where $\alpha(S_i)$ is the size of orientation aperture of state S_i given in degrees.

6.1.2 State Transitions

Between two steps the model can undergo a change of states according to a set of transition probabilities associated with each state pairs. In general the transition probabilities are defined as follows:

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \quad (6.2)$$

where i and j indices refer to states of the HMM, $a_{ij} \geq 0$, and for a given state $\sum_{j=1}^N a_{ij} = 1$ holds. The transition probability matrix is denoted by $\mathbf{A} = \{a_{ij}\}_{1 \leq i, j \leq N}$

To build a Markov model means learning its parameters (π , \mathbf{A} , and emission probabilities) by examining typical examples. However, our case is special: the probability of going from one state to an other severely depends on the users's behavior, interest and also on the frame rate of the camera. Thus we can not follow the traditional way, to use the Baum-Welch algorithm for parameter estimation based on several training samples. Contrary, thanks to the orientation sensors, we can directly compute transition probabilities based on geometric probability as follows.

First define $\Delta_{t-1,t}$ as the orientation difference between two successive observations:

$$\Delta_{t-1,t} = \alpha(o_t) - \alpha(o_{t-1}) \quad (6.3)$$

Now define R_i as the aperture interval belonging to state S_i by borderlines:

$$R_i = [S_i^{min}, S_i^{max}[\quad (6.4)$$

The back projected aperture interval is the range of orientation from where the previous observation should originate:

$$L_j = [S_j^{min} - \Delta_{t-1,t}, S_j^{max} - \Delta_{t-1,t}[\quad (6.5)$$

Now we have arrived to estimate the transition probability by the geometrical probability concept applied on the intersection of L_j and R_i :

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) = \frac{\alpha(L_j \cap R_i)}{\alpha(L_j)} \quad (6.6)$$

Please see Figure 6.1 for illustration.

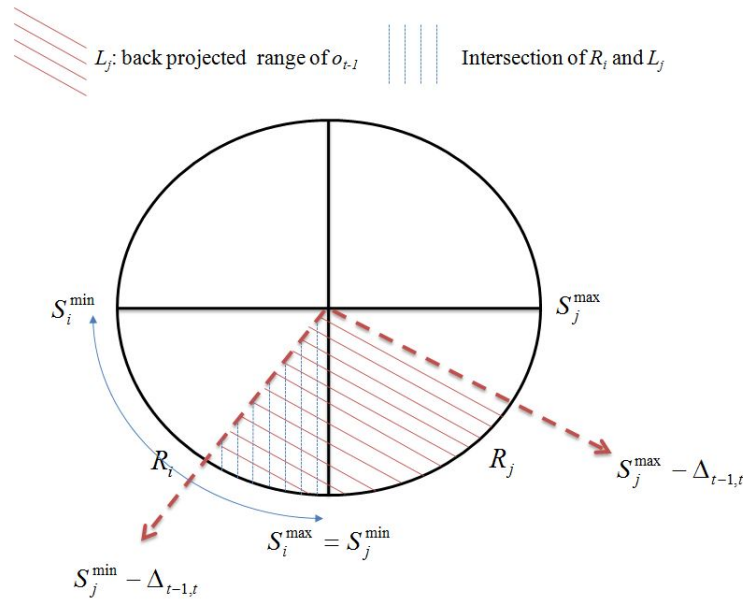


Figure 6.1: Geometrical interpretation of transition probabilities.

6.1.3 Hidden States Approximated by Observations with Compact Descriptors

The appearance of objects may significantly differ from those made during model generation under controlled circumstances. The changes in illumination, color balance, viewing angle, geometric distortion and image noise can result in heavily distorted feature descriptors. Thus observations only resemble the descriptors of the model states. Let $O = \{o_1, o_2, \dots, o_T\}$ denote the set of observation sequence. The emission probability of a particular o_t observation for state S_i is defined as

$$b_i(o_t) = P(o_t | q_t = S_i) \quad (6.7)$$

In Chapter 2, we have shown that the CEDD is a robust low dimensional descriptor for 3D object retrieval and recognition. Now Equation 6.7 can be rewritten as:

$$b_i(o_t) = \frac{TC(C(S_i), C(o_t))}{\sum_{j=1}^N TC(C(S_j), C(o_t))} \quad (6.8)$$

where C stands for the descriptor generating function of CEDD. Since each model state can cover a large directional range we will use the average CEDD vector, of available model samples within, to represent the whole state with a single descriptor.

Now we have the complete set of parameters of all HMMs denoted by $\lambda_k = (\mathbf{A}, b, \pi)$, $k \in M$. The task is to find the most probable state sequence \hat{S}_k , for all possible candidate objects, based on observations.

6.1.4 Decoding for Retrieval

We use the well-known Viterbi algorithm, explained in Chapter 2, to get the state sequence with the maximum likelihood. To achieve object retrieval we have to find the most probable state sequence \hat{S}_k with the above steps for all possible candidate objects. Now, to select the winner object, we have to compare the observations with the most probable state sequence:

$$\hat{k} = \arg \max_{\forall k \in M} \left(\frac{\sum_{i=1}^N TC(C(o_i), C(\hat{S}_{k,i}))}{N} \right). \quad (6.9)$$

6.2 Experimental Evaluation

6.2.1 Retrieval Performance on COIL-100, ALOI and SMO with Various Backgrounds

We evaluate retrieval with clear and heavily distorted queries using Gaussian noise and motion blur. For different tests different numbers (2, 4, 6, 8) of hidden states were generated by equally dividing the full circle. Each state was represented with its average CEDD descriptor vector. The average hit-rate of retrieval is measured by taking the average of 10 experiments with all database objects with randomly generated queries (the orientation angle of subsequent queries were increased monotonically).

As shown in Figures 6.2, 6.3, 6.4 and 6.5 for different quality queries, as the number of queries increases the average hit-rate increases monotonically. It is also true that higher number of states gives better results. We tested no more states than 8, where it reached the maximum performance in most cases.

We show comparisons with the VCI algorithm. There is an obvious 2-6% gain

over VCI observable. Please note, that the same visual descriptors and orientation sensor was used by VCI in previous tests.

6.2.2 Running Time and Memory Requirements

Tests were run on the tablet (specified earlier). No code optimization or parallelism was carried out and only the CPU was used during calculations. As given in Table 6.1 even for 8 queries the whole processing chain is within 1 second on the specified mobile computing hardware. This is a fraction of the complexity of VCI.

Table 6.1: Running times in seconds for the retrieval of one object from 100.

Phase	Number of Query Views (N_f^q)			
	2	4	6	8
CEDD generation	0.08	0.16	0.24	0.32
HMM evaluations	0.11	0.15	0.18	0.23
SUM	0.19	0.31	0.42	0.55

The advantage of using compact descriptors is the very limited memory requirement of object models. A CEDD descriptor occupies 144 Bytes in memory and orientation can be stored in 4 Bytes. For 100 objects and 8 states we need to store roughly 120 KB ($100 \times 8 \times 148$ Bytes).

6.3 Conclusion and Future Works

The main purpose and contribution of this chapter is twofold:

- building the 3D object retrieval framework with Markovian inference and multimodal information fusion in a viewer-centric model, and
- showing that its implementation is robust and resource efficient to be used in mobile devices.

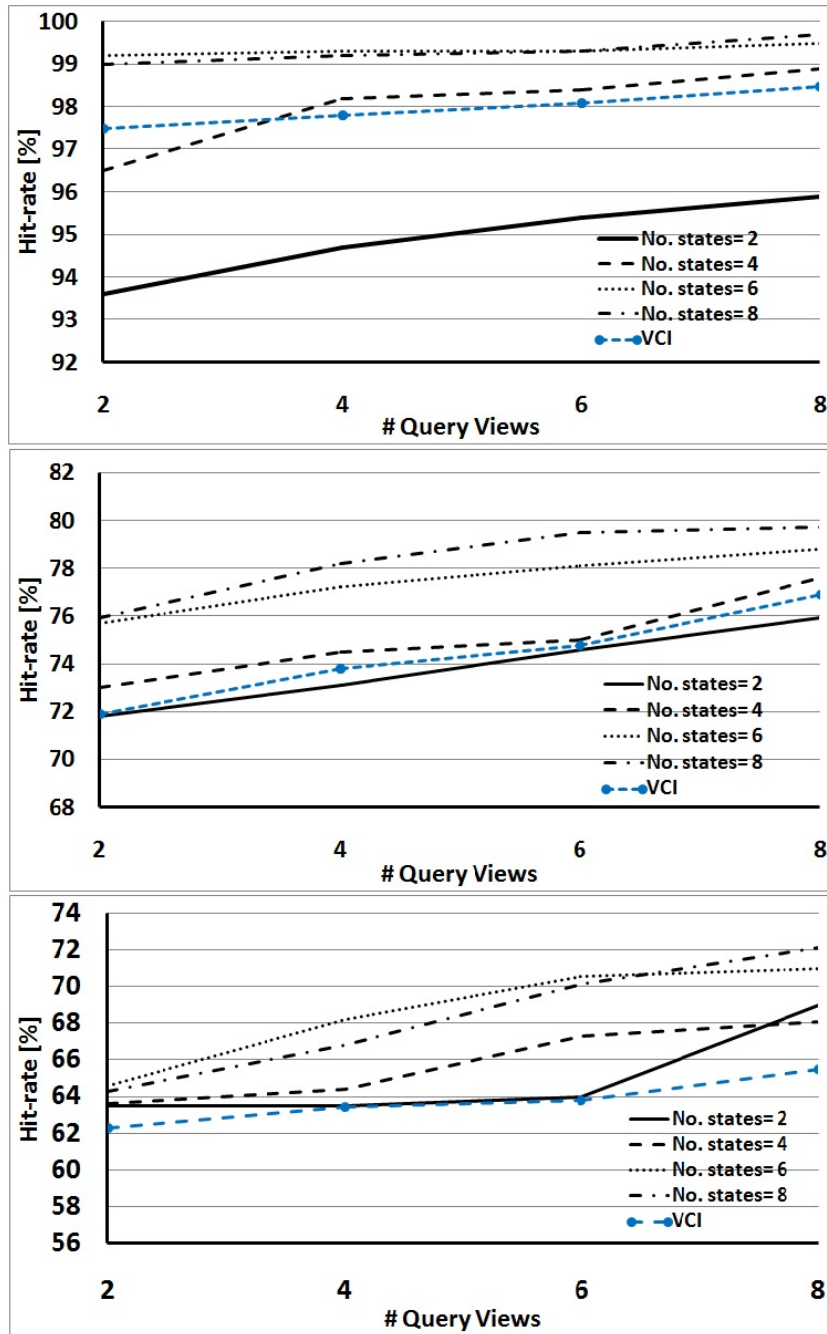


Figure 6.2: Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the COIL-100 database.

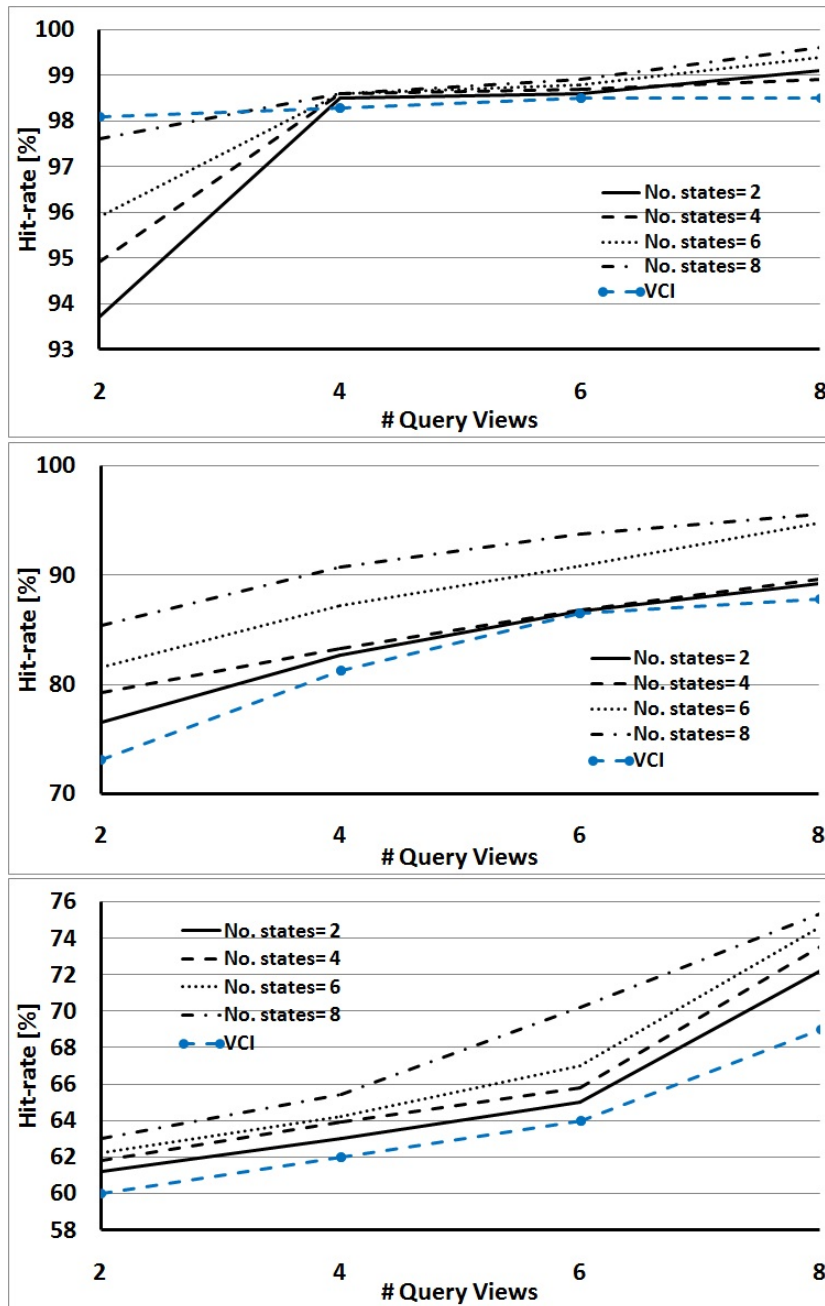


Figure 6.3: Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the ALOI database.

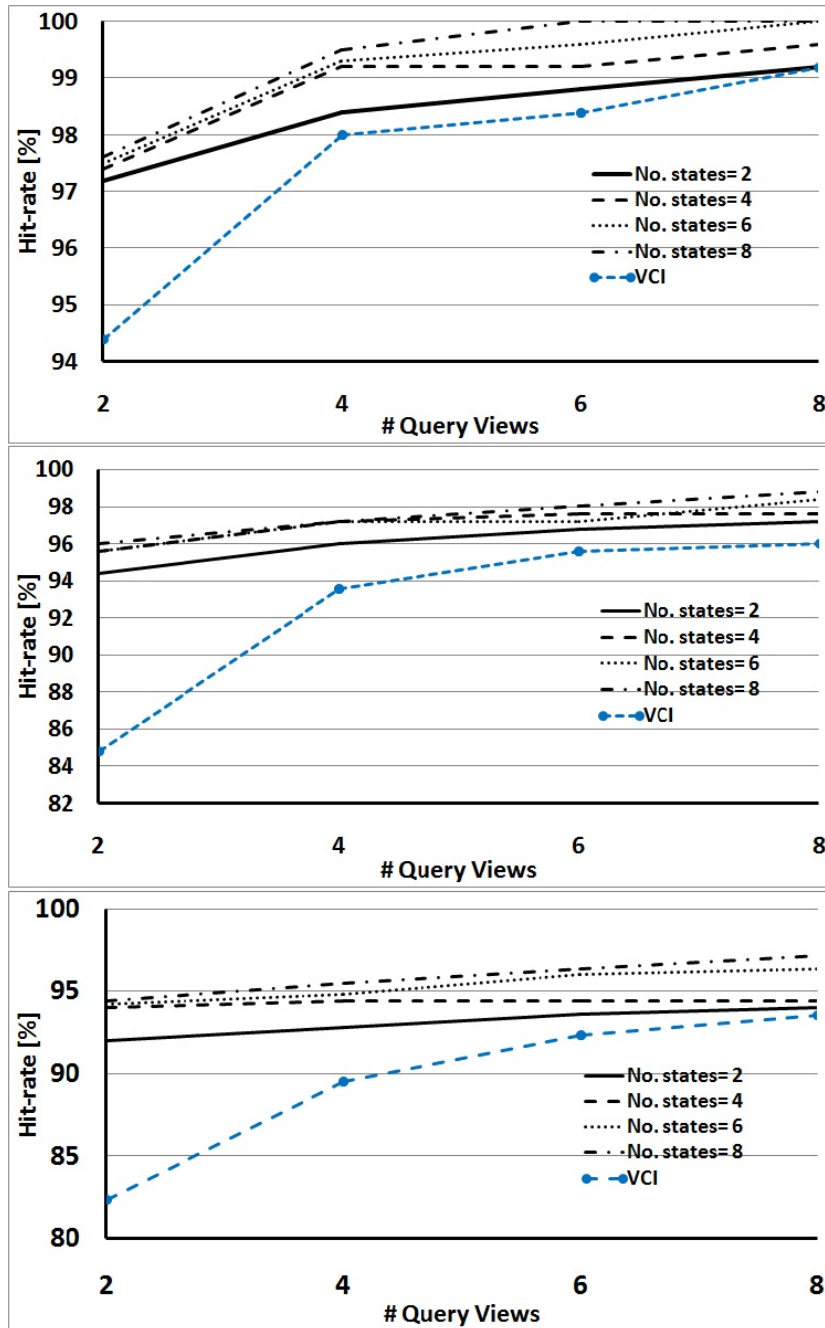


Figure 6.4: Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the SMO database with uniform backgrounds.

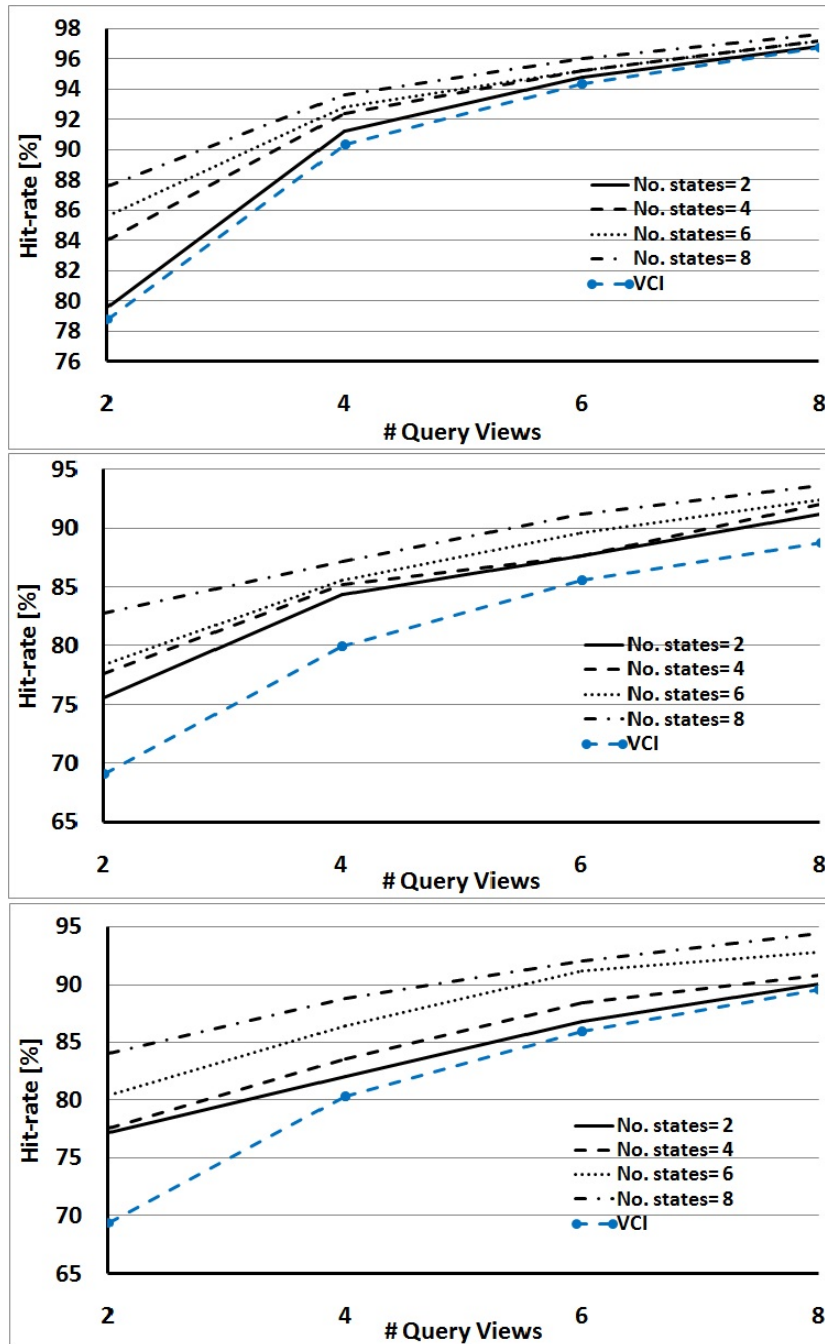


Figure 6.5: Average hit-rate for query images without distortion (top), with strong motion blur (middle), and heavy additive Gaussian noise (bottom) for the SMO database with textured backgrounds.

We presented our results over different databases of 3D objects with clear and noisy queries. While results are better than with our previous model (VCI), still there is a lot to do: we are to develop a clustering technique to build optimal states instead of the uniformly distributed states and we should work also to improve automatic object segmentation and tracking.

Chapter 7

Conclusion

In this thesis we dealt with the recognition of 3D objects with lightweight devices. The main idea is to use weak classifiers and also to utilize orientation sensors as additional information. Knowing the change in viewing direction can help us to model temporal/directional inference of object views in our viewer-centered model resulting in higher retrieval performance. Our proposed techniques were tested with thousands of test images using either clear or distorted queries. The complexities were calculated and the running times were measured on a mobile computing device. All these prove the our proposed methods can be used in real-life applications in mobile computing environments.

A great advantage of the Hough framework is that it can be used to create a technique where several optical descriptors and orientation information can be combined efficiently. Since the evaluation of the independent queries is relatively fast and it is a continuous process, active perception can be easily carried out. Moreover, being data-driven is a great advantage that DNNs do not have: adding new objects to the system does not require retraining, and the small size KD-Tree database can easily be extended or replaced. There are only a few parameters of the proposed method (the value of ω and the length of the independent retrieval lists), and we found the

optimal settings easily via manual tuning.

We have also shown that the proposed methods of 3D object retrieval can be extended for object recognition and can be easily combined with tracking.

While the Hough framework can be used even without orientation data, HMM based models can not be used without orientation sensors since the motions of the viewer can't be learnt but measured. In contrast, HMMs give better hit-rate probably due to the fact that the Hough-based approach uses limited retrieval lists. I. e. it has a relatively large probability that some valuable views are missed from further evaluations. In case of HMMs, although a limited number of states are used with average views, all possible states are evaluated.

Energy efficiency and the cost of intelligent sensors has been an important aspect for decades. To maximize the information collected by optical sensors at minimal cost, active vision approaches are necessary, as IMUs can add valuable information. The benefits of such solutions become very clear when the system has minimal computational power and memory and when only a limited amount of operation time is available (such as in the case of UAVs).

Below, besides the scientific results, we also give the list of journal and conference publications related to this work.

7.1 New Scientific Results

In this thesis, we have studied different viewer-centered solutions of the recognition problem. The following theses summarize the new scientific results in three main points. We also give the corresponding chapters and publications.

Thesis 1: We introduced a new lightweight video-based object retrieval method (VCI), where the camera is to be moved around the target object of inter-

est and the retrieval is based on compact image descriptors and fast retrieval mechanisms of weak classifiers. The independent sequence of descriptors are evaluated by the Hough framework where the reliability of retrieval can be enhanced by adding and processing new request queries from the video sequence. To increase speed KD-Tree indexing is applied. It was shown that the proposed mechanism is robust against strong blur and additive noise. Its robustness and fast operation make it a candidate to be applied in lightweight devices such as mobile phones or embedded systems. This thesis is explained in details in Chapter 4 and the related publications are **Met1**, **Met2**, **Met3**, **Met4**, **Met6**, and **Met7**.

Thesis 2: IMU sensors are now available in many embedded and mobile systems serving accurate data and the processing of orientation information as not computationally demanding. We've experimentally showed that orientation sensors can be successfully utilized to increase the hit-rate rate of 3D object recognition in optical recognition systems. Introducing an orientation term to the fitness function of the VCI method and extending the independent retrieval lists of possible candidates based on the orientation of the image views results in increased retrieval performance. We've also shown that, besides retrieval, the same Hough paradigm can be used for 3D object recognition with using orientation-dependent hypothesis testing in a post processing step. This thesis is explained in details in Chapter 5 and the related publications are **Met4**, **Met6**, and **Met7**.

Thesis 3: The utilization of Markovian models in 3D object recognition is not straightforward since the sequence of optical information is partly determined by the users' behavior (the path of the camera) and the temporal inference, a

crucial component of Markovian models, can not be trained in general to build an effective Hidden Markov Model. However, orientation sensors can help a lot in this problem: we have introduced a technique where the transition probabilities are estimated with the help of the orientation sensor based on geometrical probabilities. Thus transition probabilities are not part of the general model but can be estimated in situ during the measurements. Our evaluation results show that this approach can achieve higher hit-rate than the VCI approach while its computation complexity is significantly lower. Chapter 6 explained this thesis in details and the related publication is **Met5**.

7.2 Publications

Publications of Metwally Rashad are listed below.

7.2.1 Publications related to this Thesis

Met1. Czúni László, **Metwally Rashad**, Péter József Kiss, Mónika Gál and Ágnes Lipovits: Lightweight Mobile Object Recognition with Segmentation. In 10th National Conference on Image Processing and Pattern Recognition Society (KEPAF), pages 117–120, 2015.

Met2. Czúni László and **Metwally Rashad**: Lightweight Video Object Recognition Based on Sensor Fusion. In International Conference on Computational intelligence for multimedia understanding (IWCIM), pages 1–5, 2015.

Met3. Czúni László and **Metwally Rashad**: Interactive Object Recognition with Sensor Fusion. In 6th International Conference on Cognitive Infocommunications (CogInfoCom), pages 479–482, 2015.

Met4. Czúni László and **Metwally Rashad**: View Centered Video-based Object Recognition for Lightweight Devices. In 23rd International Conference on Systems, Signals and Image Processing (IWSSIP), pages 1–4, 2016.

Met5. Czúni László and **Metwally Rashad**: The Fusion of Optical and Orientation Information in a Markovian Framework for 3D Object Retrieval. In International Conference on Brain-Inspired Computer Vision (WBICV), pages 1–11, 2017.

Met6. Czúni László and **Metwally Rashad**: The Use of IMUs for Video Object Retrieval in Lightweight Devices. In Journal of Visual Communication and Image Representation, Vol. 48, pages 30–42, 2017 (**2016-IF: 2.164**).

Met7. Czúni László and **Metwally Rashad**: Lightweight Active Object Retrieval with Weak Classifiers. In Sensors, Vol. 18, pages 1–15, 2018 (**2016-IF: 2.67**).

7.2.2 Publication not related to this Thesis

M1. Karam Gouda and **Metwally Rashad**: Efficient String Edit Similarity Join Algorithm. In Journal of Computing and Informatics, Vol. 36, pages 1001–1022, 2017 (**2016-IF: 0.488**).

M2. Karam Gouda and **Metwally Rashad**: An Efficient Algorithm for String Similarity Joins. In 2nd International Conference on New Horizons in Basic and Applied Science (ICNHBAS), pages 100–104, 2015.

Bibliography

- [1] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi and V. Kasi, Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications, *International Journal of Signal Processing Systems*, 1 (2) (2013) 256–262.
- [2] M. Aly, M. Munich and P. Perona, Distributed KD-Trees for Retrieval from Very Large Image Collections, *Proceedings of the British Machine Vision Conference (BMVC)*, (2011) 1–11.
- [3] C. Anan and R. Hartley, Optimised KD-Trees for Fast Image Descriptor Matching, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2008) 1–8.
- [4] T. F. Ansary, M. Daoudi and J. P. Vandeborre, a Bayesian 3D Search Engine using Adaptive Views Clustering, *IEEE Transactions on Multimedia*, 9 (2007) 78–88.
- [5] J. Basak and S. K. Pal, Theoretical Quantification of Shape Distortion in Fuzzy Hough, *Fuzzy Sets and Systems*, 154 (2) (2005) 227–250.
- [6] R. Basri, Viewer-centered Representations in Object Recognition: a Computational Approach, *Handbook of Pattern Recognition and Computer Vision*, (1993) 863–882.
- [7] L. E. Baum and T. Petrie, Statistical Inference for Probabilistic Functions of Finite State Markov Chains, *The Annals of Mathematical Statistics*, 37 (6)

- (1966) 1554–1563.
- [8] L. E. Baum, T. Petrie, G. Soules and N. Weiss, a Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains, *The Annals of Mathematical Statistics*, 41 (1) (1970) 164–171.
- [9] H. Borotschnig, L. Paletta, M. Prantl and A. Pinz, Active Object Recognition in Parametric Eigenspace, *Proceedings of the British Machine Vision Conference (BMVC)*, (1998) 1–10.
- [10] G. J. Brostow, J. Shotton, J. Fauqueur and R. Cipolla, Segmentation and Recognition using Structure from Motion Point Clouds, *European Conference on Computer Vision*, (2008) 44–57.
- [11] B. Browatzki, V. Tikhanoff, G. Metta, H. H. Bulthoff and C. Wallraven, Active Object Recognition on a Humanoid Robot, *IEEE International Conference on Robotics and Automation (ICRA)*, (2012) 2021–2028.
- [12] A. Bruno, L. Greco and M. Cascia, Video Object Recognition and Modeling by SIFT Matching Optimization, *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, (2014) 662–670.
- [13] H. H. Bülthoff and S. Edelman, Psychophysical Support for a Two-dimensional View Interpolation Theory of Object Recognition, *Proceedings of the National Academy of Sciences*, 89 (1992) 60–64.
- [14] A. Chan, C. Achard and L. Lucat, Deeply Optimized Hough Transform: Application to Action Segmentation, *International Conference on Image Analysis and Processing*, (2013) 51–60.
- [15] S. K. Chang and E. Jungert, *Symbolic Projection for Image Information Retrieval and Spatial Reasoning*, London: Academic Press, (1996) 62–70.

- [16] S. A. Chatzichristofis and Y. S. Boutalis, Accurate Image Retrieval based on Compact Composite Descriptors and Relevance Feedback Information, *International Journal of Pattern Recognition and Artificial Intelligence*, (2010) 207–244.
- [17] S. A. Chatzichristofis, Y. S. Boutalis and M. Lux, Selection of the Proper Compact Composite Descriptor for Improving Content based Image Retrieval, *6th International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA)*, (2009) 134–140.
- [18] L. Czúni and M. Rashad, View Centered Video-based Object Recognition for Lightweight Devices, *International Conference on Systems, Signals and Image Processing (IWSSIP)*, (2016) 1–4.
- [19] L. Czúni and M. Rashad, Lightweight Video Object Recognition based on Sensor Fusion, *International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM)*, (2015) 1–5.
- [20] L. Czúni, P. J. Kiss, A. Lipovits and M. Gál, Mobile Object Recognition: an Analysis of Image Quality Factors, Technical Report, UofP, https://keplab.mik.uni-pannon.hu/images/anyagok/TC_OR_2014.pdf, (2014)
- [21] L. Czúni, P. J. Kiss, A. Lipovits and M. Gál, Lightweight Mobile Object Recognition, *IEEE International Conference on Image Processing (ICIP)*, (2014) 3426–3428.
- [22] P. Daras and A. Axenopoulos, a 3D Shape Retrieval Framework Supporting Multimodal Queries, *International Journal of Computer Vision*, 89 (2010) 229–247.
- [23] C. de Trazegnies, C. Urdiales, A. Bandera and F. Sandoval, 3D Object Recognition based on Curvature Information of Planar Views, *Pattern Recognition*,

- 36 (2003) 2571–2584.
- [24] P. Dinkova, P. Georgieva and M. Milanova, Face Recognition using Singular Value Decomposition and Hidden Markov Model, 16th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems (MAMECTIS), (2014) 144–149.
- [25] J. Engel, T. Sch and D. Cremers, LSD-SLAM: Large-scale Direct Monocular SLAM, European Conference on Computer Vision, (2014) 834–849.
- [26] D. Exner, E. Bruns, D. Kurz, A. Grundhofer and O. Bimber, Fast and Robust CAMShift Tracking, Computer Vision and Pattern Recognition Workshops (CVPRW), (2010) 9–16.
- [27] F. Fang and S. He, Viewer-centered Object Representation in the Human Visual System Revealed by Viewpoint Aftereffects, *Neuron*, 45 (2005) 793–800.
- [28] G. D. Forney, The Viterbi Algorithm, *Proceedings of the IEEE*, 61 (3) (1973) 268–278.
- [29] J. H. Friedman, J. L. Bentley and R. A. Finkel, an Algorithm for Finding Best Matches in Logarithmic Expected Time, *ACM Transactions on Mathematical Software*, 3 (1977) 209–226.
- [30] K. Fukushima and S. Miyake, Neocognitron: a Self-organizing Neural Network Model for a Mechanism of Pattern Recognition, Competition and cooperation in neural nets, (1982) 267–285.
- [31] T. Furuya and R. Ohbuchi, Dense Sampling and Fast Encoding for 3D Model Retrieval using Bag-of-Visual Features, *Proceedings of the ACM International Conference on Image and Video Retrieval*, (2009) 26.

- [32] J. Gall and V. Lempitsky, Class-specific Hough Forests for Object Detection, International Conference on Computer Vision and Pattern Recognition, (2013) 143–157.
- [33] J. Gall, A. Yao, N. Razavi, L. Van Gool and V. Lempitsky, Hough Forests for Object Detection, Tracking, and Action Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2011) 2188–2202.
- [34] S. Gammeter, A. Gassmann, L. Bossard, T. Quack and L. Gool, Server-side Object Recognition and Client-side Object Tracking for Mobile Augmented Reality, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2010) 1–8.
- [35] Y. Gao, J. Tang, R. Hong, S. Yan, Q. Dai, N. Zhang and T. S. Chua, Camera Constraint-free View-based 3D Object Retrieval, IEEE Transactions on Image Processing, 21 (4) (2012) 2269–2281.
- [36] Y. Gao, M. Wang, D. Tao, R. Ji and Q. Dai, 3D Object Retrieval and Recognition with Hypergraph Analysis, IEEE Transactions on Image Processing, 21 (9) (2012) 4290–4303.
- [37] J. M. Geusebroek, G. J. Burghouts and A. W. Smeulders, The Amsterdam Library of Object Images, International Journal of Computer Vision, 61 (2005) 103–112.
- [38] M. Godec, C. Leistner, H. Bischof, A. Starzacher and B. Rinner, Audio-visual Co-training for Vehicle Classification, 7th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 1 (2010) 586 – 592.
- [39] N. Govender, J. Claassens, F. Nicolls and J. Warrell, Active Object Recognition Using Vocabulary Trees, IEEE Workshop on Robot Vision (WORV), (2013) 20–26.

-
- [40] K. Grauman, B. Leibe, Visual Object Recognition, *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(2), (2011) 1–181.
- [41] Y. K. Ham and R. H. Park, 3D Object Recognition in Range Images using Hidden Markov Models and Neural Networks, *Pattern Recognition*, 32 (5) (1999) 729–742.
- [42] J. Han, L. Shao, D. Xu and J. Shotton, Enhanced Computer Vision with Microsoft Kinect Sensor: a Review, *IEEE Transactions on Cybernetics*, 43 (5) (2013) 1318–1334.
- [43] M. Hejrati and D. Ramanan, Analysis by Synthesis: 3D Object Recognition by Object Reconstruction, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2014) 2449–2456.
- [44] J. D. Hol, T. B. Schon and F. Gustafsson, a New Algorithm for Calibrating a Combined Camera and IMU Sensor Unit, *10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, (2008) 1857–1862.
- [45] J. Hornegger, H. Niemann, D. Paulus and G. Schlottke, Object Recognition using Hidden Markov Models, *Machine Intelligence and Pattern Recognition*, 16 (1994) 37–44.
- [46] M. Irani and P. Anandan, About Direct Methods, *International Workshop on Vision Algorithms*, (1999) 267–277.
- [47] Y. K. Jain and R. K. Singh, Efficient View based 3D Object Retrieval using Hidden Markov Model, *3D Research*, 4 (4) (2013) 5.
- [48] O. Javed, M. Shah and D. Comaniciu, a Probabilistic Framework for Object Recognition in Video, *International Conference on Image Processing (ICIP)*, (2004) 2713–2716.

- [49] H. Jegou, M. Douze and C. Schmid, Product Quantization for Nearest Neighbor Search, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33 (1) (2011) 117–128.
- [50] T. Kanungo, Hidden Markov Model Tutorial, <http://www.kanungo.com/software/hmmtut.pdf>, (1999).
- [51] J. J. Koenderink and A. J. van Doorn, The Internal Representation of Solid Shape with Respect to Vision, *Biological Cybernetics*, 32 (1979) 211–216.
- [52] A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems*, (2012) 1097–1105.
- [53] Y. Le Cun, Learning Processes in an Asymmetric Threshold Network, *Disordered Systems and Biological Organization*, (1986) 233–240.
- [54] K. Lim, Y. Hong, Y. Choi and H. Byun, Real-time Traffic Sign Recognition based on a General Purpose GPU and Deep-learning, *PLoS one*, 12(3) (2017), e0173317.
- [55] A. Lucieer, S. M. D. Jong and D. Turner, Mapping Landslide Displacements using Structure from Motion (SfM) and Image Correlation of Multi-temporal UAV Photography, *Progress in Physical Geography*, 38 (1) (2014) 97–116.
- [56] S. Maji and J. Malik, Object Detection using a Max-margin Hough Transform, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2009) 1038–1045.
- [57] O. Miksik and K. Mikolajczyk, Evaluation of Local Detectors and Descriptors for Fast Feature Matching, *International Conference on Pattern Recognition (ICPR)*, (2012) 2681–2684.

- [58] H. Murase and S. K. Nayar, Visual Learning and Recognition of 3D Objects from Appearance, *International Journal of Computer Vision*, 14 (1995) 5–24.
- [59] S. A. Nene, S. K. Nayar and H. Murase, Columbia Object Image Library (COIL-100), Technical Report CUCS, Department of Computer Science, Columbia University, (1996).
- [60] K. Nishida, T. Kurita, Y. Ogiuchi and M. Higashikubo, Visual Tracking Algorithm using Pixel-pair Feature, *International Conference on Pattern Recognition (ICPR)*, (2010) 1808–1811.
- [61] H. Noor, S. H. Mirza, Y. Sheikh, A. Jain and M. Shah, Model Generation for Video-based Object Recognition, *14th annual ACM International Conference on Multimedia*, (2006) 715–718.
- [62] C. Potthast, A. Breitenmoser, F. Sha and G. S. Sukhatme, Active Multi-view Object Recognition: a Unifying View on Online Feature Selection and View Planning, *Robotics and Autonomous Systems*, 84 (2016) 31–47.
- [63] J. Puzicha, J. M. Buhmann, Y. Rubner and C. Tomasi, Empirical Evaluation of Dissimilarity Measures for Color and Texture, *International Conference on Computer Vision*, (1999) 1165–1173.
- [64] T. Qin, Z. Chen, H. Zhang, S. Li, W. Xiang and M. Li, a Learning Algorithm of CMAC based on RLS, *Neural Processing Letters*, 19 (1) (2004) 49–61.
- [65] L. R. Rabiner, a Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, 77 (2) (1989) 257–286.
- [66] M. Rizon, H. Yazid, P. Saad, A. Y. M. Shakaff, A. R. Saad, M. Sugisaka, M. Sugisaka, S. Yaacob, M. R. Mamat and M. Karthigayan, Object Detection using Circular Hough Transform, *American Journal of Applied Sciences*, 2 (2005) 1606–1609.

- [67] C. Rother, V. Kolmogorov and A. Blake, Grabcut: Interactive Foreground Extraction using Iterated Graph Cuts, *ACM Transactions on Graphics (TOG)*, 23 (2004) 309-314.
- [68] S. D. Roy, S. Chaudhury and S. Banerjee, Active Recognition Through Next View Planning: a Survey, *Pattern Recognition*, 37 (2004) 429–446.
- [69] E. Salahat and M. Qasaimeh, Recent Advances in Features Extraction and Description Algorithms: a Comprehensive Survey, *IEEE International Conference on Industrial Technology (ICIT)*, (2017) 1059–1063.
- [70] T. Schops, J. Engel and D. Cremers, Semi-dense Visual Odometry for AR on a Smartphone, *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, (2014) 145–150.
- [71] C. Silpa-Anan and R. Hartley, Optimised KD-Trees for Fast Image Descriptor Matching, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2008) 1–8.
- [72] M. Smereka and I. Duleba, Circular Object Detection Using a Modified Hough Transform, *International Journal of Applied Mathematics and Computer Science*, 18 (2008) 85–91.
- [73] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, Going Deeper with Convolutions, *IEEE Conference on Computer Vision and Pattern Recognition*, (2015) 1–9.
- [74] A. Torralba, K. P. Murphy, W. T. Freeman and M. A. Rubin, Context-based Vision System for Place and Object Recognition, *IEEE International Conference on Computer Vision*, 1 (2003) 273–280.
- [75] P. H. Torr and A. Zisserman, Feature based Methods for Structure and Motion Estimation, *International Workshop on Vision Algorithms*, (1999) 278–294.

-
- [76] A. Tungkasthan, S. Intarasema and W. Premchaiswadi, Spatial Color Indexing using ACC Algorithm, 7th International Conference on ICT and Knowledge Engineering, (2009) 113–117.
- [77] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey and J. M. Reynolds, Structure-from-Motion Photogrammetry: a Low-cost, Effective Tool for Geoscience Applications, *Geomorphology*, 179 (2012) 300–314.
- [78] P. Wieschollek, O. Wang, A. Sorkine-Hornung and H. Lensch, Efficient Large-scale Approximate Nearest Neighbor Search on the GPU, *IEEE Conference on Computer Vision and Pattern Recognition*, (2016) 2027–2035.
- [79] J. Xiao, A. Owens and A. Torralba, Sun3d: a Database of Big Spaces Reconstructed using SFM and Object Labels, *IEEE International Conference on Computer Vision (ICCV)*, (2013) 1625–1632.
- [80] A. Yilmaz, O. Javed and M. Shah, Object Tracking: a Survey, *ACM Computing Surveys (CSUR)*, 38 (4), (2006) 13.
- [81] R. Yu and X. Zhang, The Design and Implementation of Face Tracking in Real Time Multimedia Recording System, 2nd International Congress on Image and Signal Processing, (2009) 1–3.