Technische Universität München
Lehrstuhl für Datenverarbeitung

Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Robotik und Mechatronik

Visual Servoing of Textured Free-Form Objects in 6 Degrees of Freedom

Wolfgang Sepp

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. G. Färber
Prüfer der Dissertation:
1. Univ.-Prof. Dr.-Ing. K. Diepold
2. Hon.-Prof. Dr.-Ing. G. Hirzinger

Die Dissertation wurde am 18.02.2008 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 22.07.2008 angenommen.

Acknowledgement

This thesis has been written during my employment at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR e.V.) in Oberpfaffenhofen, Germany. I have been supported in my work by many persons who I would like to mention in the following.

First of all, I would like to thank Prof. Dr.-Ing. Gerd Hirzinger for giving me the opportunity to work in this institute full of creativity and potential. I'm particularly grateful to my thesis advisor Prof. Dr.-Ing. Klaus Diepold for his support, encouragement, and especially for reminding me to answer the right questions.

I also owe to friends and colleagues. By all means the following order does not correspond to any decreasing order of gratitude. I would like to mention first Stefan Fuchs who actively supported me in the direct context of this work. I thank Ulrich Hillenbrand for all the discussion I had with him and for his valuable suggestions on the right notations. For their proofreading, I express my gratitude to Matthias Wimmer, Matthias and Helene Renz, Anita van Barneveld, Richard Stevenson, Rainer Konietschke, Martin Gröger, and Michael Suppa. In particular, I thank Klaus Strobl for his willing support and critical opinions. My appreciation goes also to Paolo Robuffo Giordano for his views on robot control and visual servoing. Furthermore, I thank Franziska Zacharias for her simulations on the dexterity of the LWR-2. Thank you all for your friendship.

Last but no least, I would like to thank my wonderful girlfriend Min who gave me support and encouragement. I am very grateful to my parents, who allowed me to freely choose my own way. And finally, I would like to dedicate this work to my grandparents Skaidritte and Erich. I will not forget their humour, temperament, care, and their hearty welcomes.

Munich, February 2008
Wolfgang Sepp

Abstract

This thesis aims at the ability of a robot to quickly react to changes in its environment qualifying the robot for seamless interactions with humans. In particular, the thesis addresses the problem of grasping textured moving objects with a robotic manipulator considering alternative, non feature-based methods of image processing.

In order to accomplish the task appropriately, the work copes with the different issues on the physical alignment of the image sensor and robot actuator, the difficulties of real-time visual pose estimation, and the aspects of robot control. Accordingly, the relevant literature is firstly examined in the fields of computer vision and robot control based on visual information.

In contrast to the feature-based approaches, this thesis presents novel methods for appearance-based tracking in 6 degrees of freedom (DoF) relying on a *textured point cloud* as rigid three-dimensional representation. According object models are matched against the actual images under full-perspective projection. Hence, the approach allows for the pose estimation of any potential free-form surface for a large range of object distances.

Two strategies are considered for the exploration of the space of object-to-camera poses: single hypothesis optimisation by means of Gauss-Newton minimisation and stochastic multi-hypotheses propagation with an annealed Markov-Chain Monte-Carlo filter. The real-time tracking performance of the former approach is improved in this thesis by a novel, analytic formula of the image Jacobian. Therewith, intensity changes respective to object motion are efficiently predicted without any actual image measurements. The presented methods are systematically evaluated with respect to their convergence properties.

In addition, the potentially adverse effects of changes in illumination are addressed. Thereby, established computer vision techniques are examined and adapted to the particular models of surface shape and appearance matching. Experiments attest the method of template updating the largest improvements in the probability of convergence.

In consideration of limited computational resources, the task of global pose estimation in 6 DoF is decomposed into a hierarchy of goals that are accomplished in succession. In a cascade of pure appearance-based approaches, the object is first localised in 2 DoF without any prior information of the object position. The obtained hypothesis is successively refined to a pose estimate in 6 DoF using the mentioned universal object model and exploration strategies.

In conclusion, the tracking cascade is integrated in robot control on the basis of an appropriate, i.e., position-based, interface. The physical configuration of both the sensing components and the actuator is discussed aiming at the achievement of a convenient and natural interaction area between the robot and the human. Finally, the system is successfully validated for the desired human-robot interaction by several experiments on catching a moving object from the users hand.

Contents

List of Symbols

General

O	:	$\mathcal{A} \to \mathbb{R}$	objective function
p	:	$\mathcal{B} \to \mathbb{R}$	probability density function (p.d.f.)
L	:	$\mathbb{R}^6 \to \mathbb{R}$	pose likelihood function
${}^tL$	:	$\mathbb{R}^6 \to \mathbb{R}$	pose likelihood function for image at time t
E	:	$Y \to y$	expectation y for a random variable Y
D	:	$Y \to y$	standard deviation y for a random variable Y
$\mathcal{N}(m, \Sigma)$			normal distribution with mean m and covariance Σ

Pose and Motion

$\boldsymbol{\mu}$	$\in$	$\mathbb{R}^6$	6-DoF pose of object in camera frame
${}^t\boldsymbol{\mu}$	$\in$	$\mathbb{R}^6$	6-DoF pose of object in camera frame at time t
${}^0\boldsymbol{\mu}$	$\in$	$\mathbb{R}^6$	6-DoF reference pose of object in camera frame
$\delta\boldsymbol{\mu}$	$\in$	$\mathbb{R}^6$	6-DoF pose variation of object in camera frame
$\hat{\boldsymbol{\mu}}$	$\in$	$\mathbb{R}^6$	6-DoF pose estimate of object in camera frame
$\boldsymbol{\mu}^\star$	$\in$	$\mathbb{R}^6$	ground-truth 6-DoF pose of object in camera frame

$m : \mathbb{R}^3 \times \mathbb{R}^6 \to \mathbb{R}^3$ 6-DoF rigid-body motion

Image and Texture

I	:	$\mathbb{R}^2 \to \mathbb{R}$	image intensity function
${}^tI$	:	$\mathbb{R}^2 \to \mathbb{R}$	image intensity function for image at time t
${}^0I$	:	$\mathbb{R}^2 \to \mathbb{R}$	image intensity function for reference image
$I_{\mathbf{x}}$	:	$\mathbb{R}^6 \to \mathbb{R}$	image intensity mapping of three-dimensional surface point $\mathbf{x}$ under 6-DoF rigid body motion
I	$\in$	$\mathbb{R}^N$	texture vector
${}^t\mathrm{I}$	$\in$	$\mathbb{R}^N$	texture vector at time t
${}^0\mathrm{I}$	$\in$	$\mathbb{R}^N$	reference texture vector
${}^t\mathrm{T}$	$\in$	$\mathbb{R}^N$	texture template at time t
${}^0\mathrm{T}$	$\in$	$\mathbb{R}^N$	reference texture template
p	:	$\mathbb{R}^3 \to \mathbb{R}^2$	full perspective projection with known intrinsic camera parameters

Surface

s	:	$\mathbb{R}^2 \rightarrow \mathbb{R}^3$	parametric surface patch
$\mathcal{X}$	$\subset$	$\mathbb{R}^3$	continous set of surface points
X	$\in$	$\mathcal{X}$	set $\{\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_N\}$ of surface points
M	$\in$	$\mathbb{R}^{N\times 3}$	matrix $(\mathbf{n}_{\mathbf{x}_1}, \mathbf{n}_{\mathbf{x}_2}, .., \mathbf{n}_{\mathbf{x}_N})^{\mathrm{T}}$ of surface normals
$\mathbf{d}_\mathbf{x}$	$\in$	$\mathbb{R}^3$	unit vector of texture-gradient direction tangential to the surface point at $\mathbf{x}$
$\mathbf{o}_\mathbf{x}$	$\in$	$\mathbb{R}^3$	normal of the plane spanned by the surface normal $\mathbf{n}_\mathbf{x}$ and gradient direction $\mathbf{d}_\mathbf{x}$ at $\mathbf{x}$
$\mathbf{r}_\mathbf{x}$	:	$\mathbb{R}^6 \rightarrow \mathbb{R}^3$	line of sight vector from the surface point $\mathbf{x}$ to the optical centre for a specific 6-DoF pose

Markov-Chain Monte-Carlo Estimation

$\mathbf{z}_k$	multi-dimensional observation at time k
$\mathbf{z}_{j:k}$	history of multi-dimensional observations from time j to k
$\mathbf{x}_k$	unobservable multi-dimensional state of a stochastic process at time k
$\mathbf{x}_{j:k}$	history of unobservable multi-dimensional state of a stochastic process from time j to k
$\mathbf{x}_k^{(i)}$	i-th sample of multi-dimensional state at time k
$w_k^{(i)}$	weight corresponding to the i-th sample of multi-dimensional state at time k

Irradiance, Radiance, and Reflectance

${}^{\mathrm{S}}L_\mathbf{x}$	$\in$	$\mathbb{R}$	radiance measured for surface point $\mathbf{x}$
${}^{\mathrm{A}}L_\mathbf{x}$	:	$\mathbb{R}^2 \rightarrow \mathbb{R}$	ambient radiance function at surface point $\mathbf{x}$ over possible radiance directions
${}^{\mathrm{A}}L$	:	$\mathbb{R}^2 \rightarrow \mathbb{R}$	ambient radiance function over possible illumination directions
${}^{\mathrm{S}}E_\mathbf{x}$	:	$\mathbb{R}^2 \rightarrow \mathbb{R}$	surface irradiance function for surface point $\mathbf{x}$ over possible radiance directions
$f_\mathbf{x} : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$			bidirectional reflectance-distribution function
$\rho_\mathbf{x}$	$\in$	$\mathbb{R}$	albedo of surface point $\mathbf{x}$
${}^{\mathrm{S}}\mathrm{L}$	$\in$	$\mathbb{R}^N$	vector of surface radiances
$\mathcal{L}$			illumination subspace

B	$\in$	$\mathbb{R}^{N\times 3}$	base of illumination subspace
$\mathrm{U_B}\cdot\Sigma_\mathrm{B}\cdot\mathrm{V_B^T}$			singular value decomposition of base B of illumination subspace
$\mathbf{s}$	$\in$	$\mathbb{R}^3$	radiance vector
a_t, b_t	$\in$	$\mathbb{R}$	intensity scaling and intensity offset parameter at time t

Hierarchical Tracking

${}^t\zeta$	$\in$	$\{1,2,\ldots,{}^\zeta N\}$	active tracking stage at time t
θ_i^-, θ_i^+	$\in$	$\mathbb{R}$	lower and upper confidence bound for tracking stage i
${}^t\gamma$	$\in$	$\mathbb{R}$	confidence computed at time t
$h:\mathbb{R}^3\supset\mathcal{C}\rightarrow\{1,2,\ldots,{}^\mathrm{b}N\}$			colour binning function
k	:	$\mathbb{R}^2\times\mathbb{R}^2\rightarrow\mathbb{R}$	kernel function
${}^tp_b$	:	$\mathbb{R}^2\times\mathbb{R}^2\rightarrow\mathbb{R}$	colour probability function at time t for colour bin b
$\mathbf{q}$	$\in$	$\mathbb{R}^{{}^\mathrm{b}N}$	prior colour probability distribution
${}^t\mathbf{p}$	$\in$	$\mathbb{R}^{{}^\mathrm{b}N}$	current colour probability distribution
$\rho:\mathbb{R}^{{}^\mathrm{b}N}\times\mathbb{R}^{{}^\mathrm{b}N}\rightarrow\mathbb{R}$			Bhattacharyya coefficient over two colour distributions
${}^t\hat{\mathbf{u}}_i$	$\in$	$\mathbb{R}^2$	object location estimate at time t iteration i
${}^t\hat{\boldsymbol{\sigma}}_i$	$\in$	$\mathbb{R}$	scale estimate at time t and iteration i
${}^t\hat{s}_i$			scale index estimate at time t and iteration i
${}^t\hat{\boldsymbol{\mu}}$	$\in$	$\mathbb{R}^2, \mathbb{R}^3, \mathbb{R}^6$	final stage pose estimate at time t
${}^t\hat{\boldsymbol{\mu}}_i$	$\in$	$\mathbb{R}^6$	stage pose estimate at time t and iteration i
${}^t\boldsymbol{\mu}_i^{(j)}$	$\in$	$\mathbb{R}^6$	j-th stage pose sample at time t and iteration i
${}^tw_i^{(j)}$	$\in$	$\mathbb{R}$	weight corresponding to the j-th sample of stage pose at time t and iteration i

Servoing

${}^-W_\mathrm{x}, {}^+W_\mathrm{x}$	$\in$	$\mathbb{R}$	lower and upper workspace limits in x-direction
${}^-W_\mathrm{z}, {}^+W_\mathrm{z}$	$\in$	$\mathbb{R}$	lower and upper workspace limits in z-direction
O_x	$\in$	$\mathbb{R}$	object width
ξ_min	$\in$	$\mathbb{R}^+$	minimal degree of relative perspective distortion
$\angle C$	$\in$	$[0,\pi]$	angle of camera aperture

List of Figures

List of Tables

1 Introduction

One of the greatest dreams of humankind is to enjoy the pleasures of life without drudgery. In this vision, hard, painful, dangerous, or simply unbeloved work is totally abolished or accomplished by some other means. This desire stimulated the development of machines to take the burdens of human work. Eventually, a general-purpose, intelligent machine, generally known as *robot*, would assist humans in every-day life.

Such a machine should perform tasks in a human environment autonomously. Since the environment is not necessarily modelled to suit the machine, the machine should be built to suit the environment. Hence, a robot should be capable of taking orders from humans, recognising its surrounding, and have the ability to interact with the environment in order to execute a particular task. In detail, the robot should exhibit sensing, interpreting, decision-making, and actuating skills in order to become a *service robot*.

So far, the intelligent, general-purpose robot has not been built, and achievements have been only reported in isolated domains. Mobile robots are able to autonomously explore indoor environments and to navigate within that environment independently. Furthermore, skills have been developed for dedicated manipulation tasks such as the autonomous grasping of known stationary objects with a dextrous robotic arm. However, for true interaction with its surroundings, especially with humans, the robot capabilities are not sufficient. Instead, a robot has to continously adapt its actions to the changing environment. This ability is still subject to research. However, upon completion, it will positively affect human-robot interaction and the acceptance of robots in a dramatic way.

This thesis aims at an important property of human-robot interaction, i.e., the robots ability to adaptively take objects over from a human (see figure 1.1). In order for this handshake to succeed, the robot has to fulfil several subtasks autonomously, e.g., first to recognise the object, then to determine its pose and motion, and finally to follow its motion with robotic actuators so as to grasp the object with the appropriate tool. This work proposes methods to estimate

object motion with the aid of visual sensors, and investigates appropriate object tracking hierarchies, as well as appropriate sensor-actuator configurations for such a task.

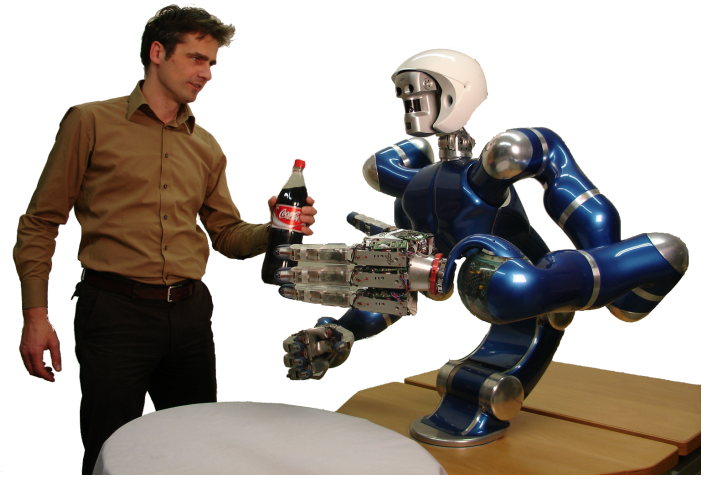

Figure 1.1: Example of human-robot interaction: Grasping of a moving object with a service robot. The figure shows the DLR humanoid two-arm system JUSTIN.

The following section 1.1 introduces the interested reader to basic problems in tracking moving objects with visual sensors. Known problems of robot actuation and control are outlined in section 1.2. The state of the art methods for physical human-robot interaction are mentioned in section 1.3, and finally an overview of the following work is given (section 1.4) highlighting the main contributions of the thesis.

1.1 Visual Tracking

Sensing the environment is a fundamental property of physically embedded intelligent systems. While simple machines expect the surrounding to be a pre-defined setting, intelligent systems are meant to sense an a priori unknown scene. Among the variety of passive and thus safe sensors, digital video cameras are advantageous due to their ability to capture high quantities of information within a single measurement cycle. With this sensor at hand, the tasks of object detection, object classification, and physical object localisation become possible and eventually allows for true interaction with the environment.

Once the category of a detected object is identified, its location or pose is still not determined accurately. In order to estimate the pose, a model representing the object in question is registered with the sensed environment. The complexity of this task depends on the constraints imposed by or on the environment. Usually, the object is assumed to be rigid, i.e., it is not articulated nor containing any deformable parts. Hence, from the computational perspective,

pose estimation is an optimisation problem in at most 6 degrees of freedom, 3 degrees of freedom for the position and 3 degrees of freedom for the rotation of the object relative to the observer.

These degrees of freedom are restricted further if constraints apply either to the manifold of object appearance or to the pose manifold. The former constraints hold for surfaces of revolution or planar surfaces whose texture does not change in tangential direction, e.g., uniformly coloured bottles. The latter constraints instead occur when the environment fixes some translational or rotational degrees of freedom, e.g., when the object is located on top of a table.

Hence, the complexity of the pose estimation task depends on the number of unconstrained degrees of freedom (DoF). While objects can be localised in 3 DoF more easily, for instance on a conveyor with two translational DoF and one rotational DoF, the estimation of a pose in 6 DoF is more difficult and an ongoing research topic. We denote the problem to be a *global pose estimation problem* when no hypotheses are given that could narrow the search space for object poses. This task proves to be demanding since the interpretation of the captured image is both difficult and ambiguous due to the manifold in the appearance of the object. Moreover, global pose estimation typically is computationally expensive.

It is often the case that the captured scene changes dynamically either through motion of the sensor or because of movement of objects. In order to react to these changes, the object has to be recurrently localised as frequently as possible. Accordingly, *tracking* approaches generate an initial pose hypothesis for the current sensor readings based on previous pose estimates, and thus perform *local pose estimation*. These approaches tackle the ambiguity of pose estimation and additionally reduce the computational costs of the task. Therefore, the estimation can be updated at significantly higher rates enabling an immediate interaction with moving parts.

The following subsections reflect different aspects of tracking within a sequence of camera images. For objects in motion, temporal constraints have to be taken into account when capturing the images (section 1.1.1). For single images, spatial constraints determine the complexity and the accuracy of pose estimation (section 1.1.2). Moreover, concurrent constraints on space and time force the balance between processes in time and processes in space (section 1.1.3).

1.1.1 Properties of Time

Digital cameras produce two-dimensional images of a part of a scene, which are processed and stored electronically. In contrast to other sensors, a camera is generally not a continuous analog sensor. The employed sensor chips based on CCD[1] or CMOS[2] technology do not directly transform light into electric signals. Instead, the radiation hitting the sensor *loads* a cell with electric charge. The charge is accumulated over a period of time known as the integration period,

[1] charge coupled device

[2] complementary metal-oxide semiconductor

and is transformed thereafter into an electric signal. The integration periods are not required to be cohesive and therefore the environmental information can be sampled and processed discontinuously.

In the following, the major dependencies between the sampling rate and the properties of the sensing equipment, the environment, and the object motion are outlined.

Sampling Radiation

Generally, sampling of continuous signals should follow the Nyquist-Shannon sampling theorem [106] to avoid aliasing effects. Accordingly, in the case of radiation reaching a single sensor cell, the temporal evolution of the signal cannot be uniquely deduced from the taken samples if the highest frequency component of the radiation signal exceeds half the sampling frequency. The scene radiance itself depends on the characteristics of the light sources, on the texture and reflection properties of the scene, and on the pose and relative motion between the sensor and the environment.

Usually, variations of radiation induced by light sources are not explicitely considered in the definition of a computer vision problem. Nevertheless, these variations can negatively affect the evaluation of the images. Typically, light changes slowly in outdoor scenes and very frequently in indoor scenes. While former changes are caused by the weather conditions and the alternation of day and night, the latter changes are related to the power frequency of the electric circuit (e.g. 50 Hz in Europe and 60 Hz in Northern America). These effects can be compensated by triggering the exposure of the sensor with a multiple of the alternation period.

Sampling Motion

The Nyquist-Shannon sampling theorem also applies to the trajectory of the moving object in oder to guarantee accurate and unambiguous reconstruction of the trajectory from a sequence of pose estimates. Oscillations of the object pose at a frequency above half the sampling frequency are not detectable. This circumstance is particularly important whenever tracking is integrated in the robot closed-loop control. The robot can follow only those trajectories that can be reconstructed from the sampled sensor signal.

The sampling frequency is not only responsible for the reconstruction of the past trajectory but also for the accuracy of pose predictions for future time instants. Because of the typical inaccuracies in pose estimation, future poses cannot be predicted without errors despite a perfectly known motion model. The more frequent the poses are estimated, the smaller the time gap becomes that needs to be bridged by the prediction. Thus prediction error is reduced accordingly.

In the case of freely moving objects, such as typical objects carried by humans, the applied forces and accelerations are not known. As a consequence, future poses cannot be predicted accurately.

1.1.2 Properties of Space

The task of visual pose estimation in 6 DoF consists of the registration of a three-dimensional object or scene model to the camera views. The complexity and accuracy of the estimation process depends to a large extent on the appearance and the shape of objects and the scene. In some cases, the environmental properties can be controlled and shapes and appearances can be predetermined in order to simplify the estimation task. However, for general settings, this is neither desired nor possible and thus the task becomes more difficult.

The perceptual diversity raises the demands on the pose estimation in several aspects. A compromise between the ambiguity of motion estimation and the velocity of motion has to be found to cope with these aspects in real-time. In the following, the major sources of inaccuracies with respect to the pose estimation problem for a single image are listed, and considerations on the computational complexity of the employed methods are given.

Sensing Ambiguity

The first source of ambiguities is given by the sensing equipment. As previously stated, the sampling process introduces aliasing effects if the Nyquist criterion is not met. Sampling on the sensor chip avoids aliasing effects in part due to the spatial dimension and the resulting low-pass filtering effect of the photo-sensible cells.

Image Matching Ambiguity

The second source of ambiguities or in accuracy is given by the process of extracting information from the image. The kind of information considered depends on the measurement domain of the pose estimation method. The two most important types of measurements are

- local appearance cues (features) and
- global appearances descriptions.

The former type denotes prototypical groups of pixels extracted at prominent image positions. The latter type comprehends group of pixels at the level of the complete appearance of the object in the image.

Obviously, the accuracy of the extracted information with respect to the actual position depends on the degree of conformity of the expected measurement with the ground truth. The versatility of the model describing the expected measurement under variation in pose as well as under variation in illumination determines to a large extent the accuracy and the robustness. Highest accuracy can be attained when the model considers the whole image formation process determined by the light sources, surface geometry and texture, as well as the properties of lens projection. Global appearance models usually aim at accounting for all these properties while local appearance models use simplified static patterns for the representation of distinct object points in the image.

Global Pose Estimation Ambiguity and Complexity

The ambiguity is directly related to the size of the considered pose manifold and the method of exploring the pose manifold. The degree of ambiguity handled by the exploration method determines its computational complexity. In accordance with the properties identified in section 1.1.1, the two major types of exploration are sequential exploration, such as gradient-descent optimisation methods, and exhaustive exploration methods, such as correlation-based methods.

The unambiguity of pose estimates is guaranteed for sequential exploration methods only within the area of convergence, which is related to both the employed objective function and the object appearance. On the contrary, exploration with exhaustive methods is not affected by local ambiguities. If a globally distinct solution to the estimation problem exists, global pose estimation can be pursued. In principle, both of the exploration methods can be adopted for feature-based pose estimation as well as for appearance-based pose estimation.

In the case of feature-based approaches, the computational complexity of feature extraction, matching, and pose estimation add up to give the overall complexity. Two main parameters determine the amount and allocation of the computational costs, i.e., the complexity of the feature model and the extent of the explored area. The combination of both determines the uniqueness of a feature within the explored area. So, increasing the complexity of the feature model increases the computational costs of feature extraction while lowering the combinatorial expenses for finding correspondences. Decreasing the area of exploration lowers the costs of both feature extraction and matching due to a more confined image region to be pre-processed and due to the reduction of potential matching candidates. Hence, local exploration is expected to be computationally less expensive than global exploration.

In the case of appearance-based approaches the efficiency depends on the matching costs and on the number of iterations for matching the model to the current appearance for a specific pose hypothesis. In contrast to sequential methods, exhaustive exploration requires many pose hypotheses to be tested and thus becomes inappropriate for real-time applications. Equivalent to features, the uniqueness of complete appearances is not merely determined by the matching criteria but rather by the surrounding scene and the extent of the considered image region. While both shape and texture define the uniqueness within the same object, the background affects the overall appearance ambiguity.

In summary, the ambiguity defines the reliability of pose estimates. Local exploration methods are preferred over global exploration methods in order to increase uniqueness and to decrease the computational complexity. In the case of sequential exploration, the uniqueness usually determines the upper distance of the initial pose estimate to the ground truth. Hence, the maximal object velocity results from the uniqueness and the runtime efficiency of the exploration algorithm.

Local Pose Estimation Ambiguity

The third source of ambiguity concerns the accuracy inherent to the correlation of a pose hypothesis with the information extracted from the image.

In the case of feature-based methods the correct pose relates feature positions either to points on the three-dimensional object or to feature positions in a second image. For the latter type of correspondences, the camera pose is typically estimated together with the scene model. Such approaches are referred to as Simultaneous Localisation and Mapping (SLAM) methods. The accuracy of feature-based pose estimation thus depends on the consistency of a potentially a priori unknown three-dimensional scene model with the observed feature positions.

In case of appearance-based methods, the accuracy is determined by the consistency of the appearance model with the currently sampled group of pixels. Misregistration is caused, for instance, by deficiencies of the model with respect to perspective distortions, illumination, or occlusion.

1.1.3 Properties of Time and Space

The application of robot interaction using visual sensors takes place simultaneously both in space and time. Existing limitations of the sensing hardware and the computation hardware cause many properties related either to space or time to affect properties of the other domain. In the following, the major interdependencies between these properties are listed.

Interdependencies of Temporal Sampling and Spatial Integration

Technological limitations force sensing to balance between the sampling rate and the image quality. The image quality assesses the ability to reconstruct the original, noise-free image from the discretised and quantised image signal. This ability is determined by the signal-to-noise ratio of a single quantised measurement (aka pixel) on the one hand, and by the signal-to-noise ratio of the spatial reconstruction of the discretised image on the other hand.

The signal-to-noise ratio related to a single pixel depends on the power of the scene radiance and the sensibility of the sensor chip in conjunction with the lens aperture and integration time. Usually, the radiation power is considered an uncontrollable external parameter. Since the sensibility of the sensor is fixed, the only controllable parameters are the lens aperture and the integration time.

The signal-to-noise ratio related to the spatial reconstruction depends on the spatial image resolution and on the degree of blurring. For static images, the degree of blurring is related to the lens aperture while for moving scenes it is related to the motion in the image plane and the integration period.

However, lowering the noise level with respect to a single pixel and lowering the noise level with respect to the spatial representation are concurrent goals. Usually, increasing the spatial resolution affects the size of the photo-sensible cells and decreases sensibility of the cells respectively. Higher lens apertures improve the signal-to-noise ratio of a single cell but introduce static blurring in the image. Last but not least, a shorter integration period decreases the amount

of motion blurring but increases the signal-to-noise ratio of a single pixel. In the end, the shortest appropriate integration time constitutes the lower bound of the sampling period.

Furthermore, the physical bandwidth supported by the transmission channels constitutes an additional limit of the sampling rate. Higher rates can only be obtained at the cost of spatial image resolution and vice versa.

Interdependencies of Temporal Sampling, Motion and Object Appearance

The radiation reaching a sensor cell changes with the object motion. The frequency of the signal over time is hereby related to the spatial frequency of surface texture, the pose of the object, and the direction of the object motion with respect to the camera. The latter induces motion of the projected surface points in the image, also known as optic flow. The radiation signal for a single sensor cell is therefore related to the radiation perceived on the sensor chip in the direction of optic flow. Consequently, faster motion shifts the frequency components of the perceived signal accordingly. The variations of the radiation signal originate from the combination of the object motion and the frequency components of the object appearance in direction of the corresponding optic flow.

In consideration of the Nyquist criterion, the sampling rate should meet the requirements set up by the object velocity, the object pose, and the object texture. Obviously, in the case of repetitive patterns on the surface texture, a violation of the Nyquist criterion prevents unambiguous motion estimation irrespective of the tracking method.

Interdependencies of Temporal Sampling and Spatial Exploration

In general, an error probability distribution can be set up to describe the uncertainty of pose predictions given knowledge, or simply assumptions, about the temporal correlation of the object pose. The uncertainties for current and future time instants have a direct impact on the computational costs of the estimation problem because they determine the domain to be explored.

Pose estimation methods typically explore the neighbourhood of initial pose hypothesis in three ways:

- regularly,
- sequentially, or
- stochastically.

The first two methods are considered here in more detail to reveal the specific correlation between the (temporal) sampling frequency and the computational costs.

Regular, grid-based exploration of the neighbourhood is characteristic for feature-based tracking techniques. The image is scanned for features in regular

spatial intervals within a determined window. This strategy can also be adopted for the exploration of the pose space in the neighbourhood of a hypothesis.

Suppose in the following that the neighbourhood is represented by a d-dimensional hypercube, where d corresponds to the number of DoF related to the estimation problem. The number of samples Q needed to regularly sample this volume for an object velocity v in a single DoF, for a temporal sampling frequency f, and for a spatial sampling frequency q in the DoF, is determined by

$$Q(v, f, q) = \left(1 + \frac{vq}{f}\right)^d \tag{1.1}$$

whereas, for simplicity, the formula encompasses rational numbers of samples. The computational power C_t provided by the target computer represents an upper limit for the evaluation of Q samples, that is

$$C_\mathrm{t} \geq f \; C_\mathrm{q} \; Q(v, f, q) \;\; , \tag{1.2}$$

where C_q is a factor corresponding to the computational costs per sample. Additional computational costs induced by increasing object velocities can be compensated, up to a certain limit, by increasing the temporal sampling frequency. Hence, the number of samples evaluated at each time instant decreases, whereas the number of samples evaluated per unit time is kept constant.

However, the number of samples per time instant is limited at the lower bound to 2^d for the exploration of a unit hypercube in d DoF. The maximal temporal sampling frequency is therefore determined by

$$f_\mathrm{max} = \frac{C_\mathrm{t}}{2^d C_\mathrm{q}} \tag{1.3}$$

and, accordingly, the velocity

$$v_\mathrm{max} = \frac{f_\mathrm{max}}{q} \tag{1.4}$$

specifies an upper limit at which objects can be tracked.

In the case of sequential exploration methods, an implicit or explicit objective function is iteratively minimised. In general, the shape of this objective function is a priori not known and highly non-linear. Hence, lower order approximations of the function cause the number of iterations employed in the minimisation to increase with the displacement of the initial pose estimate to the actual pose.

Let $s(t)$ denote the evolution of the relative error between the current estimate and the ground truth over time, which initially ($t = 0$) is equal to one and eventually converges to zero. Since convergence is reached only at infinity, the residual pose error adds to the prediction error in the consecutive frame. Suppose that the predicted pose corresponds to the pose estimated in the previous frame. Then, the pose error Υ_i at frame i accounts to

$$\Upsilon_i = \Upsilon_{i-1} \; s(P_\mathrm{t}/f) + v/f \;\; , \tag{1.5}$$

where $P_\mathrm{t} \propto C_\mathrm{t}^{-1}$ is a factor dependent on the computing power. Obviously, the sequence (Υ_i) converges to

$$\lim_{i\to\infty} \Upsilon_i = \frac{v}{f - f\ s(P_\mathrm{t}/f)}\ . \tag{1.6}$$

This limit is a strictly monotonic decreasing function of the sampling frequency f and hence, the residual $\lim_{i\to\infty} \Upsilon_i$ is minimised for the sampling frequency approaching infinity. According to the rule of Bernoulli-l'Hospital, the minimal gap between the estimate of the tracking procedure and the true object pose is given by

$$\lim_{f\to\infty} \lim_{i\to\infty} \Upsilon_i = \lim_{f\to\infty} -\frac{v}{P_\mathrm{t}\ s'(P_\mathrm{t}/f)}\ . \tag{1.7}$$

Increasing the sampling frequency lowers the gap asymptotically to the above theoretical limit.

As an example, in the case of exploration with linear convergence where $s(t) = a^t, 0 < a < 1$, the limit corresponds to $-v/\left(P_\mathrm{t} \ln(a)\right)$. In the case of exploration with quadratic convergence where $s(t) = a^{2^t-1}, 0 < a < 1$ the limit is given by $-v/\left(P_\mathrm{t} \ln(2) \ln(a)\right)$.

Obviously, both the object velocity as well as the computing power affect the gap between the initial pose estimate and the actual pose. Due to the limited area of convergence of sequential sampling methods, the gap is not allowed to exceed a certain bound. Conversely, the supported object velocity is bound to an upper limit related to the computational power.

1.2 Robot Control

The interaction of intelligent systems with their physical environment requires the actuation of moving mechanical parts. The combination of these mechanical parts form a *robot* if actions are performed autonomously. However, the degree of autonomy differs from non-adaptive autonomy to adaptive autonomy depending on the robots capability to react to changes in the environment. In classical industrial automatisation, for instance, robot applications are designed for countable recurrent environmental states. The program is not adaptive to variations apart from these states because the robot is taught to move on predefined trajectories. Instead, higher flexibility is obtained by considering current sensor readings of the environment and moving the robot accordingly. Here, vision sensors offer the richest source of information whereas the interpretation of the corresponding data is arbitrarily complex. In the following, only intelligent robots are considered, i.e., robots adapting their behaviour to the environment on the basis of sensor readings.

In general, the information provided by a sensor is mapped to suitable actions either by commanding velocities or positions (or angles) of the individual actuation units. The target velocities or target positions are reached by employing direct control methods or hierarchical control methods. In the first case, the commanded values affect the electrical motor impulses directly through a unique control loop that takes into account the whole model of the robot. In

the latter case, dynamics and kinematic properties are tackled separately by means of prioritised levels of control: The lower level deals with dynamic issus, whereas the higher level generates suitable velocity signals consistent with the robot kinematics.

In practice, robot control is not limited to the surveillance of the actual and desired velocities or positions. Many other constraints are necessary to guarantee smooth operation without incidents. In the following, the major static and dynamic constraints are listed, as concerns the visual servoing of moving objects.

1.2.1 Configuration Independent Properties

Some constraints are independent of the dynamic of interaction and can be analysed a priori based on the specifications of the robot.

The first obvious constraint is given by the kinematics of the robot. The number of joints of the robot, the type of joints, as well as the length of the connecting links determine the workspace of the robot, also known as dexterous workspace. With typical joints such as the rotatory or linear joints, at least 6 joints have to be combined to position the robot end-effector freely in 6-DoF Cartesian space. Restrictions in the dexterous space arise from the specific link lengths, joint limits, and singularities in the joint configuration. Singularities occur whenever actuation of two or more joints cause the same motion of the end-effector in Cartesian space.

The second constraint is represented by the characteristics of the joint dynamics. The specifications of motor dynamics as well as the friction characteristics determine the potential velocities and accelerations of each joint and the end-effector.

Finally, joints are arbitrarily stiff. Lack in stiffness usually affects the absolute position accuracy of the robot, as well as the transient behaviour because of dynamical coupling between joint motion and motor motion.

To recapitulate, robot control for the interaction with a moving object faces at least three configuration independent constraints, i.e., workspace limitation due to the robot kinematics, limitations of the joint dynamics, and inaccuracies in absolute position.

1.2.2 Configuration Dependent Properties

The interaction of the robot with a moving object introduces configuration dependent constraints on the robot control. In all configurations, the stability of the control mechanisms must be guaranteed such that bounded object movements lead to bounded robot movements.

In the course of interaction, the robot can become under-actuated or over-actuated depending on the joint configuration. In the former case, arbitrary robot movements exist that satisfy the constraints of the task. In the latter case, no robot trajectory exists that fulfil the requirements of the task.

Over-actuation occurs, for instance, if the objective is to reach a certain pose in 6 DoF and the robot consists of more than 6 joints, or the objective is

to translate the end-effector in 3 DoF with a 6-joint robot. For such tasks, the robot can exploit the redundant DoFs in order to meet additional constraints such as optimising given criteria. If the object motion is known a priori, then the robot can simultaneously follow this motion and avoid joint limits as well as singularities in the joint configuration.

As as result, any motion generation scheme becomes ill-conditioned close to singularities and therefore robot control should avoid these singular configurations whenever possible.

1.3 State of the Art

Any type of interaction between humans and robots can be categorised with respect to the robot input and output channels. In general, communication with humans[3] is forced to be acoustic, visual, and/or physical. The present work focuses on systems performing physical actions, i.e., actuated robotic systems, for the purpose of service robotics. Thus, all other types of machines providing only auditive or visual feedback, e.g., desktop computers, are not considered, which excludes a big part of the existing works in the very general field of *human-machine interaction*.

With regard to the robot input, contact sensors such as tactile sensors have been the first devices used to command robots because joysticks and keyboards were the primary interfaces to computers at the time. Subsequently, force/torque sensors have been developed, by eventually culminating in robots that perceive and react to external forces [67]. Different kinds of haptic interfaces are used in the field of tele-robotics to guide a robot and to give immediate feed-back on collision and forces exchanges with the user, e.g., by means of pen-style devices[4] or exoskeletal systems[5].

Among contactless sensors, passive acoustic sensors became increasingly popular input devices thanks to the achievements in speech recognition and the availability of commercial products [74, 57]. Certainly, speech recognition brought human-machine interaction significantly further. However, when it comes to manipulation tasks, it only allows for robot (re-)actions at the slow rate of the user instructions.

Active acoustic sensors, instead, enable the robot not to wait for commands but to react autonomously, and possibly in advance to collisions. The low cost of ultrasound sensors and their ability to measure distances led active acoustic sensors to become a standard equipment of mobile robots. However, these sensors provide only coarse information of the surroundings, and, thus, are not suited for accurate manipulation tasks.

On the other hand, optic sensors arrays, in particular digital cameras, gather information at a much higher spatial resolution. However, the interpretation of this information proves to be more difficult. Typically, the complexity of the visual input only allows to *either* process a detailed scene at a slow rate

[3] neglecting the possibility of direct electrical interfaces

[4] e.g. the PHANTOM device: http://www.sensable.com

[5] e.g. the CyberGrasp device: http://www.immersion.com

compared to the robot control cycle (e.g. [66]), *or* to seamlessly control the robot on the basis of simple visual stimuli (e.g. [55, 137]) or constrained object motion (e.g. [1, 15]).

Among the above mentioned types of input devices, digital cameras are best suited for task of precise human-robot interaction. This type of sensors allows the robot not only to obtain user commands, e.g., via gesture recognition, but also to immediately and precisely react on environmental changes. Especially when it comes to close interaction, such as grasping an object from the users hand, the advantage of digital cameras becomes evident. The sensor allows to gather information about the pose of the object in space, which is then needed to control the robot.

From the methodotical point of view on image processing, the best results for interaction with moving objects were attained with so-called feature-based approaches, in particular based on contour or edge features (e.g. [147, 39]). However, these approaches are known to suffer from cluttered background and textured surfaces.

Feature-based approaches are typically limited to simple or piecewise planar shapes with a homogenous surface texture. Complementary types of object, i.e., textured and free-form surfaces, are not covered by the mentioned methods. This thesis aims exactly at coping with these objects in order to eventually allow the robot to "see" and interact with *arbitrarily shaped* and *non-homogeneously* textured objects.

1.4 Thesis Outline

This work addresses the problem of tracking a known, free moving, object in 6 DoF, and of catching the object with an autonomous dexterous robot. The previous sections outlined the basic constraints imposed on both visual perception of motion (section 1.1) and actuation of robotic systems (section 1.2). These constraints are inherent to the problem and equally affect all potential solutions.

1.4.1 Contributions

The thesis comprises novel contributions in several fields associated to visual servoing.

First of all, the work explores pure appearance-based real-time tracking methods in 6 DoF. Here, three major improvements are reached: the consideration of full-perspective projection, the support of arbitrary surface shapes, and the computationally efficient prediction of pose Jacobians based solely on the reference view. The support of full-perspective cameras allows to apply the methods not only to the task of object tracking but also to the potential estimation of ego-motion in indoor scenes.

Previous approaches introduced constraints either in the camera model (e.g. [21, 116, 148]), the supported object shapes (e.g. [30, 29, 7, 14]), or both. Still, the camera model based on weak perspective projection enjoys great popularity due to the linearity of the underlying mapping. This model linearises the

perspective projection by considering all object points to be equidistant with respect to the camera. Linearisation is also adopted to object shapes, which are typically assumed planar or piecewise planar. The present work, however, employs a full-perspective projection for a camera with known intrinsic parameters. This allows for accurate modelling of perspective distortions in scenes with small and large variations in depth. The latter occurs especially in configurations where the object is close to the camera. Moreover, the restriction to (piecewise) planar surfaces (e.g. [7, 34]) is successfully abolished by representing surfaces as unordered, textured, three-dimensional point clouds. No linearisation is applied at this level allowing for any potential free-form surface to be modelled. In the effort to provide real-time methods, previous approaches did not succeed in lowering the perspective constraint or the constraints on the surface shape.

In addition, this thesis presents an efficient prediction of the non-constant and computationally expensive motion Jacobian. The corresponding analytic formula is valid for surface patches with a first-order differentiable shape and texture. In practice, infringements of the latter assumption are tolerated and surface points at discontinuities of the surface curvature can be excluded from the object model. Hence, these requirements do not represent a serious limitation.

In the field of illumination compensation, the present work simplifies the previous formalisations of pattern matching in the orthogonal illumination subspace [13, 62]. Additionally, it is shown, contrarily to the expectations, that the introduction of the more general model of the illumination subspace does not necessarily increase the tracking performance.

With respect to the initialisation problem of the tracking procedure as well as to its re-initialisation to recover from "lost objects", a new, purely appearance-based tracking cascade is presented, as opposed to [140]. Unlike feature-based approaches, the cascade supports free-form, textured surfaces on all stages.

Moreover, from an experimental point of view, this thesis includes systematic evaluation of the appearance-based tracking approaches. An experimental database is built so as to reflect multiple objects and multiple object poses. The database comprises object poses fairly distributed within the visible volume of the perspective camera. Finally, the devised methods are successfully evaluated for interaction with a 7-DoF lightweight robot actuator and a dextrous anthropomorphic hand showing the desired capabilities for human-robot interaction.

1.4.2 Overview

In order to detail the approaches required for completing the task, chapter 2 enumerates the major scientific areas involved, and presents the related publications. In general, *visual servoing* is the scientific field that comprehends both aspects of computer vision and robot control.

Some researchers address visual servoing from a control point of view. Accordingly, the variability of appearances in real world is greatly simplified. In-

stead in this work, special attention is paid to these appearances and to the capability of the approach to handle real-world objects. In contrast to feature-based visual servoing typically relying on three-dimensional edges, a novel, complementary approach is presented in chapter 3, which considers arbitrarily textured free-form surfaces. See figure 1.2 for an illustration of the relation to the contents mentioned in the following.

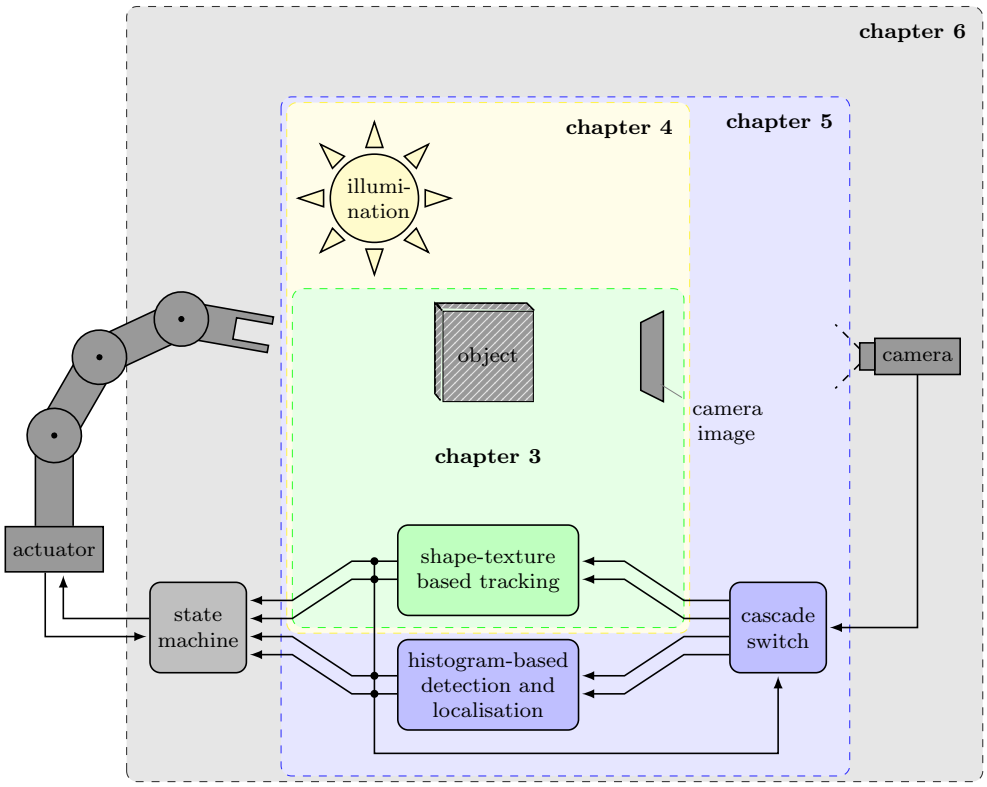

Figure 1.2: Graphical illustration of the content of the thesis.

While feature-based methods perform badly in presence of cluttered background, the class of texture-based tracking approaches are affected by varying illumination. In chapter 4, common methods to handle the illumination variations are analysed and evaluated for suitability

Initial object detection and localisation, as well as the ability to robustly re-localise lost objects, are mandatory requirements for the acceptance of the application for human-robot interaction. To this end, a novel cascade of hierarchically organised appearance-based detection and tracking stages is devised. As its main contributions, it ensures simple initial object detection, accurate tracking, and seamless transition to object re-localisation, as described in chapter 5.

The physical constraints of human-robot interaction are considered in detail, and a generally suitable configuration of robot, camera, and interaction area is discussed. The appearance-based hierarchy is integrated in the robot control in chapter 6 via a position-based interface, which eventually allows for visual servoing and to succeed in capturing moving objects.

Finally, chapter 7 discusses the main achievements and proposes subsequent investigations and future developments in this field.

2
Related Work

The design and implementation of visual servoing applications requires knowledge in the fields of computer vision, optimisation theory, and robot control. In detail, these applications consist of the tasks of image acquisition, image interpretation, robot motion control, and joint actuation. In general, the overall objective, as well as the objectives of the subtasks, can be implicitly or explicitly formulated as optimisation problems, which can be addressed with methods from optimisation theory.

In the context of this work, the overall objective is to approach and grasp freely moving objects with a dextrous manipulator. In order to succeed, the robot end-effector has to move to a defined position and orientation relative to the target object. The movement is performed by either estimating the current object pose or estimating the incremental motion to the target pose. The configuration of sensors and actuators as well as the specific formulation of the task to be performed are handled within the superordinate field of visual servoing.

Hereafter, the problem of motion estimation is subsumed by the problem of pose estimation. Accordingly, three problems are identified within the above scientific areas, effectively capturing the main topics, namely

1. pose estimation,
2. handling of variations in illumination and occlusion, and
3. pose prediction.

The first topic addresses the challenge of pose estimation from image data in a general way, e.g., ranging from the extraction of information up to the estimation of the motion between two views. The next issue concerns the robustness of such estimation to variation of illumination or to partial occlusion of the object. The robustness is also affected by the employed model of the object dynamics, which allows for the prediction of object poses in future time instants.

The following sections review the relevant scientific literature in the above-mentioned areas. In particular, section 2.1 reviews main scientific works within the established categories of visual servoing. The pose estimation problem is addressed at a general level in section 2.2 by classifying the approaches with respect to the domain of motion, the type of visual information employed, and the representation of the underlying object model. Methods handling changes in illumination are considered in section 2.3. The approaches rely either on image information invariant to illumination, attempt to compensate the variations, adopt statistical methods for robust parameter estimation, or require certain constraints to hold. Section 2.4 outlines the approaches supporting models of object dynamics for the prediction of future poses. Finally, section 2.5 discusses the existing approaches and provides the motivation for the development of the novel methods presented in the following chapters.

2.1 Visual Servoing

Controlling a robot by means of information extracted from camera images is defined visual servoing. The complexity of the visual servoing applications depends primarily on the task, the setup, and the capability of the robot. For the setup, two configuration of camera to robot are possible, namely

- eye-in-hand configuration and
- eye-to-hand configuration.

In the former setup, the camera is rigidly mounted on the last link of the robot [4, 92, 15, 87, 82], while in the latter case the camera stands apart from the robot [131]. Eye-in-hand configurations have the advantage that more accurate information can be extracted from the image as the robot comes closer to the object of interest. Which of the two setups is used depends on the the accuracy required from the task, the robot capabilities, visibility constraints, and other requirements of the specific application.

In control theory, two methods are available to deal with absolute position accuracy of the robot and the requirements, namely either

- end-effector closed-loop control or
- end-effector open-loop control.

In the former method, the information extracted from the images depends simultaneously from the motion of the target and from the motion of the robot end-effector. This coupling allows to approach an object irrespectively of the robot position accuracy [131, 92, 87]. The method is implicitly used for an eye-in-hand configuration, while an eye-to-hand configuration requires that both the robot end-effector and the target are tracked simultaneously. In the latter method, the extracted visual information does not change with the motion of the end-effector. Thus, end-effector open-loop control is given only for those eye-to-hand solutions that do not track robot motion.

Visual servoing systems can be classified with respect to the type of control architectures in

- direct control and
- hierarchical control.

In the former type of architectures, the robot is controlled within a single control loop. In the latter case, two or more control loops exist. These control loops are organised hierarchically so that an *outer* control loop determines the parameters for an *inner* control loop. Hierarchical control architectures allow for the transition between high control cycle frequencies at an inner loop and lower cycle frequencies at the respective outer loop.

Hierarchical control is typically used for visual servoing where the image acquisition and recognition operate at frequencies significantly lower than the robot cycle frequency. In extremis, information is extracted from the image only once at the beginning of a task. This information is used thereafter in an inner control loop to autonomously reach the target position. These approaches, known as dynamic look-and-move approaches, are only appropriate for static scenes.

The most prominent distinction between visual servoing algorithms is made with respect to the task space, differentiating between

- image-based visual servoing and
- position-based visual servoing.

The following sections briefly describe these two categories. Refer to [31, 32] for a recent tutorial on the topic.

2.1.1 Image-based Visual Servoing

In image-based visual servoing the robot control loop minimises an error in the image plane [136, 137, 113]. Thus for eye-in-hand configurations, the approach implicitly tries to keep the target in the field of view of the camera.

Originally, image-based visual servoing was intended to overcome hand-eye calibration by learning the visual-motor model a priori, e.g., [4]. The visual-motor model directly relates image variations to joint variations. This relation however, expressed at first order by the image-to-robot Jacobian depends on the robot pose. To simplify the computation, the image-to-robot Jacobian is decomposed into Jacobians relating image information to camera pose, camera pose to tool-centre pose, and tool-centre pose to joint positions. The former Jacobian is also known as image Jacobian and the combination of the latter two is known as robot Jacobian. The on-line computation of the pose-dependent robot Jacobian can be supported by off-line calibration steps. In detail, the camera-to-tool-centre Jacobian can be determined off-line through hand-eye calibration [133] and the tool-centre-to-joint Jacobian can be parameterised appropriately by means of the prior identification of the robot kinematics.

The majority of approaches combine these calibrations with the image Jacobian, which is determined frame by frame [136, 137, 87, 113]. The estimation of the image Jacobian allows to map image velocities to joint velocities via the

robot Jacobian for the current robot pose. Image-based visual servoing approaches do not require accurate camera or robot calibrations since calibration errors affect the trajectory but not the final pose.

2.1.2 Position-based Visual Servoing

In position-based visual servoing the robot control loop minimises a Cartesian error [51, 15] between the target pose of the robot and the current pose. Basically, robot vision is separately kept from robot control imposing a Cartesian interface between both modules and employing a hierarchical robot control architecture. It allows to decouple the processing rate of the vision algorithm from the robot cycle frequency. The drawback of position-based control, however, is that visibility of the target is not ensured in the case of an eye-in-hand configuration.

Traditionally, position-based visual servoing relies on a hand-eye calibration and a calibrated robot kinematic [92, 82]. Approaches exist learning the image to robot Jacobian on-line [131]. However, position-based visual servoing is not robust to calibration errors, which affect the trajectory *and* the final pose of the robot.

2.2 Pose Estimation

The problem of *pose estimation* has a long history in computer vision and traces back to the registration problems of early computer vision. The objective is the determination of the parameters of a coordinate transformation between a body frame and the sensor frame. Usually, rigid-body transformations are considered, which preserve distances and angles. The term *motion estimation* is closely related to pose estimation whereas, in this case, an arbitrary but fixed body coordinate frame. As a result, the object motion between two time instants is estimated in contrast to the object pose relative to the sensor frame. Hereafter, no distinction is made between approaches for pose estimation and motion estimation since one formulation can be transformed to the other with the appropriate combination of single rigid-body transformations and a particular reference frame.

Significant differences between the approaches in motion or pose estimation stem from the available a priori information, the kind of information extracted from the image, and the type and number of parameters to be estimated. These approaches are classified according to three distinct characteristics

1. the motion domain,
2. the measurement domain, and
3. the interpretation domain.

These characteristics are analysed in the following subsections. First, the different types of motion models, from rigid-body transformation to deformable models are considered. Then, the different types of information extracted from

the image are addressed. Finally, an overview of the object models is given that are typically used for motion or pose estimation.

2.2.1 Motion Domain

The motion domain establishes the relation between sensor readings at one time instant (or sensor position) to the sensor readings at another time instant (or position). This relation is typically expressed analytically by the following parametric transformation

$$\mathbf{x}' = m(\mathbf{x}, \boldsymbol{\mu}) \ , \quad \mathbf{x}, \mathbf{x}' \in \mathbb{R}^N, \boldsymbol{\mu} \in \mathbb{R}^M \tag{2.1}$$

known as motion model, where $\mathbf{x}'$ and $\mathbf{x}$ are N-dimensional points sensed at different time instants (or positions) and where $\boldsymbol{\mu}$ is a M-dimensional motion parameter vector.

A variety of motion models have been employed for tracking and motion estimation. In the following, the models are categorised according to the dimensionality N into motion on the image plane ($N = 2$) and in three-dimensional space ($N = 3$).

Further distinction is made with respect to the angle and scale preserving characteristics of the transformation $m(\cdot, \cdot)$. Indeed, as an example, rigid-body motions preserve angles and scale while deformation transformations do not. The transformations called articulated motion deserve particular attention because they resemble deformation models at a global scale and rigid-body motion on a local scale. In this case, the body of the moving object is composed of several connected segments, so that the object changes its appearance globally according to the current configuration of the segments. However locally, each segment is rigid and its motion is constrained by the motion of the connected segments.

Motion in $\mathbb{R}^2$

Motion in the image plane has been investigated since the early stage of computer vision. The most popular motion models are

- translation and scale,
- affine distortion,
- polynomial distortion,
- linear shape models (ASM),
- projective invariants and
- unconstrained motion.

Among these models, affine distortion is in widespread use, e.g., [91, 111, 61, 62, 7, 87, 73], mainly because of its simplicity and linearity despite its expressive power. The model allows to describe rigid-body transformation in the image

consisting of rotation and translation as well as deformations such as shearing and scaling. It has been suggested [138, 128] to restrict tracking of object patches to translational parameters as long as the residual of affine motion does not exceed a predefined threshold. Recently, its validity has been confirmed for a medical application [59]. The restriction to the affine parameters of translation and scale is also in use, e.g., for tracking based on mean-shift [104, 105, 36].

Other deformation models are used to cope with more complex, non-rigid motion. Polynomial distortion models, for instance, have been applied to track facial expressions [115, 17]. The same objective is followed by linear shape models, which constrain potential variation to a linear combination of motion vectors [40, 7, 41, 95].

In general, the above methods are used to register two sets of measurements by parametric motion models. The independent, two-dimensional motions of the measurements in one camera view to the corresponding measurements in another view is known in literature as optic-flow. Sparse optic-flow can be computed for salient image positions without the consideration of parametric motion, that is, without constraining two-dimensional motion at these positions. Dense optic-flow, however, is an ill-posed problem. In order to find plausible optic-flow for all pixels of an image, the motion has to be regularised [69].

Such regularisation usually follow smoothness constraints in the optic-flow field. This is a first step toward the introduction of some kind of knowledge on the shape of the observed three-dimensional scene. A special type of constraint, though not directly related to optic-flow, is the projective invariance explored in [85], which restricts the manifold of possible deformations.

It can be observed that the motion models in the image plane aim at compensating two types of distortions. The motion models are adopted to cope either with non-rigid three-dimensional deformations or with the perspective distortions of unknown but rigid three-dimensional objects. Thus, more accurate motion models can be devised by taking the three-dimensional structure and the perspective distortions explicitly into account.

Motion in $\mathbb{R}^3$

Motion estimation in three-dimensional Euclidean space is appropriate whenever perspective distortions arise or when the three-dimensional transformation between the object frame and camera frame is requested. The latter is mandatory for position-based visual servoing applications where the robot is commanded according to the three-dimensional information extracted from the image.

The most common motion models in three dimensions, which have bee investigated in the computer vision literature, can be classified according to their projection model as

- orthographic projection,
- weak perspective projection,
- full perspective projection, and

- homography mapping.

The simplest and most restricted projection model is orthographic projection, which neglects any perspective distortion. Instead, three-dimensional object points are projected perpendicularly to the image plane and neither the distance nor the scale of the object is determined [30, 65, 12]. Weak perspective projection extends orthographic projection by estimating additionally the overall distance or scale [21, 18, 116, 148, 49]. Obviously, these two models are only legitimate when the object extension in depth is small with respect to the distance to the camera.

Therefore, to obtain validity for a broad range of scenarios many approaches, especially in visual servoing, rely on the perspective projection model [3, 79, 80, 130, 10, 146, 45, 65, 29, 131, 75, 46, 124, 50, 63, 92, 143, 126, 34, 94, 25, 26, 35, 113, 38, 86, 101, 107, 20, 100, 60, 82]. For the special case of planar surfaces, the usually non-linear constraints of full perspective projection can be expressed by a linear homography mapping [93, 23, 42, 24, 14].

In the case of articulated objects, suitable constraints have been successfully imposed so as to estimate three-dimensional motion of the individual segments consistent with the motion of the connected parts [50].

Apart from rigid objects, also deformable objects have been considered by estimating the deformation parameters in addition to the rigid-body motion [109, 114, 9].

2.2.2 Measurement Domain

The type of information gathered from images represents the most salient criterion for the classification of pose estimation approaches. Considering the level at which the extracted information form a unit, the following taxonomy is derived:

1. methods based on local appearance cues,

2. methods based on global appearance representation,

3. methods based on global appearance description

4. hybrid methods.

The first type of approaches looks for recurrent local groups of pixels, so called features, and estimates the object or camera pose by evaluating the location of these features in the image. The second type of methods, instead, directly considers the global appearance information by analysing the object at pixel-level in accordance with an overall, shape-dependent sampling rule. In the case of appearance-descriptor based methods, the collected feature information or pixel information is further condensed into a single low-dimensional vector. Sometimes, the type of extracted information is combined in the so called hybrid methods.

Local Appearance Cues

The extraction of local cues from a data set is a natural approach to reduce the dimensionality in information. Also pose estimation takes advantage of feature extraction by interpreting the arrangement of a reduced set of selected object points. These points are matched either with the corresponding points on the three-dimensional object or with the corresponding points in another image. The location of the correspondences in the measurements reflects a certain pose.

Three classes of feature-based motion estimation algorithms exist according to the dimensionality of feature location. These are

- geometric (3d $\leftrightarrow$ 3d),
- projective (3d $\leftrightarrow$ 2d), and
- visual (2d $\leftrightarrow$ 2d)

correspondences, where 3d denotes three-dimensional points on the object and 2d indicates correspondences in the two-dimensional image. An additional differentiating factor is the plurality of the established correspondences. Feature-based motion estimation algorithms can handle either established correspondences of two feature sets (*one-to-one* relations) or yet to establish correspondences (*one-to-many* relations).

Geometric 3d-3d correspondences typically rely on range images, such as the ones generated by stereo algorithms or time-of-flight cameras. Rigid motion estimation is solved for the simple case of *one-to-one* relations [52, 68] with the most prominent solution based on the singular-value-decomposition [6]. If the point sets represent irregular samples of an object then the motion can be estimated with the iterative-closest-point algorithms [16, 83] as long as object rotation is small.

Projective 3d-2d and visual 2d-2d correspondences show higher popularity for motion estimation compared to pure 3d correspondences since they are based on single (and relatively inexpensive) camera modules for 2d imaging. The former type requires a 3d model of the object, while the latter implicitly constructs a 3d model. These algorithms originate from feature-based optical-flow methods and are also known as simultaneous-localisation-and-mapping (SLAM) or structure from motion (SFM) approaches.

Generally, procedures for feature-based motion estimation consist of three consecutive steps, namely

- feature extraction,
- feature matching, and
- pose estimation,

where the first or latter two steps are often tightly coupled. The type and discriminative power of the extracted features are mainly responsible for the complexity of the matching step. Once the correspondences are established, the object motion or pose can be estimated.

The most common feature types employed for projection-based and image-based estimation are point features [130, 146, 85, 90, 149, 21, 20, 92, 26], contour features [9, 131, 50, 63, 39, 126, 113, 38, 87, 107], area features [25, 94, 143, 86, 107, 82] and a combination of them [84]. Point features and contour features can be extracted from the image without prior information, but they limit the discriminative power of the feature to the respective class. Area features, instead, gain discriminative power with the size of the image patch they represent. Obviously, a trade-off exists between small, ambiguous patches and large patches, which may suffer from perspective distortions.

The correspondence problem is solved either by locally searching in temporally consecutive images [146, 10, 131, 50, 63, 126, 25, 26, 94, 143, 113, 38, 87, 86, 107, 82] or by finding the best match within all features under geometric and appearance constraints [86, 107] or geometric constraints only [21]. Some publications do not consider the problem at all and establish the correspondences manually [130, 2, 90, 93, 149, 20, 92, 60].

Accordingly, the object pose is estimated for given 3d-2d correspondences either locally by means of regression techniques [146, 2, 131, 46, 50, 63, 92, 126, 25, 26, 94, 143, 38, 87, 60, 82], semi-locally through particle filters [20], or globally convergent [90]. Recently, a globally optimal approach has been devised for this problem [76]. The global pose can be estimated for ambiguous correspondences with random sampling consensus (RANSAC) techniques [86, 107] or factorisation methods [21]. In this case, feature correspondences are simultaneously established.

In the case of visual 2d-2d correspondences, the estimation of motion implicitly determines the 3d structure of the object or scene at the same time. This field has been intensively studied (see [64] for a detailed insight into the topic). It is known that structure and camera translation can be estimated only up to scale unless additional metric information is available. While the motion of calibrated cameras can be estimated for two views through the essential matrix, e.g., [130], with a translation up to scale, the motion of an uncalibrated camera can be determined only from multiple views because of perspective ambiguities. Linear methods have been devised to estimate motion and structure when additional constraints apply [117, 118, 72]. Otherwise motion and structure are iteratively estimated through non-linear minimisation [11, 119, 25, 26, 110, 60], eigenvalue decomposition [149], factorisation [139, 141], or, most importantly, through bundle adjustment, e.g., [142, 127, 97].

Appearance-based Methods

In contrast with the above mentioned methods, appearance-based pose estimation does not look for certain features in the image individually. Instead, the overall appearance of the object is gathered from the image by picking a collection of pixels. The sampling process requires some kind of knowledge on how the pose is related to the appearance of the object. This knowledge is either given by a database of view dependent appearances or through a three-dimensional geometrical surface model.

View-based methods take pictures of the object off-line from as many poses

as possible and compare them on-line with the current image, e.g., [145]. Here, the image has to be segmented correctly into foreground (object) and background prior to being stored into the database. Obviously, runtime performance and estimation accuracy depend on the number of pictures stored in the database.

Geometry-based, or simply model-based methods are able to reproduce[1] the appearance from any view by assuming a certain three-dimensional model of the surface in connection with a second view of the surface. Therefore, two possibly consecutive views of the object are related by the geometric model and either by the camera motion between the two views or by the corresponding object poses. If one view is a priori registered with the 3-d model then a textured 3-d representation of the object can be set up. This representation allows either

- image-based or
- texture-based

correlation with the current view. Image-based approaches compare the rendered image of the textured 3-d model for a given pose with the current image, while texture-based approaches compare the texture of the 3-d model with the current image inversely mapped to the 3-d model.

Generally, the task of finding the right pose or motion for the current appearance can be formulated as an optimisation problem. Different numerical optimisation methods have been employed in literature to solve the problem, e.g., graph cuts [54], Monte-Carlo sampling [49], singular-value decomposition [42], general factorisation methods [21], and general non-linear minimisation methods [129, 12, 116, 14, 109, 18]. The simplest non-linear minimisation methods adopted for motion estimation are gradient-descent methods [30, 17].

Faster convergence compared to gradient-descent methods is achieved for second order approximations of the error function, as implicitly adopted for instance in Kalman filter approaches [45]. Standard Newton methods have been also applied [115] by relying on an analytically derived Hessian matrix. However, Newton methods with a Gauss-Newton approximation of the Hessian show in practise higher robustness and convergence compared to standard Newton methods [112].

Herein, the computation of the image Jacobian with respect to the object pose, hereafter simply called pose Jacobian, represents the computationally most expensive part of minimisation. Usually, the Jacobian is recomputed at each iteration [91, 61, 62, 124, 23, 24, 114, 148, 100] but some approaches assume a constant Jacobian and thus also a constant Hessian [40, 29, 75, 7, 116, 95, 34, 35, 73, 101]. This assumption is valid for planar motion models such as homography but not for rigid-body motion of non-planar surfaces under full perspective projection.

Generally, the approaches relying on local, non-linear optimisation with an on-line or off-line computed pose Jacobian are denoted *pixel-based* methods or direct methods in the context of optic-flow. These approaches collect information (usually intensity) at selected coordinates in the image, and map this

[1] if the illumination is known

information directly to motion parameters. Single intensity variations, however, can only contribute to motion components in normal direction of the local intensity profile[2]. Therefore, multiple pixel measurements are considered and an appropriate model links variation in motion to variation of single intensity values.

Examples of such methods for tracking in the image plane are given by affine motion [91, 61, 62, 73] and polynomial distortion [115]. Linear two-dimensional shape models have also been considered for tracking facial expressions [41, 95]. The estimation of motion in three-dimensional space, instead, requires the description of three-dimensional object shape. Methods have been devised primarily for primitive surfaces, such as planes [48, 30, 45, 7, 23, 42, 14, 24, 113], composition of planes [34, 35], cylinders and spheres [30], free-form surfaces [124, 114, 101], and linear three-dimensional shape models [116, 148]. A special case is the estimation of pure rotational motion [129] where variation in depth can be neglected. Linear three-dimensional morphable models have also been explored for representation and tracking of facial expressions [116, 148], mostly at the cost of several approximations of perspective mapping.

The information of three-dimensional structure required for non-primitive surfaces is either provided a priori [124] or by up-to-date depth images [79, 80, 65]. If shape information is missing then structure and motion have to be recovered simultaneously given a sequence of images, e.g., [12, 132, 108]. As with feature-based structure from motion algorithms, translation and structure can be estimated only up to scale if no additional metric information is available (cf. 2.2.2).

Usually, the pose Jacobian required for pixel-based methods is computed analytically. The presence of a textured three-dimensional model of the object allows, however, to determine the Jacobian using techniques from computer graphics [100]. Other approaches rely as well on the numerical derivation of a Jacobian, whereas the aim is to learn a constant first and second order relationship of image intensity and motion. While [40] considers motion in the image plane for linear shape models, the approaches for tracking in three-dimensional space rely on the existence of primitive shapes such as ellipsoids [29] and planes [75]. Note again that neither the Jacobian nor the Hessian are constant for non-planar surfaces under arbitrary poses.

Appearance Descriptor Methods

While feature-based methods and appearance-based methods rely on a set of object samples, the class of appearance-descriptor-based approaches condense a single view-dependent or view-independent property extracted from the image in a single vector. These properties are typically related to either the area or the contour of the appearance. The integration of local properties over the image or over the surface of the object yields descriptors such as colour histograms [104, 105], moments [111, 136, 137] or Fourier descriptors [3].

In practise, histogram-based approaches have been employed only for 2-

[2] also known as aperture problem

DoF[3] or 3-DoF tracking in the image plane due to the lack in sensitivity to out-of-plane rotations. In contrast to histogram-based approaches, the extraction of Fourier descriptors from contours or the extraction of moments from contours, point-sets, or textured areas allow for the motion estimation in full 6 DoF. However, these approaches require prior object-background segmentation, and the subsequent motion estimation process proves to be sensitive to such a segmentation step, especially for higher-order moments. Moreover, any partial occlusion in the perceived shape can heavily affect the reliability of these methods.

Hybrid Methods

Rarely, different cues have been used simultaneously to estimate object motion. Refer to [113] for an example on the combination of texture and contour information.

2.2.3 Interpretation Domain

Motion estimation algorithms rely explicitly or implicitly on a particular object model. This model primarily links variation in pose to variations in the measurements. Thus, a geometric, three-dimensional representation of the object is not necessary unless the measurements are of the same three-dimensional type. Therefore, a three-dimensional model might only represent a convenient structure to attain a particular goal.

In the following, existing models are analysed with respect to their assumption on the shape and their employed representation. Shape is the object property that is assumed, at least implicitly, for a specific motion estimation approach. Representation, instead, is the explicit requirement of the algorithm and, possibly, also of the approach.

Object Shape

Shapes can be attributed to one of two classes, namely primitive surfaces or free-form surfaces. The former class consists of planes, spheres, cylinders, ellipsoids etc., while the latter consists of arbitrary, unconstrained shapes. Within the class of primitive surfaces, planar shapes are the most popular, beginning with affine motion models [91, 61, 62, 73], which neglect perspective distortions, planes [111, 45, 75, 42, 136, 137, 87, 107], and homographies [93, 7, 23, 14]. Other primitive shapes, such as spheres [30], cylinders [30, 29], and ellipsoids [10, 29] are used as well.

In contrast to primitive surfaces, no compact parametric description exists for arbitrary, free-form surfaces and therefore the change in appearance cannot be established analytically in closed form. The majority of approaches assume rigid free-form surfaces [130, 12, 65, 46, 86, 60, 94, 143, 24, 146, 79, 80, 9, 85, 132, 18, 148, 95, 49, 100, 131, 50, 63, 38], whereas non-rigid, linear deformations are adopted specially for face tracking [21, 101, 40, 7, 41, 116, 95].

[3] degrees of freedom

Object Representation

The concept of representation is closely related to the concept of shape, and denotes the information used by the algorithm to express the shape of the object. The representation is determined first and foremost by the measurement domain.

Appearance descriptor approaches are typically not based on a three-dimensional representation of the object. Instead, the object is represented by the moments perceived at a particular view [3, 111, 136, 137]. Similarly, view-based methods do not rely on a three-dimensional object representation, neither. In this case, images from the different view-points are stored [145].

Other approaches explicitly model the three-dimensional geometry of the object, with the exception of three-dimensional motion estimation from the fundamental or essential matrix, e.g., [130]. Appearance-based motion estimation for planar shapes, for instance, stores the information either as image patches [45, 75, 14] or as 2-d image coordinates together with the corresponding intensity values [7, 23, 42]. Free-form surfaces are represented either by a composition of textured planes [94, 143, 24], by a set of textured 3-d object points [65, 132, 46, 21, 86, 101], by a set of un-textured 3-d object points [85, 60], by a textured wire-frame model [79, 80, 29, 18, 116, 109, 148, 49, 100], by an untextured wire-frame model [9], or by a 3-d contour model [146, 131, 50, 63, 38]. Also planar contour or key-point models exist [111, 136, 137, 87, 107], as well as combinations of keypoints and object contours [82].

2.3 Handling of Illumination and Occlusion

For the task of motion estimation, images are analysed for pose dependent attributes. The perceived image however depends not only on the pose of the object or scene relative to the camera, but is a superposition of effects attributed to illumination, surface reflectivity, view direction as well as occlusion. Typically, only the effects related to the view direction are evaluated for motion estimation, while the other components represent incidental, negative effects.

In literature, several strategies have been devised to handle these negative effects. In detail, the approaches for pose estimation under changes in illumination and/or occlusion can be attributed to one of four types:

- estimation of motion from information invariant to illumination,
- concurrent estimation of motion and illumination/occlusion,
- robust estimation of motion from information affected by illumination/occlusion,
- estimation of motion with constraints on illumination/occlusion.

Approaches of the first kind process information that is invariant to illumination. In contrast, approaches of the second type explicitly model the acquired information to be function of motion, illumination, and occlusion. Approaches of the third kind do not explicitly model the effects of illumination and/or

occlusion. Instead, robust methods are employed with the ability to handle reasonable alterations of illumination and/or occlusion. Hence, estimation is not totally insensitive to these effects. The latter approaches are not robust to changes in illumination and/or occlusion. Therefore, correct estimations are only guaranteed, if certain constraints on illumination and occlusion are met.

2.3.1 Estimation Invariant to Illumination

Generally, no method exists that is completely invariant to changes in illumination. Appearance invariance is usually obtained by separating the method from illumination dependent pre-processing steps.

Accordingly, appearance-descriptor methods based on area moments or contour moments [3, 111, 136] rely on a prior correct image segmentation. This pre-processing step is per se a difficult task for inhomogeneous objects.

Similarly, feature extraction is expected to be in large extent invariant to illumination changes. While contour features like the ones used in [131, 63, 126, 87, 107], point features as employed in [130, 85, 93, 60], and a combination of both as used in [84, 82] are preserved under illumination changes, the detection of these features is usually not invariant to illumination effects. In particular, the extraction and matching of image patches is sensitive to illumination effects.

Appearance-based motion estimation on the other hand is generally not invariant to illumination. Here, surface irradiance may vary over time as the object moves. In addition, non-Lambertian reflections cause the radiance perceived form the surface to depend also on the view-point. The latter effects are minimised by correlating the intensity profiles of narrow baseline stereo images [42, 114] .

2.3.2 Concurrent Estimation of Motion and Illumination

Modelling variations attributed to the simultaneous change in pose *and* illumination has achieved significant attention in appearance-based pose estimation and recognition. Here, the effects of illumination are compensated by explicitly or implicitly computing the parameters of illumination simultaneously to the parameters of motion.

A simple method is given by the global adjustment of brightness and contrast [91, 109]. Its suitability is however limited to planar patches or moderate illumination changes. Alternatively, a complex model of surface reflectivity and illumination can be taken into account. In practice however, only a reduced set of parameters can be estimated [18].

The determination of an illumination subspace [13] combines expressive power with moderate computational requirements. It shows high popularity for motion estimation in the image plane [61, 62, 40, 41, 95] and in three-dimensional space [12, 29, 18, 116, 95, 148, 49, 24, 35, 101]. Minimally, a set of three illumination bases is required to model the light effects. However, the dimensionality of the subspace grows roughly to amount of non-parallel surface patches when self-shadowing is additionally taken into account. In the case of tracking of facial expressions with active appearance models [40], the sub-

space is used to jointly estimate illumination effects and appearance changes not attributed to motion.

2.3.3 Robust Estimation of Motion

Motion estimation is possibly affected by perturbations at all processing levels, starting from scene formation over image acquisition to image processing. The first source of error, however, is not the scene itself but rather the inconsistency between the assumptions of scene formation and the reality. Erroneously, the acquisition process is usually considered the main source of noise. Early computer vision systems tended to introduce significant level of noise, but nowadays this factor has been greatly reduced by the manufacturers of camera systems. The last source of perturbations, i.e., the error introduced by image processing, can be attributed again to the discrepancy between real image formation and the assumed image formation.

Generally, an estimation can be either biased or unbiased depending on whether the perturbations match the model assumptions about the error. Noise at the level of the acquisition process is typically modelled as additive and white Gaussian. This model assumption is usually not appropriate for other sources of perturbations such as occlusion or illumination changes. Only those processes are considered robust in the statistical sense that produce an unbiased estimate despite of non-normally distributed measurements errors.

The design of specific methods for robust motion estimation depends strongly on the underlying measurement domain (see section 2.2.2). For instance, reasonable robustness is achieved for descriptor-based motion estimation with histograms as appearance descriptors by varying the bin size according to the expected level of perturbations.

In contrast, appearance-based motion estimation allows for a wider range of approaches. Special attention is paid to the matching criteria between two patches. Normalised cross correlation [143] is adopted, as well as mutual information [144, 109, 77] and local intensity ordering criteria [54, 86]. These methods aim to primarily compensate for illumination effects. Given a sequence of chronological ordered images, slow illumination changes can also be compensated by estimating motion from frame to frame [79, 11, 80, 65, 132, 108]. The related tendency to drift can be eliminated combining a model update with tracking of the original model [96]. For appearance-based estimation, statistically robust methods such as well known m-estimators [98] are applied [116, 113].

In the case of feature-based motion estimation, perturbations are attributed to the position of the extracted features. Well known approaches are RANSAC methods, e.g., [86], and parameter clustering methods, e.g.,[4], which search a subset of features that is consistent with the model. High popularity is observed for m-estimator methods, which neglect the measurements that do not cope with a common variance [146, 50, 39, 143, 51, 38, 113].

2.3.4 Estimation with Constrained Illumination and Occlusion

A pragmatic approach to deal with illumination changes and occlusion is to impose certain assumptions on the scene that constrain the perturbations to a regime that can be handled by motion estimation algorithms [115, 17, 145, 45, 30, 75, 23, 126, 34, 94, 14, 15, 21, 73, 100].

The assumptions typically address surface reflectivity and visibility. Variablity of irradiance is usually decoupled from the viewer position by assuming perfectly diffusive (Lambertian) surfaces, a common legitimate approximation. Likewise, a visibility constraint is also valid as long as it reflects reality.

2.4 Pose Prediction

The task of image-based pose or motion estimation is usually formulated as an optimisation problem. In practice, the corresponding objective function shows to be highly non-linear and non-convex, and therefore the optimisation process is likely to get stuck in local extrema representing alternative solutions. The ambiguities of solutions can be alleviated by considering previous images of the image sequence. That is, future poses are predicted with certain assumptions on the motion dynamics and the history of past estimations. Thus, the task of global pose estimation based on highly non-linear objective functions changes into a tracking problem with local optimisation.

However, in the case of an observer or an object moving autonomously, the pose parameters cannot be reliably predicted even for the very next frame, because the energy introduced in the motion prevails over the inertia of the masses. Therefore, the simplest prediction is to consider the pose of the camera at the previous frame, e.g., [130, 49]. This behaviour is assumed to be default for the majority of tracking algorithms that do not explicitly mention this problem. The basic model is slightly extended by many applications so as to consider not only the pose at the previous frame, but also the object velocity [45, 73, 104, 105].

A common framework for combining an observation model with a motion model is given by the Markov-Chain models. While Kalman filtering approaches follow a single hypothesis based on a Gaussian error model [130, 45, 87, 107], the more general Sequential Markov-Chain Monte-Carlo methods are able to consider multiple hypothesis at once, and without requiring necessarily a Gaussian error model [70].

2.5 Discussion

The above sections outline the variety of approaches coping with the problems of pose estimation, handling of variations in illumination, pose prediction, and visual servoing. In the following, their suitability for the overall problem of tracking and grasping of non-cooperative objects for the purpose of human-robot interaction is assessed, by considering the four categories in a top-down manner.

2.5.1 Visual Servoing

The most prominent distinction of visual servoing is made with respect to the task space, that is the domain of control. In image-based visual servoing the task consists in minimising distances of a feature-vector defined in the image-domain. Instead, position-based visual servoing specifies the task for the robot end-effector in the Cartesian domain determining the poses that the end-effector shall approach.

Originally, image-based visual servoing methods excelled in the presence of a rough calibration of the camera and robotic system and in the lack of a priori object models. As in any other robotic task however, the robot-Jacobian, which maps motion in the camera frame to joint motion, is determined off-line accurately. Moreover, the image-Jacobian, which links motion in the image plane to motion of the camera frame, typically needs additional spatial information such as the average distance of the object to the camera in order to be estimated correctly. The lack of a priori object models tempts to assume that no object constraints are employed. However, the object is implicitly assumed flat or almost flat in order to obtain an analytical expression of the image-Jacobian, whose evaluation is needed by the control algorithm.

With respect to the trajectories of the robot end-effector, position-based visual servoing allows to follow a straight line in Cartesian space to reach the desired position. Image-based visual servoing, on the other hand, controls the end-effector such that the object features detected in the image follow a straight line toward the desired feature positions. Thus, image-based visual servoing implicitly keeps the object in the visible area but unpredictable Cartesian trajectories may arise.

Position-based visual servoing exhibits two important advantages. First, the robot can be commanded to approach any pose relative to the object, provided that the object is visible at the final configuration. In contrast to image-based visual servoing, the view to the object at the target position does not have to be taught a priori. Second, the position interface allows to easily combine the goal with higher level Cartesian plans. Thus, once on-line path planning becomes feasible, the robot movements can be flexibly adapted to external events such as collision avoidance guaranteeing a globally optimal path at any time instant.

On overall, position-based visual servoing is hereafter preferred to image-based servoing. The often cited advantages of image-based visual servoing fade in the presence of accurate calibrations and implicit model assumptions. Moreover, position-based servoing easily integrates with path planners and allows for arbitrary target positions without additional training efforts.

The decision does affect neither the choice of the camera to robot configuration nor the choice for end-effector open-loop or end-effector close-loop control. This decisions are postponed to the Chapter 6.

2.5.2 Pose Estimation

The commitment to position-based visual servoing excludes tracking methods with planar motion models and thus focuses on tracking in three-dimensional

Cartesian space. Approaches based on orthographic projection, weak perspective projection, or homography projection impose serious constraints on the object and/or the camera-to-object configuration and are therefore also excluded from further considerations. The full perspective projection, on the other hand, does not only allow to track objects accurately in appropriate distances to the camera but also in close-up configurations, where the effects of perspective projection become dominant.

Furthermore, the choice of position-based visual servoing affects the measurement domain. Descriptor-based methods are closely connected to image-based visual servoing approaches and, therefore, are not suited. Methods based on local appearance cues, on the other hand, have been successfully employed for position-based visual servoing.

In the following however, the complementary domain is explored comprising methods based on *global appearance* representation. These methods are not affected by the texture of objects nor by a cluttered background. Up to now, appearance-based approaches have been proposed that rely on representations severely restricting the object shape or the camera-to-object configuration. This thesis explores methods that assume general but known *free-form* objects and, at the same time, impose no restrictions on the perspective camera model. Furthermore, the internal object representation is not restricted to a composition of locally planar patches, and hence no linearisation of the object shape is employed at this stage.

2.5.3 Handling of Illumination

Handling of variations in illumination is still a hard problem in computer vision. All 2-d image-based pose estimation methods, regardless of their measurement domain, access pixel values at the first processing stages and, therefore, are subject to variations in illumination. It would be highly desirable to base pose estimation on illumination invariant quantities but, up to now, no description exists for general types of surfaces and surface reflectances. Hence, this thesis explores existing solutions of illumination compensation and adopts them to the problem under examination. The methods considered comprise robust estimation of motion from information affected by illumination/occlusion, concurrent estimation of motion and illumination, and estimation of motion with constraints on illumination/occlusion.

2.5.4 Pose Prediction

Different models of motion dynamics have been proposed relying, in general, on the knowledge about the energy introduced in the system and the inertia of the moving object. In the special case of tracking non-cooperative objects moved by humans, the inertia of the object is considered very low compared to the energy employed by the human. Thus, in the following, no assumption on the object trajectory is introduced other than continuity.

3

Shape-Texture Based Tracking

The image taken from the physical environment depends on the properties of the objects in scene, properties of the light sources and the camera, as well as on the relative positions of all these components. In detail, the objects in the scene are described by their shape and their material properties such as texture and reflection characteristics. Light sources, on the other hand, are distinguished by their power spectrum and direction of radiation. Finally, the cameras can be characterised by their focal length, lens distortion, principal point, and the physical resolution of the sensor chip.

Theoretically, it is possible to build a sophisticated imaging model dependent on the manifold of all the parameters of the environment and the camera. However, the computational power of desktop computer systems is not yet sufficient for photo-realistic rendering of complex scenes for a given parameter set in real-time. The inverse problem of determining these parameters from an image is computationally even much more expensive. Indeed, rendering of many such images may be required for parameter estimation.

The complexity of the problem is generally tackled by the introduction of constraints that reduce the number of free parameters of the environment. First, it is assumed that no interdependence exists between parts of the scene with regard to their appearance. The objects are not obscured by other objects neither in terms of visibility nor in terms of illumination. Accordingly, the object in question can be isolated from the background and their appearance depends solely on the characteristics of the objects themselves, the lighting conditions, and the camera. Second, the objects are assumed to be rigid and therefore shape parameters do not vary over time. Finally, constant and a priori known intrinsic camera parameters represent the third popular constraint.

In the case of moving objects the complexity can be considerably reduced by *tracking* an object. That is, the problem of global pose parameter estimation at each time instance reduces to a local problem starting from an initial estimate of the parameters. Such an initial estimate, called hypothesis, is refined in the

image, and its state is subsequently propagated to a pose hypothesis in the following image.

In the following, the type of camera is restricted to commercially available monochromatic cameras as opposed to 2.5-d cameras such as time-of-flight cameras[1]. Only a single camera is used, which further reduces the cost of the system.

The goal is to consecutively locate an a priori known object in 6 degrees of freedom (DoF) in a video-stream of monocular images in hard real-time. Here, real-time refers to the frame-rate of the camera at a minimum of 25 frames per second. Either the camera, the object or both are moving, while the parameters in question describe the relative pose between the object and the camera.

The majority of 6-DoF pose estimation algorithms are based on the detection of specific features in the image such as edges and corners in the brightness image. Projective correspondences are established between the feature positions and the geometric model. Yet, these shape-cues do not exist for smooth or general free-form surfaces.

Alternatively, the object texture can be used for the task of 6-DoF pose estimation. In contrast to shading, texture is an illumination-independent property of the surface describing the portion of the reflected radiation to the received radiation. Apart from this reflection property and apart from the illumination conditions, the object appearance is determined by the object pose. In principle, the appearance given by the perspective projection of the surface texture can be inversely used to estimate this 6-DoF pose.

The challenge herein consists in efficiently matching either appearance to pose or appearance variation to pose variation. View-based approaches matching appearance to pose are typically not adequate for real-time pose estimation due to their computational complexity. On the other hand, tracking methods matching variations in appearance to variations in pose so far have been established only for primitive surfaces or simplified projective models.

Hereafter, appearance-based approaches are developed that estimate the pose of an a priori known rigid-body in real-time. A general object representation allows to model arbitrary surfaces by samples of combined shape and texture information. This object model is used within the so called likelihood function to determine the evidence for a specific pose (section 3.1). Two approaches are devised to maximise the likelihood and thus to estimate the object pose. The first approach updates a single pose hypothesis by mapping appearance variation to pose variation (section 3.2). The second approach, instead, keeps track of multiple pose hypotheses by matching the current appearance with the appearances associated to the hypotheses (section 3.3). Both approaches are evaluated as to their functionality and convergence characteristics (section 3.4).

[1] e.g. from PMD Technologies (http://www.pmdtec.com), or Mesa Imaging (http://www.mesa-imaging.ch)

3.1 Shape-Texture Based Maximum Likelihood Estimation

As a first simplification of the image formation model the object appearance is determined by the surface texture and the pose of the surface with respect to the camera. In order to inversely estimate the pose from the object appearance an objective function is set up that reflects the evidence of the current appearance for a specific pose. This function is optimised with respect to the pose parameters to yield the estimate of the object pose. Typically, the objective function is described by three aspects. First, the objective relies on a specific meaning of the parameters. Second, the objective function incorporates a model linking the parameters to the measurements, that is, to the appearance. Last but not least, the objective function defines how the parameters agree with the measurements for the employed model.

Overall, these issues are embedded in the following in the theory of maximum likelihood estimation. This widely accepted stochastical framework allows to clearly show the objective of the task and to disclose the employed assumptions. Moreover, it allows to address the problem with stochastical methods and methods from optimisation theory, such as efficient second order minimisation for single hypothesis tracking (section 3.2) or Markov-Chain Monte-Carlo methods for tracking many hypotheses at once (section 3.3).

The following subsections start with a definition of the terms of motion and pose outlining the different possibilities for the determination of a reference frame. After a brief description of maximum likelihood estimation, the central idea of likelihood based on a combined model of shape and texture is presented.

3.1.1 Rigid-body Motion

In contrast to the motion of deformable surfaces, the notions of motion and pose refer hereafter to the coordinate transformation of object points given in a specific frame to their coordinates relative to another frame. Thus, the estimation of motion and pose relies on the determination of a particular frame, called reference frame, relatively to which the coordinate transformation will be expressed. In principle, the reference frame can be arbitrarily set. However, the placement significantly affects the mathematical structure of the problem formulation easing or complicating the derivation of efficient pose estimation algorithms.

Generally, motion can be classified into object motion and ego motion depending on whether the camera or the object is considered stationary. In the case of object motion, the transformations are estimated between consecutive locations and orientations of the object frame. For ego motion instead, the transformations are estimated relatively to the previous location of the camera frame. Analogously to the task of motion, estimation also pose estimation problems can be associated to either object pose or camera pose estimation.

Theoretically, all of these four types are interchangeable whereas pose and motion estimation are linked via an additional frame, that is the object frame. Thus, successive motion estimates can be composed to a single pose estimate,

knowing the initial relative pose between camera and object.

Hereafter, two possible combinations of transformations between object and camera frames are considered for the task of pose estimation. Generally, these approaches are independent of the representation of transformations between frames. The commitment to a specific representation however, allows to estimate the computational costs associated with the transformation. Thus, a special representation is shortly outlined first.

Representation of Rigid-body Transformation

Several representations of rigid body transformations have been proposed in the literature, such as dual quaternions, Plucker coordinates, or transformations with separate translation and Eulerian or Cartesian rotation. For the latter type, let the motion parameters $\boldsymbol{\mu} = (\mu_\alpha, \mu_\beta, \mu_\gamma, \mu_\mathrm{x}, \mu_\mathrm{y}, \mu_\mathrm{z}) \in \mathbb{R}^6$ define the transformation

$$m(\mathbf{x}, \boldsymbol{\mu}) = R(\mu_\alpha, \mu_\beta, \mu_\gamma)\,\mathbf{x} + t(\mu_\mathrm{x}, \mu_\mathrm{y}, \mu_\mathrm{z}) \tag{3.1}$$

of a point $\mathbf{x} \in X$ from one coordinate frame to another coordinate frame. Here, $R(\mu_\alpha, \mu_\beta, \mu_\gamma) \in \mathrm{SO}(3)$ denotes the rotation and $t(\mu_\mathrm{x}, \mu_\mathrm{y}, \mu_\mathrm{z})$ specifies the translation in three dimensions associated with the pose $\boldsymbol{\mu}$.

Many representations exist for the rotation in SO(3), ranging from quaternions, angle-axis representations to Euler angles. The latter representation is very popular and widely used in robotic systems because of to its comprehensibility and its simplicity. Among the possible combinations of Eulerian rotation axes, the rotation can be specified around the moving Cartesian axes x, y and z,

$$R(\mu_\alpha, \mu_\beta, \mu_\gamma) = \tag{3.2}$$
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\mu_\alpha) & -\sin(\mu_\alpha) \\ 0 & \sin(\mu_\alpha) & \cos(\mu_\alpha) \end{pmatrix} \begin{pmatrix} \cos(\mu_\beta) & 0 & \sin(\mu_\beta) \\ 0 & 1 & 0 \\ -\sin(\mu_\beta) & 0 & \cos(\mu_\beta) \end{pmatrix} \begin{pmatrix} \cos(\mu_\gamma) & -\sin(\mu_\gamma) & 0 \\ \sin(\mu_\gamma) & \cos(\mu_\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Generally, any other parametrisation of rotation can be adopted. However, the choice has to be conscious about the singularities in representation. It has to be ensured, that the motion has not to be estimated close to these singularities. Last but not least, the translations are usually defined by its vector components,

$$t(\mu_\mathrm{x}, \mu_\mathrm{y}, \mu_\mathrm{z}) = \begin{pmatrix} \mu_\mathrm{x} \\ \mu_\mathrm{y} \\ \mu_\mathrm{z} \end{pmatrix}, \tag{3.3}$$

which are performed according to equation 3.1 subsequently to the rotation.

Motion w.r.t. a Stationary Reference Frame

For the task of pose estimation, the rigid-body transformation between an object frame and the camera frame is eventually determined. Generally, the problem of global estimation of this transformation is simplified to a local problem

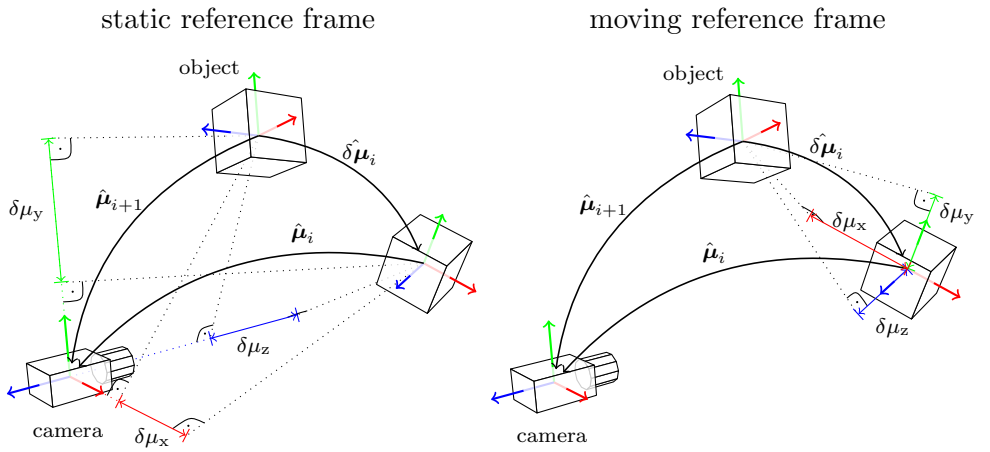

Figure 3.1: Stationary reference frame (left) and moving reference frame (right).

using a prior pose hypothesis close to the true pose. The rigid-body transformation between the object frame given by the hypothesis and the object frame at the true pose can be expressed relative to either of the frames or relative to the stationary camera frame.

Tracking approaches generate pose hypothesis $\hat{\boldsymbol{\mu}}_i$ at any, possibly discrete, time instant i. Thus, if the pose increment $\delta\boldsymbol{\mu}_i$ between the time instant i and $i+1$ refers to the same stationary camera frame then the rule

$$\hat{\boldsymbol{\mu}}_{i+1} = \hat{\boldsymbol{\mu}}_i + \delta\boldsymbol{\mu}_i \tag{3.4}$$

updates the pose estimate at time i to the next time instant $i+1$. Refer to figure 3.1 for an illustration. The corresponding rigid-body transformation is determined according to

$$m\big(\mathbf{x}, \hat{\boldsymbol{\mu}}_{i+1}\big) = m(\mathbf{x}, \hat{\boldsymbol{\mu}}_i + \delta\boldsymbol{\mu}_i) \ . \tag{3.5}$$

Baker and Matthews [8] refer to these approaches using the stationary reference frame as *additive* methods.

Motion w.r.t. a Moving Reference Frame

Alternatively, the rigid-body transformation between the object frame at the pose hypothesis and the true object frame can be expressed w.r.t. the former frame. This object related frame is continously moving relative to the stationary camera frame.

That is, the approach relies again on a hypothesis $\hat{\boldsymbol{\mu}}_i$ of the object pose at time instant i. If the motion between the time instants i and $i+1$ is modelled as the nested transformation of initial pose $\hat{\boldsymbol{\mu}}_i$ and the pose variation $\delta\boldsymbol{\mu}_i$ then the resulting rigid-body transformation reads

$$m\big(\mathbf{x}, \hat{\boldsymbol{\mu}}_{i+1}\big) = m(m(\mathbf{x}, \delta\boldsymbol{\mu}_i), \hat{\boldsymbol{\mu}}_i) \ . \tag{3.6}$$

Accordingly, the pose estimate at time instant i is updated to the next instant $i+1$ through

$$\hat{\boldsymbol{\mu}}_{i+1} = \hat{\boldsymbol{\mu}}_i \circ \delta\boldsymbol{\mu}_i \ , \tag{3.7}$$

where the operator $\circ$ composes the motion parameters in line with equation 3.6. See figure 3.1 for an illustration of the concept. In the literature, tracking methods based on this definition of a moving reference frame are referred to as *compositional* approaches [8].

3.1.2 Maximum Likelihood Estimation

In statistics, maximum likelihood estimation (MLE) is an established method for the estimation of the parameters of a probability density function from given data samples. The theory dates back to R. A. Fisher at the beginning of the 20th century [53] and gained more and more popularity through the years.

In maximum likelihood estimation the term likelihood is closely linked to the probability of joint occurrence of a set of data samples $\mathrm{I} = (I_1, I_1, \ldots, I_N)$ given some (multi-dimensional) parameter $\boldsymbol{\mu}$. The conditional probability of the set of samples for given model parameters is expressed by the probability density function (pdf)

$$p(\mathrm{I}|\boldsymbol{\mu}) = p(I_1, I_2, \ldots, I_N|\boldsymbol{\mu}) \ . \tag{3.8}$$

The so-called likelihood function binds the set of data samples to known values and reduces the above definition consequently to

$$L(\boldsymbol{\mu}) = p(\mathrm{I}|\boldsymbol{\mu}) \ , \tag{3.9}$$

which reflects the probability density of the constant set of samples for a given parameter $\boldsymbol{\mu}$. The objective of a maximum likelihood estimator is to find the pose parameter $\hat{\boldsymbol{\mu}}^\star$ that maximises the probability density of the given data samples, that is

$$\hat{\boldsymbol{\mu}}^\star = \arg\max_{\boldsymbol{\mu}} L(\boldsymbol{\mu}) \ . \tag{3.10}$$

Given the correct probability model, the maximum likelihood estimator has optimal asymptotic ($N \rightarrow \inf$) properties (cf. [43])

- the MLE is asymptotically unbiased,
- the MLE is statistically consistent and
- the MLE is asymptotically efficient.

The latter asserts that the MLE reaches asymptotically the minimal expected squared error achievable with any estimator.

3.1.3 Shape-Texture Based Likelihood

The likelihood is adopted in this work for the evaluation of the joint occurrence of object pose parameters and image measurements. In contrast to feature-based pose estimation, the measurements are constituted neither by the locations of sparse, salient features in the image nor by the position of shape features of the surface. Instead, pixel measurements are taken on image locations not

restricted to salient brightness changes. The set of pixel measurements describe the appearance of the object.

Assume $\mathrm{I} = (I_1, I_1, \ldots, I_N)$ to be a N-dimensional vector of measured intensity values of the surface texture that are conditionally independent given the object pose. Thus, the probability density of the measurement I, given the 6-dimensional pose vector $\boldsymbol{\mu}$, is specified by the product of the pixel individual conditional probabilities

$$p(\mathrm{I}|\boldsymbol{\mu}) = \prod_{i=1}^{N} p(I_i|\boldsymbol{\mu}) \ . \tag{3.11}$$

The conditional probabilities are determined by setting up an appearance model and comparing the appearance linked to a specific pose with the current measurements. In contrast to view-based pose or object recognition, no image database is set up correlating appearances to viewpoints. Instead, a functional model for the local object *radiance* towards the camera is built. Note the difference to a model for the local *irradiance* received from the object: the former projects the image of the object back to its surface while the latter evaluates the image captured from the object. This difference is analogous to the one between the texture-based and image-based approaches outlined in section 2.2.2.

Unlike approaches based on the surface texture expressing the illumination-independent albedo of the surface, the functional model for the local surface radiance reflects illumination-dependent quantities. This model allows to directly map measured image intensity values to the surface.

The functional model is composed of the object shape description, the motion model, the model of image projection, and the image of the object. The object shape, which is assumed a priori known, is represented in terms of an unordered set of sample points

$$X = \{\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_N\} \subset \mathbb{R}^3 \tag{3.12}$$

in three-dimensional Euclidean space. This description does not impose any constraints other than visibility on the surface, as opposed to the shape constraints imposed by any parametric surface description, e.g. planes, cylinders, and NURBS[2] surfaces.

The quantity of light emitted from the shape X towards the viewer is determined sampling an image for a specific pose of the shape relative to the camera. Here, this pose is represented by the rigid-body transformation of equation 3.1, whereas in theory other representations are possible.

The model points transformed into the camera frame are linked to the corresponding image coordinates by the model of projection. Up to now, efficient solutions for motion estimation adopted either an orthographic or a weak perspective projection model. These models are an approximation of full perspective projection but allow for the estimation problem to become linear and to

[2] Non-uniform rational b-spline

be solved with linear methods. Instead, the surface model points $\mathbf{x} \in X$ are non-linearly mapped here to the image under the full perspective projection

$$p(\mathbf{x}) = \left(\frac{\mathbf{k}_1^{\mathrm{T}} \cdot \mathbf{x}}{\mathbf{k}_3^{\mathrm{T}} \cdot \mathbf{x}}, \frac{\mathbf{k}_2^{\mathrm{T}} \cdot \mathbf{x}}{\mathbf{k}_3^{\mathrm{T}} \cdot \mathbf{x}} \right)^{\mathrm{T}} , \tag{3.13}$$

which accounts for perspective distortions for any object and object pose. The intrinsic camera parameters $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3$ are known a priori and are generally defined by

$$\begin{pmatrix} \mathbf{k}_1^{\mathrm{T}} \\ \mathbf{k}_2^{\mathrm{T}} \\ \mathbf{k}_3^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & u_1 \\ 0 & 0 & 1 \end{pmatrix} . \tag{3.14}$$

Here, α and β denote the scale parameters in horizontal and vertical directions, respectively. The parameter γ represents skew, while the optical centre in the image is given by (u_0, u_1).

Let $I(\mathbf{v})$ be the intensity value in the *current* image at position $\mathbf{v} \in \mathbb{R}^2$. Now, the radiances $\mathrm{I} = (I_{\mathbf{x}_1}, I_{\mathbf{x}_2}, \ldots, I_{\mathbf{x}_N})$ of the surface X are related to the image I and the pose $\boldsymbol{\mu}$ by

$$I_{\mathbf{x}}(\boldsymbol{\mu}) = I(p(m(\mathbf{x}, \boldsymbol{\mu}))) , \quad \mathbf{x} \in X . \tag{3.15}$$

Likewise, the radiances ${}^0\mathrm{I} = ({}^0I_{\mathbf{x}_1}, {}^0I_{\mathbf{x}_2}, \ldots, {}^0I_{\mathbf{x}_N})$ of the surface in a *reference* image ${}^0I$ and the reference pose ${}^0\boldsymbol{\mu}$ are defined by

$${}^0I_{\mathbf{x}}({}^0\boldsymbol{\mu}) = {}^0I\big(p\big(m\big(\mathbf{x}, {}^0\boldsymbol{\mu}\big)\big)\big) , \quad \mathbf{x} \in X . \tag{3.16}$$

The radiance values do not change, if either the surface has Lambertian reflectance properties and the illumination remains fixed relative to the surface or if the surface is homogeneously illuminated. In this chapter, the radiances are assumed constant under pose variation and consistently I and ${}^0\mathrm{I}$ are referred to as the current and reference *texture*, respectively.

Generally, handling of the likelihood function can be greatly simplified by assuming normally distributed measurement errors. Thus, the pdf of conditionally independent observations $I_{\mathbf{x}}$ given the pose parameter $\boldsymbol{\mu}$ is finally defined as

$$p(\mathrm{I}|\boldsymbol{\mu}) = \prod_{\mathbf{x} \in X} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(- \frac{\big(I_{\mathbf{x}}(\boldsymbol{\mu}) - {}^0I_{\mathbf{x}}({}^0\boldsymbol{\mu})\big)^2}{2\sigma^2} \right) . \tag{3.17}$$

The corresponding likelihood function is hence maximised when the radiances sampled in the current image for the pose $\boldsymbol{\mu}$ resemble the reference texture. The problem is equivalent to the maximisation of the log-likelihood function

$$\ln L(\boldsymbol{\mu}) = -|X| \ln \sqrt{2\pi}\sigma - \frac{1}{2\sigma^2} \sum_{\mathbf{x} \in X} \big(I_{\mathbf{x}}(\boldsymbol{\mu}) - {}^0I_{\mathbf{x}}({}^0\boldsymbol{\mu})\big)^2 , \tag{3.18}$$

which is essentially the sum of squared differences (SSD) between the intensity values in the current and reference views.

3.2 Single-Hypothesis Tracking

Appearance-based pose estimation in 6 DoF can be accomplished by propagating and refining a single pose hypothesis from frame to frame. Despite the reduced complexity, considerable computational power is still needed for this local pose estimation task. The unpredictability of object motion and the time elapsed between two observations dictate the size of the pose space to be explored and respectively affect the computational requirements (cf. chapter 1).

Under the assumption of uni-modality, the shape-texture based likelihood function of subsection 3.1.3 can be maximised sequentially for the given pose hypothesis. Typical sequential optimisation methods are first-order and second-order optimisation techniques, whereas both rely on the computation of the Jacobian and respectively of the Hessian of an objective function. Accurate and quick convergence is obtained for an analytical representation of the Jacobian (and Hessian) as opposed to the numerical computation of the Jacobian. In the case of appearance-based pose estimation, the calculated Jacobian represents the first-order relation between variation of appearance and variation in pose.

In contrast to non time-critical optimisation problems, the speed of convergence plays an important role for pose tracking. The speed of convergence determines the maximal object velocities supported by the approach. The requirement to raise computational efficiency can be met by computing parts of the current motion Jacobian off-line. Implicitly, the question is raised how the current, non-constant motion Jacobian can be related to past Jacobians, especially to the Jacobian of the reference view to precompute part of the Jacobian a priori.

In the following subsection 3.2.1, sequential second-order minimisation by means of the well known Gauss-Newton algorithm is described first. The computational complexity involved in the computation of the Jacobian is outlined motivating the development of approaches for the efficient prediction of the Jacobian. The link between past and present motion Jacobians is established in subsection 3.2.2 by the image-constancy assumption specialised to the rigid-motion of arbitrary free-form surfaces. An efficient analytic formula is derived in subsection 3.2.3 predicting the spatial texture Jacobian for any object pose. In subsection 3.2.4, the spatial texture Jacobian is mapped to the motion Jacobian to allow for efficient tracking with a moving reference frame [121]. Furthermore, the analytic formula allows to determine the conditions for relaxing the image-constancy assumption as described in subsection 3.2.5. Accordingly, the approximation error can be quantified for the assumptions on the constancy of the texture Jacobian. Hence, the relaxation leads to a constant motion Jacobian for moving reference frames, which further increases the computational efficiency.

3.2.1 Sequential Maximisation of Likelihood

Obviously, the likelihood for a pose is maximised by minimising the negative of the log-likelihood 3.18. Generally, parameter optimisation is defined as the problem of finding the extrema of the cost function, characterised by the zero

crossing of the first derivative of the cost function.

In numerical analysis, various local first-order and second-order minimisation techniques exist, for instance gradient descent methods, Newton-Raphson method, quasi-Newton methods, and the modified Newton method employed in the Levenberg-Marquardt minimisation. The latter is the most common for SSD approaches. Lucas and Kanade [91] introduced such a modified Newton method to the vision community, which proved very valuable for image registration tasks.

The negative log-likelihood function is minimised here with Newton's algorithm and a Gauss-Newton approximation to the Hessian by iteratively solving the linear equation system at the pose estimates $\hat{\boldsymbol{\mu}}$

$$\sum_{\mathbf{x}\in X} \partial_{\boldsymbol{\mu}} I_{\mathbf{x}}(\boldsymbol{\mu})^{\mathrm{T}} \cdot \partial_{\boldsymbol{\mu}} I_{\mathbf{x}}(\boldsymbol{\mu})\Big|_{\boldsymbol{\mu}=\hat{\boldsymbol{\mu}}} \hat{\delta\boldsymbol{\mu}} = \sum_{\mathbf{x}\in X} \partial_{\boldsymbol{\mu}} I_{\mathbf{x}}(\boldsymbol{\mu})^{\mathrm{T}}\Big|_{\boldsymbol{\mu}=\hat{\boldsymbol{\mu}}} \left(I_{\mathbf{x}}(\hat{\boldsymbol{\mu}}) - {}^{0}I_{\mathbf{x}}({}^{0}\boldsymbol{\mu})\right), \tag{3.19}$$

for the pose increment $\hat{\delta\boldsymbol{\mu}}$. Here, $\partial_{\boldsymbol{\mu}}$ denotes the partial derivative with respect to $\boldsymbol{\mu}$.

Newton methods show quadratic convergence to the minimum, which is the major strength of the method. Additionally, this second-order minimisation can be gradually reduced to a first-order or gradient descent method, a strategy realised by the Levenberg-Marquardt algorithm.

Commonly, both first-order and second-order minimisation algorithms rely either on the numerical or analytical derivative of the model function. Computing the first order derivative (Jacobian) numerically requires an evaluation of the model function for every dimension of the parameter space. This approach is not feasible for real-time applications, if a single evaluation of the model function is computationally expensive. In contrast, analytic determination of the Jacobian is computationally less expensive and has moreover the advantage that the closest minimum is reached with higher accuracy.

In order to compute the pose Jacobian $\partial_{\boldsymbol{\mu}} I$ at a specific pose $\hat{\boldsymbol{\mu}}$ analytically, the derivative of the image as well as the derivatives of the projection and object motion have to be computed. The floating point operations involved in the computation of the Jacobian amount to $26N$ multiplications and $11N$ additions for N 3-d model points as depicted in Table 3.1. The complexity for computing

	$\partial_{\mathbf{u}} I$	$\partial_{\mathbf{x}} p$	$\partial_{\boldsymbol{\mu}} m$	$\partial_{\mathbf{u}} I \cdot \partial_{\mathbf{x}} p$	$\partial_{\mathbf{u}} I \cdot \partial_{\mathbf{x}} p \cdot \partial_{\boldsymbol{\mu}} m$	total
#fp mult.	0 (min)	9	9	5	3	26
#fp add.	2 (min)	1	6	2	2	13

Table 3.1: Number of floating point operations employed for the computation of the texture Jacobian for a single 3-d model point.

the image derivatives $\partial_{\mathbf{u}} I$ depends on the sub-pixel interpolation scheme and numerical differentiation in the discretely sampled image I. Bilinear interpolation and the robust differentiation with the Sobel operator, for instance, require additionally 5 floating point multiplications and 12 additions.

The effective performance eventually depends not only on the number of floating point operations but also on the structure of the program. Computation of the chained derivative for example can hardly be vectorised to benefit from SIMD instructions on modern CPUs. Another important aspect is the memory throughput, which suffers from random access to image pixels. Typically, the projections of the 3-d model points spread all over the image making efficient memory block transfers useless.

All these factors decrease the performance of first-order and higher-order minimisation methods such as the Newton method. A constant frame-rate and limited processing resources constrain the minimisation to few iteration steps. The speed of convergence and robustness are heavily affected, motivating the development of more powerful approaches.

3.2.2 Image-Constancy Assumption in 3-d

In optical flow theory, the so called image-constancy assumption for Lambertian surfaces asserts that consecutively captured images of a scene are congruent under local coordinate transformations, that is

$$ {}^{t}I(\mathbf{u}_{\mathbf{v}}(t)) = \text{const} \tag{3.20} $$

for a given path $\mathbf{u}_{\mathbf{v}}(t) : \mathbb{R} \rightarrow \mathbb{R}^2$ of the pixel coordinate $\mathbf{v}$ in space time continuum. The derivative $\partial_t \mathbf{u}_{\mathbf{v}}(t)$ is called optic flow and is usually given by numerical differentiation. The total derivative of the above equation with respect to t

$$ \partial_t {}^{t}I(\mathbf{u}) \Big|_{\mathbf{u}=\mathbf{u}_{\mathbf{v}}(t)} = -\partial_{\mathbf{u}} {}^{t}I(\mathbf{u}) \Big|_{\mathbf{u}=\mathbf{u}_{\mathbf{v}}(t)} \partial_t \mathbf{u}_{\mathbf{v}}(t) \tag{3.21} $$

represents the standard optic flow equation. In practice, the discrete version of the first-order derivative relates two consecutive camera images. However, equation 3.21 is not suited to express the relation between two images subject to substantial camera or object motion because it is typically based on local spatial and temporal derivatives. Also equation 3.20 is not appropriate for object pose estimation since the optic flow $\partial_t \mathbf{u}_{\mathbf{v}}(t)$ is not constrained by variations of pose.

Here, two images are instead related via surface *and* texture, which leads to a specialisation of the standard image-constancy assumption. Let $I_{\mathbf{x}}(\boldsymbol{\mu})$ and ${}^{0}I_{\mathbf{x}}({}^{0}\boldsymbol{\mu})$ again denote the functions perspectively mapping a three-dimensional object point $\mathbf{x}$ to intensity values of the current image I and reference image ${}^{0}I$ for the respective rigid-body motions $\boldsymbol{\mu}$ and ${}^{0}\boldsymbol{\mu}$. The intrinsic camera parameters including possible distortion parameters are assumed known. Then, the standard image-constancy assumption is specialised imposing simultaneously surface and texture constraints at any point $\mathbf{x}$ of the *uncountable* set of scene surface points $\mathcal{X} \supset X$. The resulting function

$$ I_{\mathbf{x}}(\boldsymbol{\mu}) = {}^{0}I_{\mathbf{x}}({}^{0}\boldsymbol{\mu}) \;, \quad \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^3 \tag{3.22} $$

is referred to as image-constancy assumption in 3-d. In contrast to the standard constancy assumption, equality is not postulated for the image pixels but for the surface texture.

3.2.3 Analytic Prediction of the Spatial Texture Jacobian

Respectively to the definition of the pose Jacobian $\partial_{\boldsymbol{\mu}} I_{\mathbf{x}}(\boldsymbol{\mu})$ for a surface point $\mathbf{x} \in \mathcal{X}$, the partial derivative of the texture function with respect to the three-dimensional object coordinates

$$\nabla I_{\mathbf{x}}(\boldsymbol{\mu}) = \left(\partial_{x_1} I_{\mathbf{x}}(\boldsymbol{\mu}) \, , \partial_{x_2} I_{\mathbf{x}}(\boldsymbol{\mu}) \, , \partial_{x_3} I_{\mathbf{x}}(\boldsymbol{\mu})\right) \ , \quad \mathbf{x} = (x_1, x_2, x_3) \tag{3.23}$$

is called in the following spatial texture Jacobian. The definition of the spatial texture Jacobian $\nabla {}^0I_{\mathbf{x}}\left({}^0\boldsymbol{\mu}\right)$ for the reference image ${}^0I$ and reference pose ${}^0\boldsymbol{\mu}$ are obtained analogously.

In order to express the current spatial texture Jacobian in terms of the reference Jacobian, the derivative of the image-constancy assumption with respect to the surface is considered. For this purpose, let first $B_\epsilon(a, A) = \{b \in A | \|a - b\| < \epsilon\}$ with $0 < \epsilon$ denote an ϵ-neighbourhood of a point a in domain A. Then, a continously differentiable local parameterisation

$$s_{\mathbf{x}} : \mathcal{B}_\eta\left(\mathbf{0}, \mathbb{R}^2\right) \rightarrow \mathcal{B}_\zeta(\mathbf{x}, \mathcal{X}) \ , \quad s_{\mathbf{x}}(\mathbf{0}) = \mathbf{x} \ , \quad s_{\mathbf{x}} \in \mathrm{C}^1 \tag{3.24}$$

of the surface $\mathcal{X}$ is specified at $\mathbf{x} \in \mathcal{X}$. This function maps motion on a two-dimensional grid to motion in three-dimensional space. Consequently, the image-constancy assumption reads for parametrised surfaces in 3-d:

$$I_{s_{\mathbf{x}}(\mathbf{u})}(\boldsymbol{\mu}) = {}^0I_{s_{\mathbf{x}}(\mathbf{u})}\left({}^0\boldsymbol{\mu}\right) \ , \quad \mathbf{x} \in \mathcal{X} \ . \tag{3.25}$$

The derivative of the image-constancy assumption with respect to the surface parameters is now given by

$$\nabla I_{\mathbf{x}}(\boldsymbol{\mu}) \cdot \partial_{\mathbf{u}} s_{\mathbf{x}}(\mathbf{u}) = \nabla {}^0I_{\mathbf{x}}\left({}^0\boldsymbol{\mu}\right) \cdot \partial_{\mathbf{u}} s_{\mathbf{x}}(\mathbf{u}) \, , \quad \mathbf{x} \in \mathcal{X}, \tag{3.26}$$

which holds for continuously differentiable images I and ${}^0I$ at any point of the surface. This equation binds parameter variation in $\mathcal{B}_\eta\left(\mathbf{0}, \mathbb{R}^2\right)$ to brightness variation in $\mathbb{R}$. Note that the surface variation spans a three-dimensional plane tangential to the surface at $\mathbf{u}$. The base vectors of the plane are given by the columns of the gradient $\partial_{\mathbf{u}} s_{\mathbf{x}}(\mathbf{u})$, which are hereafter assumed w.l.o.g. orthonormal at $\mathbf{u} = \mathbf{0}$.

Obviously, the image-constancy assumption in 3-d and its derivative are restricted to the surface shell. Hereafter, the equality 3.26 is transformed from variations on the two-dimensional surface patch to variations in three-dimensional space. The objective is achieved through an injective mapping from local three-dimensional variations to lower-dimensional variations on the tangential plane of a surface point. This rank-deficient linear mapping is accomplished by the projector $\mathbb{1} - \mathbf{n}_{\mathbf{x}} \mathbf{n}_{\mathbf{x}}^{\mathrm{T}}$, where $\mathbf{n}_{\mathbf{x}}$ is the surface normal at a point $\mathbf{x} \in \mathcal{X}$. The multiplication of the image-constancy assumption 3.26 with the transpose $\left(\partial_{\mathbf{u}} s_{\mathbf{x}}(\mathbf{u})\right)^{\mathrm{T}}$ leads at $\mathbf{u} = \mathbf{0}$ to the first assertion.

Assertion 1 (Constancy of Spatial Texture Jacobian) *Let $\mathbf{x} \in \mathcal{X}$ and $\mathbf{n}_{\mathbf{x}} \in \mathbb{R}^3, \|\mathbf{n}_{\mathbf{x}}\| = 1$ be a point and the corresponding normal on a continously differentiable rigid surface with a continuously differentiable texture invariant to*

changes of viewpoint. Moreover, let ${}^0I, I : \mathbb{R}^2 \to \mathbb{R}$ *denote two camera images with respective object poses* ${}^0\boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^6$.

The projection of the spatial texture derivative $\nabla I_{\mathbf{x}}(\boldsymbol{\mu})$ *matches the projection of* $\nabla\, {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)$ *on the tangential plane* $\mathbb{1} - \mathbf{n}_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}^{\mathrm{T}}$, *i.e.,*

$$\nabla I_{\mathbf{x}}(\boldsymbol{\mu}) \cdot \big(\mathbb{1} - \mathbf{n}_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}^{\mathrm{T}}\big) = \nabla\, {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big) \cdot \big(\mathbb{1} - \mathbf{n}_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}^{\mathrm{T}}\big) \ .$$

Obviously, the view dependent spatial derivative $\nabla I_{\mathbf{x}}(\boldsymbol{\mu})$ cannot be computed directly from an a priori known reference image ${}^0I$ and reference pose ${}^0\boldsymbol{\mu}$. The rank-deficient, constant mapping $\mathbb{1} - \mathbf{n}_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}^{\mathrm{T}}$ constrains the spatial derivative only up to a linear subspace. Therefore, an additional constraint is needed. Prior to the introduction of this constraint, assertion 1 is analysed in more detail. Consider a slightly rearranged equation of the assertion given by

$$\nabla I_{\mathbf{x}}(\boldsymbol{\mu}) - \nabla\, {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big) = \big(\nabla I_{\mathbf{x}}(\boldsymbol{\mu}) - \nabla\, {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)\big)\, \mathbf{n}_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}^{\mathrm{T}} \ . \tag{3.27}$$

Then, the first fundamental characteristic concerning the direction of the one-dimensional linear subspace containing the spatial derivatives follows immediately.

Corollary 1 (Orientation of Residual) *Let* $\mathbf{x} \in \mathcal{X}$, $\mathbf{n}_{\mathbf{x}} \in \mathbb{R}^3$, ${}^0I, I : \mathbb{R}^2 \to \mathbb{R}$, *and* ${}^0\boldsymbol{\mu}, \boldsymbol{\mu} \in \mathbb{R}^6$ *be given as in assertion 1.*

The difference between any two spatial texture derivatives $\nabla I_{\mathbf{x}}(\boldsymbol{\mu})$ *and* $\nabla\, {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)$ *is parallel to the surface normal* $\mathbf{n}_{\mathbf{x}}$, *i.e.,*

$$\big(\nabla I_{\mathbf{x}}(\boldsymbol{\mu}) - \nabla\, {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)\big) \parallel \mathbf{n}_{\mathbf{x}} \ .$$

This stationary property is split in the following into two orthogonal constraints (illustrated in figure 3.2), which facilitate an efficient computation of the spatial texture Jacobian.

Corollary 2 (Coplanarity with Origin) *Let* $\mathbf{x} \in \mathcal{X}$, $I : \mathbb{R}^2 \to \mathbb{R}$, *and* $\boldsymbol{\mu} \in \mathbb{R}^6$ *be given as in assertion 1.*

The spatial texture derivative $\nabla I_{\mathbf{x}}(\boldsymbol{\mu})$ *lies on a plane through the origin:*

$$\exists\, \mathbf{o}_{\mathbf{x}} \in \mathbb{R}^3, \|\mathbf{o}_{\mathbf{x}}\| = 1 : \quad \nabla I_{\mathbf{x}}(\boldsymbol{\mu}) \cdot \mathbf{o}_{\mathbf{x}} = 0 \ .$$

Corollary 3 (Coplanarity with Spatial Surface Texture Gradient) *Let* $\mathbf{x} \in \mathcal{X}$, $I : \mathbb{R}^2 \to \mathbb{R}$, *and* $\boldsymbol{\mu} \in \mathbb{R}^6$ *be given as in assertion 1.*

The spatial texture derivatives $\nabla I_{\mathbf{x}}(\boldsymbol{\mu})$ *lies on a plane defined by the spatial surface texture gradient* $\mathbf{g}_{\mathbf{x}}$, *i.e.,*

$$\exists\, \mathbf{g}_{\mathbf{x}} \in \mathbb{R}^3 \backslash \mathbf{0} : \quad \nabla I_{\mathbf{x}}(\boldsymbol{\mu}) \cdot \mathbf{g}_{\mathbf{x}} = \|\mathbf{g}_{\mathbf{x}}\|^2 \ .$$

Here, the spatial surface texture gradient $\mathbf{g}_{\mathbf{x}}$ corresponds to the three-dimensional gradient of the surface texture, which, per definition, lies on the tangential plane defined by $\mathbf{n}_{\mathbf{x}}$.

The corollaries 2 and 3 define two-dimensional subspaces containing the spatial derivative irrespective the actual object pose. The planes corresponding

to theses subspaces, given by $\mathbf{o_x}$ for corollary 2 and $\mathbf{g_x}$ for corollary 3, are orthogonal to the surface normal $\mathbf{n_x}$ and are orthogonal to each other. Thus, the normals corresponding to these three planes form an orthonormal system. The intersection of the planes $\mathbf{o_x}$ and $\mathbf{g_x}$ defines the static, one-dimensional linear subspace of the spatial texture derivative. The spatial surface texture gradient $\mathbf{g_x}$ is a special representative of this subspace.

The direction $\mathbf{d_x}$ of the spatial surface texture gradient $\mathbf{g_x}$ and the normal of the orthogonal plane $\mathbf{o_x}$ are computed a priori by

$$\mathbf{o_x} = \frac{\mathbf{n_x} \times \nabla {}^0I_\mathbf{x}({}^0\boldsymbol{\mu})}{\|\mathbf{n_x} \times \nabla {}^0I_\mathbf{x}({}^0\boldsymbol{\mu})\|} \quad , \qquad \mathbf{d_x} = \mathbf{o_x} \times \mathbf{n_x} \; , \tag{3.28}$$

for a surface model point $\mathbf{x}$.

Additionally to the above stationary constraints derived from equation 3.27 a view dependent constraint is introduced that allows to uniquely define the spatial texture Jacobian within the one-dimensional subspace (see also figure 3.2). For this purpose, a basic property of full perspective projection is used. For every $\mathbf{y} \in \mathbb{R}^3$, the derivative of perspective projection spans a two-dimensional space orthogonal to the line of sight $\mathbf{y}$, that is

$$\partial_\mathbf{y} p(\mathbf{y}) \cdot \mathbf{y} = \mathbf{0} \; . \tag{3.29}$$

This fact leads to the last, dynamic constraint. Beforehand, the pose dependent line of sight for the object point $\mathbf{x}$ is defined relative to the object frame reading

$$\mathbf{r_x}(\boldsymbol{\mu}) = m^{-1}(\mathbf{0}, \boldsymbol{\mu}) - \mathbf{x} = -\left(\partial_\mathbf{x} m(\mathbf{x}, \boldsymbol{\mu})\right)^{-1} m(\mathbf{x}, \boldsymbol{\mu}) \; . \tag{3.30}$$

The orthogonality of the ray to the partial derivative $\partial_\mathbf{x} p(m(\mathbf{x}, \boldsymbol{\mu}))$ follows immediately from equation 3.29 and 3.30.

Assertion 2 (Orthogonality with Line of Sight) *Let $\mathbf{x} \in \mathcal{X}$, $I : \mathbb{R}^2 \to \mathbb{R}$, and $\boldsymbol{\mu} \in \mathbb{R}^6$ be given as in assertion 1.*

The spatial texture derivative $\nabla I_\mathbf{x}(\boldsymbol{\mu})$ is orthogonal to the line of sight $\mathbf{r_x}(\boldsymbol{\mu})$ from the surface point to the optical centre:

$$\nabla I_\mathbf{x}(\boldsymbol{\mu}) \cdot \mathbf{r_x}(\boldsymbol{\mu}) = 0 \; .$$

Also this last assertion determines a plane constraining the surface texture Jacobian. Obviously, the combination of all three planar constraints determines the Jacobian uniquely. In a straightforward approach, the Jacobian can be computed by setting up a linear equation system by the assertion 1 (corollaries 2 and 3) and the assertion 2. Hereafter however, a computationally more efficient solution is proposed.

By merging the constraints of corollary 2 and assertion 2 the spatial texture Jacobian is defined up to scale, namely $\nabla I_\mathbf{x}(\boldsymbol{\mu}) \parallel \mathbf{o_x} \times \mathbf{r_x}(\boldsymbol{\mu})$. The combination with the corollary 3 then allows to map the current spatial texture derivative step by step

$$\nabla I_\mathbf{x}(\boldsymbol{\mu}) \; \mathbf{n_x}\mathbf{n_x}^\mathrm{T} \quad \overset{C3}{=} \quad \frac{\nabla {}^0I_\mathbf{x}({}^0\boldsymbol{\mu}) \; \mathbf{d_x}}{\nabla I_\mathbf{x}(\boldsymbol{\mu}) \; \mathbf{d_x}} \nabla I_\mathbf{x}(\boldsymbol{\mu}) \; \mathbf{n_x}\mathbf{n_x}^\mathrm{T} \tag{3.31}$$

$$\overset{C2\&\,A2}{=} \quad \nabla {}^0I_\mathbf{x}({}^0\boldsymbol{\mu}) \; \mathbf{d_x} \frac{\left(\mathbf{o_x} \times \mathbf{r_x}(\boldsymbol{\mu})\right)^\mathrm{T} \mathbf{n_x}}{\left(\mathbf{o_x} \times \mathbf{r_x}(\boldsymbol{\mu})\right)^\mathrm{T} \mathbf{d_x}} \mathbf{n_x}^\mathrm{T} \tag{3.32}$$

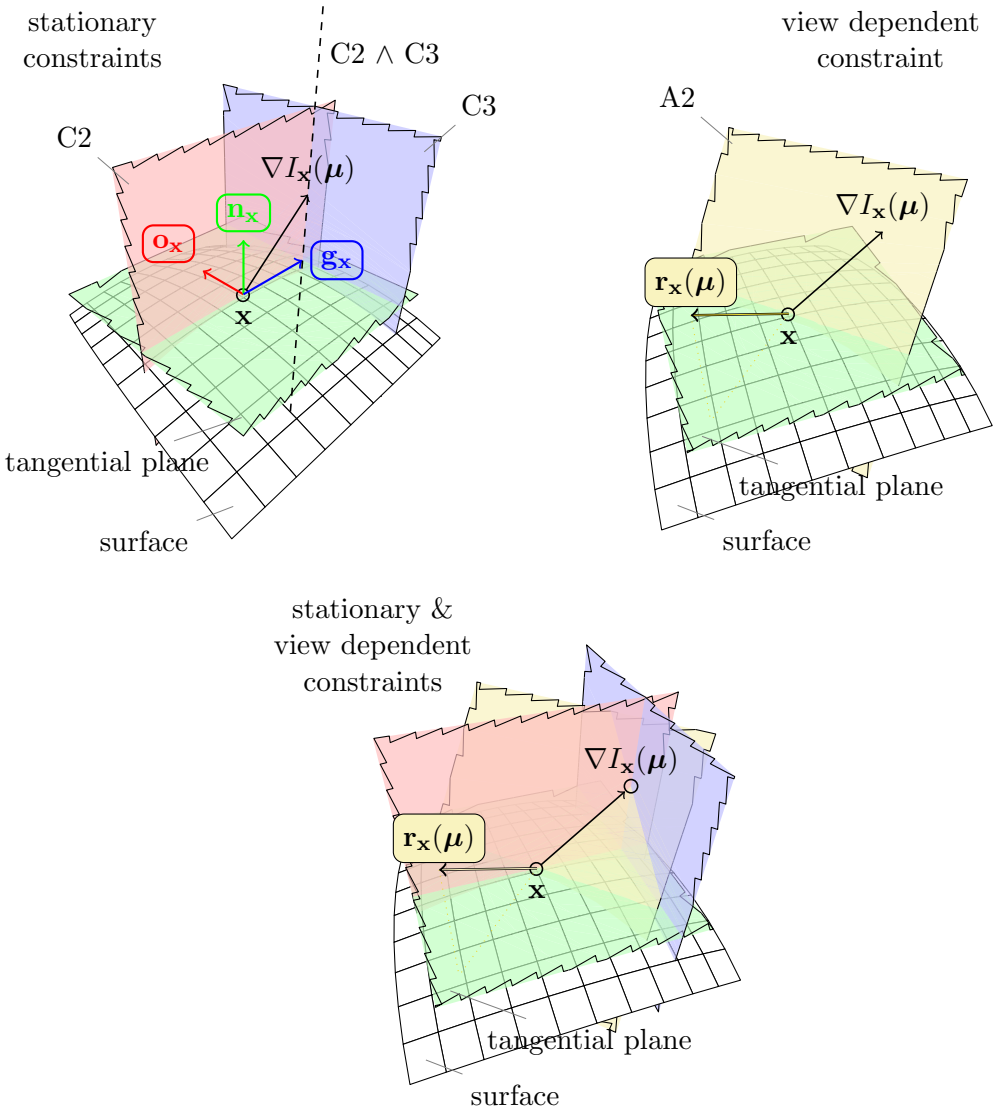

Figure 3.2: Illustration of the image-constancy constraints in 3-d for a surface point $\mathbf{x}$, surface normal $\mathbf{n}_{\mathbf{x}}$, spatial surface texture gradient $\mathbf{g}_{\mathbf{x}}$, and viewing vector $\mathbf{r}_{\mathbf{x}}(\boldsymbol{\mu})$. Top left: stationary constraints represented in corollaries 1 and 2. Top right: dynamic constraint expressed in assertion 2. Bottom: combination of stationary and dynamic constraints.

into a term independent of the current image derivative. Note, that the fraction $\frac{(\mathbf{o}_{\mathbf{x}}\times\mathbf{r}_{\mathbf{x}})^{\mathrm{T}}\mathbf{n}_{\mathbf{x}}}{(\mathbf{o}_{\mathbf{x}}\times\mathbf{r}_{\mathbf{x}})^{\mathrm{T}}\mathbf{d}_{\mathbf{x}}}$ corresponds to the tangent of the angle enclosing the spatial texture derivative and the surface texture gradient. The expression is further simplified taking the orthonormality of $\mathbf{n}_{\mathbf{x}}$, $\mathbf{o}_{\mathbf{x}}$, $\mathbf{d}_{\mathbf{x}}$ into account and finally reads

$$\nabla I_{\mathbf{x}}(\boldsymbol{\mu})\ \mathbf{n}_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}^{\mathrm{T}} = -\nabla {}^{0}I_{\mathbf{x}}({}^{0}\boldsymbol{\mu})\ \mathbf{d}_{\mathbf{x}}\frac{\mathbf{r}_{\mathbf{x}}(\boldsymbol{\mu})^{\mathrm{T}}\mathbf{d}_{\mathbf{x}}}{\mathbf{r}_{\mathbf{x}}(\boldsymbol{\mu})^{\mathrm{T}}\mathbf{n}_{\mathbf{x}}}\mathbf{n}_{\mathbf{x}}^{\mathrm{T}}\,. \tag{3.33}$$

Thus by computing the line of sight $\mathbf{r}_{\mathbf{x}}(\boldsymbol{\mu})$ to the surface point in question, the

current spatial image derivative

$$\nabla I_{\mathbf{x}}(\boldsymbol{\mu}) = \nabla {}^{0}I_{\mathbf{x}}({}^{0}\boldsymbol{\mu}) \left(\mathbb{1} - \mathbf{n}_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}^{\mathrm{T}} - \frac{\mathbf{r}_{\mathbf{x}}(\boldsymbol{\mu})^{\mathrm{T}}\,\mathbf{d}_{\mathbf{x}}}{\mathbf{r}_{\mathbf{x}}(\boldsymbol{\mu})^{\mathrm{T}}\,\mathbf{n}_{\mathbf{x}}} \mathbf{d}_{\mathbf{x}}\mathbf{n}_{\mathbf{x}}^{\mathrm{T}} \right) \tag{3.34}$$

can be expressed in terms of the spatial derivative of the reference image times a factor related to the imaging geometry.

3.2.4 Tracking with the Image-Constancy Assumption (IC)

The relationship established in the previous section between the spatial texture derivative in two arbitrary views allows to predict the spatial Jacobian for any object pose given the spatial Jacobian in the reference image [121]. In order to track the object in a sequence of camera images by means of a maximum likelihood estimator, the spatial Jacobian $\nabla I_{\mathbf{x}}(\boldsymbol{\mu})$ has to be transformed to the current pose Jacobian $\partial_{\boldsymbol{\mu}} I_{\mathbf{x}}(\boldsymbol{\mu})$. The overall efficiency depends thus not only on the prediction of the spatial Jacobian but also on the subsequent transformation to the motion Jacobian.

Here, the choice of the motion reference frame determines the computational efficiency of the approach. By definition, the spatial derivatives analysed in detail in the previous section refer to texture variation for unconstrained translation of the surface point in the object coordinate frame. By virtue of its definition, the spatial derivative points in the direction of highest change of the intensity value. These derivatives are finally mapped to motion derivatives via the motion field, which describes the motion of all single surface points under variation of pose. Since the spatial derivatives are computed with respect to the object coordinate frame, it is most convenient to refer pose variations to the same coordinate frame. The motion field is constant if the reference frame of section 3.1.1 is set to the moving object frame.

Thus, the objective function for a maximum likelihood estimator is set up for the moving reference frame. Here, the overall motion is considered as a composition of two nested motions, the previous motion estimate $\hat{\boldsymbol{\mu}}$ and a small variation in motion $\delta\boldsymbol{\mu}$. Herewith, the formerly defined objective function 3.18 is reformulated to

$$O(\delta\boldsymbol{\mu}) = \sum_{\mathbf{x}\in X} \left(I_{\mathbf{x}}(\hat{\boldsymbol{\mu}} \circ \delta\boldsymbol{\mu}) - {}^{0}I_{\mathbf{x}}({}^{0}\boldsymbol{\mu}) \right)^2 \ . \tag{3.35}$$

This objective function is minimised with a Gauss-Newton approximation to the Hessian. The corresponding linear equation system

$$\begin{aligned} \sum_{\mathbf{x}\in X} \partial_{\delta\boldsymbol{\mu}} I_{\mathbf{x}}(\hat{\boldsymbol{\mu}} \circ \delta\boldsymbol{\mu})^{\mathrm{T}} \cdot \partial_{\delta\boldsymbol{\mu}} I_{\mathbf{x}}(\hat{\boldsymbol{\mu}} \circ \delta\boldsymbol{\mu})\Big|_{\delta\boldsymbol{\mu}=0} \hat{\delta\boldsymbol{\mu}} = \\ \sum_{\mathbf{x}\in X} \partial_{\delta\boldsymbol{\mu}} I_{\mathbf{x}}(\hat{\boldsymbol{\mu}} \circ \delta\boldsymbol{\mu})^{\mathrm{T}}\Big|_{\delta\boldsymbol{\mu}=0} \left(I_{\mathbf{x}}(\hat{\boldsymbol{\mu}}) - {}^{0}I_{\mathbf{x}}({}^{0}\boldsymbol{\mu}) \right) \end{aligned} \tag{3.36}$$

is repeatedly solved at the moving pose estimates $\hat{\boldsymbol{\mu}}$ for the pose variation $\hat{\delta\boldsymbol{\mu}}$. The estimated variation $\hat{\delta\boldsymbol{\mu}}$ is successively combined with the motion estimate $\hat{\boldsymbol{\mu}}$

to a new pose estimate according to the directives for moving reference frames of section 3.1.1. Hence, $\delta\boldsymbol{\mu} = 0$ at the beginning of each iteration, which conforms with the identity transform.

The derivatives with respect to the pose variation $\delta\boldsymbol{\mu}$ for the moving reference frame is composed of the spatial derivatives $\nabla I_{\mathbf{x}}(\hat{\boldsymbol{\mu}})$ and the motion field $\partial_{\delta\boldsymbol{\mu}} m(\mathbf{x}, \delta\boldsymbol{\mu})$ according to the chain rules of derivation, reading

$$\partial_{\delta\boldsymbol{\mu}} I_{\mathbf{x}}(\hat{\boldsymbol{\mu}} \circ \delta\boldsymbol{\mu}) \Big|_{\delta\boldsymbol{\mu}=0} = \nabla I_{\mathbf{x}}(\hat{\boldsymbol{\mu}}) \cdot \partial_{\delta\boldsymbol{\mu}} m(\mathbf{x}, \delta\boldsymbol{\mu}) \Big|_{\delta\boldsymbol{\mu}=0} . \tag{3.37}$$

With the relationship established in equation 3.34 this Jacobian is approximated for a pose $\hat{\boldsymbol{\mu}}$ close to the true pose $\boldsymbol{\mu}^\star$ by

$$\begin{aligned} &\partial_{\delta\boldsymbol{\mu}} I_{\mathbf{x}}(\hat{\boldsymbol{\mu}} \circ \delta\boldsymbol{\mu}) \Big|_{\delta\boldsymbol{\mu}=0} \approx \\ &\quad \nabla {}^0 I_{\mathbf{x}}({}^0\boldsymbol{\mu}) \left(\mathbb{1} - \mathbf{n}_{\mathbf{x}} \mathbf{n}_{\mathbf{x}}^{\mathrm{T}} - \frac{\mathbf{r}_{\mathbf{x}}(\hat{\boldsymbol{\mu}})^{\mathrm{T}} \mathbf{d}_{\mathbf{x}}}{\mathbf{r}_{\mathbf{x}}(\hat{\boldsymbol{\mu}})^{\mathrm{T}} \mathbf{n}_{\mathbf{x}}} \mathbf{d}_{\mathbf{x}} \mathbf{n}_{\mathbf{x}}^{\mathrm{T}} \right) \partial_{\delta\boldsymbol{\mu}} m(\mathbf{x}, \delta\boldsymbol{\mu}) \Big|_{\delta\boldsymbol{\mu}=0} . \end{aligned} \tag{3.38}$$

The expression can be efficiently computed since only the ray $\mathbf{r}_{\mathbf{x}}(\hat{\boldsymbol{\mu}})$ depends on the current pose estimate and all other terms can be computed a priori. In detail, the motion Jacobian is composed of two terms. One term specifies the motion Jacobian for a perpendicular view on the surface point $\mathbf{x}$, reading

$$\Omega_{\mathbf{x}} = \nabla {}^0 I_{\mathbf{x}}({}^0\boldsymbol{\mu}) \left(\mathbb{1} - \mathbf{n}_{\mathbf{x}} \mathbf{n}_{\mathbf{x}}^{\mathrm{T}} \right) \partial_{\delta\boldsymbol{\mu}} m(\mathbf{x}, \delta\boldsymbol{\mu}) \Big|_{\delta\boldsymbol{\mu}=0} . \tag{3.39}$$

The other term refers to the component of the motion Jacobian in normal direction. The direction of this component is hereafter referred to by

$$\Omega_{\mathbf{x}}^{\perp} = -\nabla {}^0 I_{\mathbf{x}}({}^0\boldsymbol{\mu}) \left(\mathbf{d}_{\mathbf{x}} \mathbf{n}_{\mathbf{x}}^{\mathrm{T}} \right) \partial_{\delta\boldsymbol{\mu}} m(\mathbf{x}, \delta\boldsymbol{\mu}) \Big|_{\delta\boldsymbol{\mu}=0} . \tag{3.40}$$

Both 6-dimensional vectors $\Omega_{\mathbf{x}}$ and $\Omega_{\mathbf{x}}^{\perp}$ are constant for each model point $\mathbf{x}$ and can be easily computed off-line. At runtime, they are merged to the motion Jacobian at a specific pose estimate $\hat{\boldsymbol{\mu}}$ according to

$$\partial_{\delta\boldsymbol{\mu}} I_{\mathbf{x}}(\hat{\boldsymbol{\mu}} \circ \delta\boldsymbol{\mu}) \Big|_{\delta\boldsymbol{\mu}=0} \approx \Omega_{\mathbf{x}} + \frac{\mathbf{r}_{\mathbf{x}}(\hat{\boldsymbol{\mu}})^{\mathrm{T}} \mathbf{d}_{\mathbf{x}}}{\mathbf{r}_{\mathbf{x}}(\hat{\boldsymbol{\mu}})^{\mathrm{T}} \mathbf{n}_{\mathbf{x}}} \Omega_{\mathbf{x}}^{\perp} . \tag{3.41}$$

The number of floating point operations for the computation of the texture Jacobian as shown in Table 3.2 are thus substantially reduced compared to the straight forward computation reported in Table 3.1.

	#fp mult.	#fp add.
$\partial_{\delta\boldsymbol{\mu}} I_{\mathbf{x}}(\hat{\boldsymbol{\mu}} \circ \delta\boldsymbol{\mu})$	13	10

Table 3.2: Number of floating point operations employed for the prediction of the Jacobian for a single 3-d model point under the image-constancy assumption.

3.2.5 Tracking with the Relaxed Image-Constancy Assumption (IC-R)

Prediction of the view-dependent motion Jacobian with the image-constancy assumption in 3-d reduces the computational efficiency of maximum likelihood estimation considerably. Minimisation by means of the Newton algorithm, however, leaves room for further improvements. So, the Gauss-Newton approximation of the Hessian requires $21N$ floating point multiplications and $21(N-1)$ additions.

The computational efforts can be further reduced by assuming that the motion Jacobian remains constant over changes in view [125]. This functional model represents an approximation and thus introduces an error in the estimation process [121].

The error by the motion Jacobian is traced back to the difference between the true spatial Jacobian and the constant spatial Jacobian of the reference view. The difference, as described in equation 3.27, is most easily quantified for points $\mathbf{x} \in X$ where the line of sight is perpendicular to the surface in the reference view, that is $\nabla {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big) \cdot \mathbf{n}_{\mathbf{x}} = 0$. Then the difference is entirely expressed by the component of the true spatial Jacobian in normal direction, that is

$$\big\|\nabla I_{\mathbf{x}}(\boldsymbol{\mu}^\star) - \nabla {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)\big\| = \big|\nabla I_{\mathbf{x}}(\boldsymbol{\mu}^\star)\,\mathbf{n}_{\mathbf{x}}\big| \quad \Leftrightarrow \quad \mathbf{r}_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big) \times \mathbf{n}_{\mathbf{x}} = 0\,. \tag{3.42}$$

According to equation 3.33 the true spatial Jacobian can be expressed by the reference Jacobian and thus its Euclidean norm is rewritten as

$$\big|\nabla I_{\mathbf{x}}(\boldsymbol{\mu}^\star)\,\mathbf{n}_{\mathbf{x}}\big| = \big|\nabla {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)\,\mathbf{d}_{\mathbf{x}}\big| \left|\frac{\mathbf{r}_{\mathbf{x}}(\boldsymbol{\mu}^\star)^{\mathrm{T}}\,\mathbf{d}_{\mathbf{x}}}{\mathbf{r}_{\mathbf{x}}(\boldsymbol{\mu}^\star)^{\mathrm{T}}\,\mathbf{n}_{\mathbf{x}}}\right|\,. \tag{3.43}$$

Now, let α denote the angle between the surface normal and the projection of the line of sight on the plane $\mathbf{n}_{\mathbf{x}} \times \mathbf{d}_{\mathbf{x}}$, and accordingly $\tan(\alpha) = \left|\frac{\mathbf{r}_{\mathbf{x}}(\boldsymbol{\mu}^\star)^{\mathrm{T}}\mathbf{d}_{\mathbf{x}}}{\mathbf{r}_{\mathbf{x}}(\boldsymbol{\mu}^\star)^{\mathrm{T}}\mathbf{n}_{\mathbf{x}}}\right|$. With this substitution at hand and with the initial assumption $\nabla {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)\cdot\mathbf{n}_{\mathbf{x}} = 0$ equivalent to $\nabla {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big) \parallel \mathbf{d}_{\mathbf{x}}$, equation 3.42 is rewritten as

$$\big\|\nabla I_{\mathbf{x}}(\boldsymbol{\mu}^\star) - \nabla {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)\big\| = \big\|\nabla {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)\big\|\tan(\alpha) \quad \Leftrightarrow \quad \mathbf{r}_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big) \times \mathbf{n}_{\mathbf{x}} = 0\,. \tag{3.44}$$

It can be easily inferred that the residuum grows as the line of sight approaches the direction $\mathbf{d}_{\mathbf{x}}$ of the surface texture gradient.

For moderate differences between the angle in the current view and the reference view the residual can be neglected and the spatial derivative reduces to

$$\nabla I_{\mathbf{x}}(\boldsymbol{\mu}) \approx \nabla {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)\;, \quad \mathbf{x} \in \mathcal{X}\,. \tag{3.45}$$

This approximation corresponds to the derivative of a relaxed image-constancy assumption in 3-d that reads

$$I_{\mathbf{y}}(\boldsymbol{\mu}) = {}^0I_{\mathbf{y}}\big({}^0\boldsymbol{\mu}\big)\;, \quad \mathbf{y} \in \bigcup_{\mathbf{x}\in\mathcal{X}} \mathcal{B}_\gamma\big(\mathbf{x}, \mathbb{R}^3\big) \tag{3.46}$$

in a γ-neighbourhood of the surface $\mathcal{X}$. Here, relaxed refers to constancy and not to assumption. The approximation 3.45 simplifies the image Jacobian 3.36 at $\delta\boldsymbol{\mu} = 0$ to

$$\partial_{\delta\boldsymbol{\mu}} I_{\mathbf{x}}(\hat{\boldsymbol{\mu}} \circ \delta\boldsymbol{\mu}) \Big|_{\delta\boldsymbol{\mu}=0} \approx \nabla {}^0I_{\mathbf{x}}({}^0\boldsymbol{\mu}) \cdot \partial_{\delta\boldsymbol{\mu}} m(\mathbf{x}, \delta\boldsymbol{\mu}) \Big|_{\delta\boldsymbol{\mu}=0} . \tag{3.47}$$

Hence, the image Jacobian and Hessian are constant in the framework of the relaxed image-constancy assumption (cf. [120]).

Consequently, at runtime only a minimal set of calculations is required for the Newton algorithm. First, the residual $I_{\mathbf{x}}(\hat{\boldsymbol{\mu}}) - {}^0I_{\mathbf{x}}({}^0\boldsymbol{\mu})$ between the current and reference texture is computed. This residual is mapped thereafter to the motion templates represented by the constant image Jacobian $\partial_{\delta\boldsymbol{\mu}} {}^0I_{\mathbf{x}}({}^0\boldsymbol{\mu} \circ \delta\boldsymbol{\mu})$ at $\delta\boldsymbol{\mu} = 0$. Finally, the linear equation system is solved for the pose variation $\delta\boldsymbol{\mu}$.

Note that the estimation of motion between two iterations of the Newton algorithm is a process integrating contributions from all points of the surface model. Thus, the error on the direction of the spatial derivative for a single surface point is alleviated if texture gradient directions are uniformly distributed on the surface.

3.3 Multi-Hypotheses Tracking

The robustness of tracking by means of maximum likelihood estimation as presented in the previous sections depends among other things on the speed and the region of convergence of the applied minimisation algorithm. Because of the ragged nature of the likelihood function, the minimisation process starting from a single pose hypothesis is prone to get caught in a local minimum representing a wrong pose estimate.

In order to increase robustness, prior information on the motion dynamics can be added to the estimation process. Here, Kalman filters are well known representatives of single-hypothesis trackers incorporating constraints imposed by an a priori known motion model. The specific approaches of Kalman Filter (KF) or Extended Kalman Filters (EKF), however, are not feasible here because of the high dimensionality of the image Jacobian.

Another possibility to reduce the risk for local minima is to evaluate multiple pose hypotheses at once. These samples are optimally distributed given the model of dynamics, the complete history of observations, and knowledge of the initial pose. Systematic sampling is accomplished by Monte-Carlo methods, which approximate an arbitrary continuous probability density with a finite set of such samples. Thus, Monte-Carlo methods can handle not only Gaussian distributions as assumed in Kalman filtering but also general uni-modal or multi-modal distributions that cannot be modelled analytically. The model of dynamics, which constrains the state space over time, is assumed here as dependent on only a pair of successive states. Thus, propagation of state distributions from one observation to the following is reduced to Markov-Chain rules. The combination of the observation model presented in section 3.1.3 with a sim-

ple model of motion dynamics allows to devise a Markov-Chain Monte-Carlo (MCMC) method for appearance-based object tracking in 6 DoF.

3.3.1 Markov-Chain Monte-Carlo Methods

The concept of sampling from a continuous probability distribution and tracking the distribution represented through the samples over time has been developed separately by researches in different contexts. In the community of probabilistic inference these approaches are referred to as Markov-Chain Monte-Carlo Methods. These methods represent the dynamics in Bayesian filters with a finite set of samples of the continuous state and have been intensively studied in the past century. Markov-Chain Monte-Carlo Methods have their counterparts in computer vision, where they are called Particle Filters. Isard and Blake [70] introduced Particle Filters to the vision community with the CONDENSATION algorithm in 1998. Like bootstrap filters this type of particle filter uses a special strategy for sampling from the posterior distribution.

The following outlines Bayesian filtering as the theoretical background of Markov-Chain Monte-Carlo methods for continuous probability densities. Subsequently, the major strategies for sampling from the densities are mentioned. See [5] for more details on the topic.

Bayesian Filter

Bayesian filters offer a methodology for recursively computing a posterior probability density function (pdf) based on a previous density function.

Let $\mathbf{x}_k$ be the unobservable and thus hidden multi-dimensional state of a stochastic system at time $k \in \mathbb{N}$.The corresponding multi-dimensional observation for the same time instant is denoted by $\mathbf{z}_k$ while the set of all observations up to time k is given by $\mathbf{z}_{1:k}$. The probability density function in question is $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ for state $\mathbf{x}_k$ given the current measurement together with all previous measurements $\mathbf{z}_{1:k}$.

The link from a previous density function to the posterior probability density function is established in two stages. The first stage involves the *prediction* of state $\mathbf{x}_k$ based on the pdf $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ describing the system dynamics. Given the Markov assumption, the transition probability from all states and observations up to time $k-1$ to the next state $\mathbf{x}_k$ depends only on the previous state $\mathbf{x}_{k-1}$, hence $p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{z}_{1:k-1}) \approx p(\mathbf{x}_k|\mathbf{x}_{k-1})$. The probability for the current state $\mathbf{x}_k$ given the previous observations is then obtained with the Chapman-Kolmogorov equation

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\, p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\, d\mathbf{x}_{k-1} \,. \tag{3.48}$$

In the second stage, the current observation $\mathbf{z}_k$ is used to *update* the predicted density function to the posterior pdf. Following Bayes' rule, the posterior pdf is expressed by

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \,, \tag{3.49}$$

which comprises the probability $p(\mathbf{z}_k|\mathbf{x}_k)$ of the current observation given the current state $\mathbf{x}_k$ and the normalisation constant $p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k) \cdot p(\mathbf{x}_k|\mathbf{z}_{1:k-1})\, d\mathbf{x}_k$.

The model employed in the Bayesian filter is a hidden Markov model (HMM) with the transition probability $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and observation probability $p(\mathbf{z}_k|\mathbf{x}_k)$.

Sequential Importance Sampling (SIS)

In general, the above recursive propagation of the posterior density cannot be solved analytically. Therefore, sampling techniques are employed to approximate the continuous density function by a set of random samples of the hidden state. In the following, sequential importance sampling is presented as a common basis for many Monte-Carlo filters.

Let a sample $\mathbf{x}_{0:k}$ represent the history of all states up to time k including the initial state $\mathbf{x}_0$. In the simulation, the posterior pdf is characterised by the set $\{(\mathbf{x}_{0:k}^{(i)}, w_k^{(i)})|i = 1, 2, \ldots, {}^{\mathrm{s}}N\}$ consisting of the samples $\mathbf{x}_{0:k}^{(i)}$ and the corresponding weights $w_k^{(i)}$. The continuous posterior density is approximated by

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{{}^{\mathrm{s}}N} w_k^{(i)} \delta\Big(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^{(i)}\Big) \ , \tag{3.50}$$

whereas $0 \leq w_k^{(i)}$ and $\sum_i w_k^{(i)} = 1$. The representation strongly relies on the choice of the samples also called particles. Since sample states $\mathbf{x}_{0:k}$ cannot be drawn from the yet unknown pdf $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ an auxiliary and yet to be defined distribution $q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ known as *importance density* is used instead. So, $\mathbf{x}_{0:k}^{(i)} \sim q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ while the weights are defined up to proportionality by

$$w_k^{(i)} \propto \frac{p\Big(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{1:k}\Big)}{q\Big(\mathbf{x}_{0:k}^{(i)}|\mathbf{z}_{1:k}\Big)} \ . \tag{3.51}$$

This principle is called *importance sampling* and forms the basis for a recurrent description of the weights and thus also of the posterior density. Generally, arbitrary importance densities can be used whereas $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \neq 0 \Rightarrow q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \neq 0$. However, the better the chosen importance density resembles the unknown pdf $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ the less samples have to be drawn to obtain a good approximation of the unknown pdf.

Moreover, a wisely chosen importance density eases the simulation of the process. For instance, the equation 3.51 can be transformed into a more convenient expression if the importance density is selected such that it factorises to the form

$$q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})\, q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \ . \tag{3.52}$$

Given the Markovian assumption, the posterior density can be described up to proportionality by

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{x}_{k-1})\, p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \ , \tag{3.53}$$

which reflects the stages of prediction 3.48 and update 3.49 as described in the above subsection. Refer to [5] for a detailed derivation of the expressions. In conjunction with the factorisation 3.52, the computation of weights 3.51 can be transformed into the recursive expression given by

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p\left(\mathbf{z}_k|\mathbf{x}_k^{(i)}\right) p\left(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}\right)}{q\left(\mathbf{x}_k^{(i)}|\mathbf{x}_{0:k-1}^{(i)}, \mathbf{z}_{1:k}\right)} \ . \tag{3.54}$$

The Markovian assumption simplifies the expressions further. The history of states is sufficiently represented by the current state $\mathbf{x}_k$, which, in turn, depends only on the previous state $\mathbf{x}_{k-1}$ and the current measurement $\mathbf{z}_k$. Hence, the set $\{(\mathbf{x}_{0:k}^{(i)}, w_k^{(i)})|i = 1, 2, \ldots, {}^{\mathrm{s}}N\}$ can be replaced by $\{(\mathbf{x}_k^{(i)}, w_k^{(i)})|i = 1, 2, \ldots, {}^{\mathrm{s}}N\}$. Accordingly, samples are drawn from the single state importance density $q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$ instead of $q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$, which eventually simplifies to $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$.

Sampling Importance Resampling (SIR)

The SIS particle filters as described above suffer from the degeneration of the set of samples, which occurs when the particles are spread over the state space and exhibit insignificant weights except for one state. This problem can be solved in two ways. The first possibility consists in altering the importance density to condition the compactness of the samples. The second optimisation involves a resampling stage where the particles are not drawn from the importance density distribution but from the approximated posterior distribution itself.

Both optimisations are considered by the method of sampling importance resampling with simple choices for the importance density and the resampling stage. The importance density $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$ is set to the prediction density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, which simplifies the computation of the weight to

$$w_k^{(i)} \propto w_{k-1}^{(i)} p\left(\mathbf{z}_k|\mathbf{x}_k^{(i)}\right) \ . \tag{3.55}$$

In the resampling stage a new set $\{\mathbf{x}_{k-1}^{(i*)}|i = 1, 2, \ldots, {}^{\mathrm{s}}N\}$ is generated by drawing samples from the approximated posterior density function 3.50, which ensures that $p\left(\mathbf{x}_{k-1}^{(i*)} = \mathbf{x}_{k-1}^{(i)}\right) = w_{k-1}^{(i)}$. Thereafter, the weights $w_{k-1}^{(i)}$ are reset to $1/{}^{\mathrm{s}}N$ since the density of the states represents already the desired distribution. Therefore, the computation of new weights simplifies further to

$$w_k^{(i*)} \propto p\left(\mathbf{z}_k|\mathbf{x}_k^{(i*)}\right) \ . \tag{3.56}$$

However, the above choices for the importance density and the re-sampling exhibit some drawbacks. The prediction with the density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ neglects the current observation and thus the state space will not be efficiently explored. Furthermore, the set of particles might quickly loose diversity because the particles are re-sampled from a discrete distribution rather than from a continuous distribution. Nevertheless, these weaknesses are compensated by the simplicity of the importance density responsible for sampling prominent states.

3.3.2 Monte-Carlo Based Shape-Texture Tracking

Up to now, Markov-Chain Monte-Carlo methods have been used in computer vision mainly for the purpose of tracking contours in the image. The state space for such applications is usually the parameter space of snakes describing the object contours whereas the observation is given by intensity gradients or texture gradients on selected intervals along the contour.

In the following, a Markov-Chain Monte-Carlo method for shape-texture based tracking in 6 DoF is presented [123, 56] that does not rely on features in the image such as edges or corners, i.e. the contour. Conforming to subsection 3.1.3, the object is tracked instead by a set of unordered 3-d points

$$X = \{\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_N\} \subset \mathbb{R}^3 \; . \tag{3.65}$$

These points are again characterised by their associated grey values ${}^0I_{\mathbf{x}}$, $\mathbf{x} \in X$, leading to a textured 3-d model of the object. Overall, ${}^0\mathrm{I} = ({}^0I_{\mathbf{x}_1}, {}^0I_{\mathbf{x}_2}, \ldots, {}^0I_{\mathbf{x}_N})$ refers to the reference texture of the model. The reference texture is gathered from any view on the object with known or manually registered object-to-camera pose following the rule established in equation 3.16.

The continuous state of the Monte-Carlo approach is the 6-DoF pose $\boldsymbol{\mu} \in \mathbb{R}^6$ of the object relative to the camera consisting of three translational and three rotational parameters as pointed out in subsection 3.1.1. The object pose changes over time as the object moves relative to the camera. Generally, the probability of transition from a specific pose to another is a priori unknown. In particular, neither object masses nor forces or torques applied to move the object are assumed to be known. Hence, the trajectory of the object cannot be physically predicted.

In consequence, the process noise is designed here to account for the unknown system dynamics. The process resembles simulated annealing exploration of the pose space as motivated in [47]. Hence, for a single observation multiple iterations k of a diffusion process are realised with

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + b^k \; \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{v}}) \tag{3.66}$$

where $\mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{v}})$ represents the white normal process noise with covariance matrix $\Sigma_{\mathbf{v}}$ and $0 < b < 1$ denotes the base of the decay term. In terms of the prediction or transition probability density this means

$$p\big(\boldsymbol{\mu}_{k+1}|\boldsymbol{\mu}_k\big) \propto \exp\left(-\frac{1}{2} b^k \left(\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k\right)^{\mathrm{T}} \Sigma_{\mathbf{v}} \left(\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k\right)\right) \; . \tag{3.67}$$

This simplistic model neglects possible hidden states such as velocity or acceleration of the object. However, it allows to use the model not only for pose prediction between two frames but also for the iterative re-evaluation of the set of particles for the same observation. According to the rules of sampling importance resampling, a previous distribution of object poses is propagated to a new distribution via the prediction density. The resulting distribution is updated thereafter to the posterior pdf $p\big(\boldsymbol{\mu}_{k+1}|I\big)$ given the current observation I. The

weights for the particle poses are updated within the annealed Monte-Carlo simulation according to the observation probability density

$$p\big(I|\boldsymbol{\mu}_{k+1}\big) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}\left\|\mathrm{I}\big(\boldsymbol{\mu}_{k+1}\big) - {}^{0}\mathrm{I}\big({}^{0}\boldsymbol{\mu}\big)\right\|^2\right) , \tag{3.68}$$

which conforms to the pdf 3.17 defined in section 3.1.3. In this expression, $\mathrm{I}\big(\boldsymbol{\mu}_{k+1}\big)$ represents the vector of texture values for the image I and the pose $\boldsymbol{\mu}_{k+1}$ at iteration $k+1$ computed according to equation 3.15.

Within the Monte-Carlo simulation, the set $\{(\boldsymbol{\mu}_k^{(i)}, w_k^{(i)})|i = 1, 2, \ldots, {}^{\mathrm{s}}N\}$ contains the pose hypotheses and the corresponding weights, i.e., the corresponding likelihood, at the current time instant t. Typically, only a single representative pose is condensed from this simulated distribution and passed to an application, e.g., robot control. A reasonable estimate $\hat{\boldsymbol{\mu}}_{k+1}$ is determined here by the particle with the highest weight

$$\hat{\boldsymbol{\mu}}_{k+1} = \boldsymbol{\mu}_{k+1}^{(j)} , \quad j = \underset{i}{\operatorname{argmax}}\, w_{k+1}^{(i)} . \tag{3.69}$$

Hence, this Particle Filtering process aims at maximising the likelihood $L(\hat{\boldsymbol{\mu}})$ of the pose estimate $\hat{\boldsymbol{\mu}}$ for the current observation I. See figure 3.3 for a summary of the algorithm.

For any iteration $k \geq 0$ process the particle set $\{(\boldsymbol{\mu}_k^{(i)}, w_k^{(i)}) | i = 1, 2, \ldots, {}^{\mathrm{s}}N\}$

1. Resampling
   Based on the discrete distribution of the current particle weights

$$W_{k,j} = \sum_{i=1}^{j} w_k^{(i)}, \quad 1 \leq j \leq {}^{\mathrm{s}}N \tag{3.57}$$

   Sample state $\boldsymbol{\mu'}_k^{(i)}$ for all $i \in \{1, 2, \ldots, {}^{\mathrm{s}}N\}$

   (a) Draw a uniformly distributed random number

$$r_k^{(i)} \sim \mathcal{U}(0, W_{k,{}^{\mathrm{s}}N}) \tag{3.58}$$

   (b) Determine state $\boldsymbol{\mu'}_k^{(i)} = \boldsymbol{\mu}_k^{(l)}$ with

$$l = \min\{j \mid r_k^{(i)} \leq W_{k,j}\} \tag{3.59}$$

2. Prediction
   For all re-sampled states $\boldsymbol{\mu'}_k^{(i)}$, $i \in \{1, 2, \ldots, {}^{\mathrm{s}}N\}$

   (a) Draw a normally distributed white noise vector

$$\mathbf{v}_k^{(i)} \sim \mathcal{N}(0, \Sigma_{\mathbf{v}}) \tag{3.60}$$

   (b) Propagate the state according to the process model

$$\boldsymbol{\mu}_{k+1}^{(i)} = \boldsymbol{\mu'}_k^{(i)} + b^k \, \mathbf{v}_k^{(i)} \ , \quad 0 < b < 1 \tag{3.61}$$

3. Observation
   For all new states $\boldsymbol{\mu}_{k+1}^{(i)}$, $i \in \{1, 2, .., {}^{\mathrm{s}}N\}$

   (a) Determine weight based on the current observation

$$w'^{(i)}_{k+1} = p\Big(I | \boldsymbol{\mu}_{k+1}^{(i)}\Big) \tag{3.62}$$

   (b) Normalise the new weights

$$w_{k+1}^{(i)} = \frac{1}{\sum_{j=1}^{{}^{\mathrm{s}}N} w'^{(j)}_{k+1}} w'^{(i)}_{k+1} \tag{3.63}$$

4. Extraction
   Determine the most probable state

$$\hat{\boldsymbol{\mu}}_{k+1} = \boldsymbol{\mu}_{k+1}^{(j)} \ , \quad j = \underset{i}{\operatorname{argmax}} \, w_{k+1}^{(i)} \ . \tag{3.64}$$

Figure 3.3: Shape-Texture based annealed Particle Filtering algorithm.

3.4 Evaluation

In the following, the tracking methods described in sections 3.2 and 3.3 are empirically evaluated and compared under both non real-time and real-time conditions. The performance of a method is determined first by its general ability to correctly estimate the object pose in relation to the inaccuracy of the initial pose estimate. Subsequently, these results are re-interpreted taking the computational costs into account. Hence, the methods are assessed in the end with respect to real-time conditions of limited computational resources and rotational and translational object velocities (subsection 3.4.4).

The performance evaluation is preceeded by the description of the objects to be tracked (subsection 3.4.1) and the description of the setup used for the evaluation (subsection 3.4.2). These subsections exemplify the processes of model acquisition and data representation. Furthermore, the objective function, the prediction the motion Jacobian, and the computational costs are evaluated in order to illustrate the properties of the methods presented in this chapter (subsection 3.4.3).

3.4.1 Model Acquisition and Representation

Three objects with distinct shape characteristics are used as a test set, namely a box, a bottle, and a sculpture. Surface parts are manually identified that are not subject to self-occlusion and that are characteristic for the object. These parts, as sketched in figures 3.4, 3.5, and 3.6, are of different sizes and exhibit a variety of shapes: a piecewise planar shape (box), a cylindrical shape (bottle), and a free-form shape (sculpture).

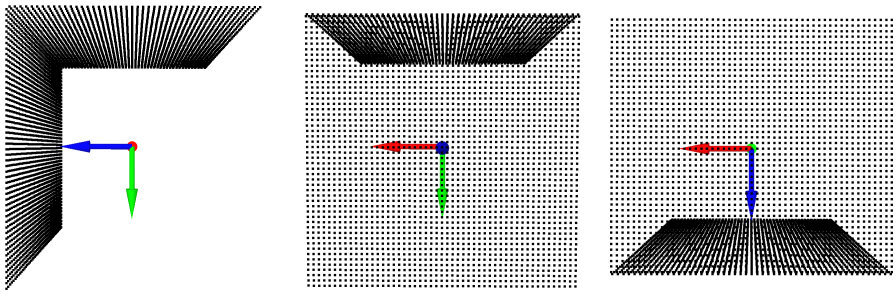

Figure 3.4: The three-dimensional point-set model of a rectangular edge of size 100 mm × 100 mm × 90 mm. From left to right: lateral, frontal, and top view.

The surfaces are sampled individually in regular steps of 2 mm (box), 1 mm (bottle), and approximately 2 mm (sculpture), leading to surface models of 4947 (box), 3721 (bottle), and 3790 (sculpture) 3-d points. While the former two shapes are sampled based on analytic surface descriptions, the latter shape is digitised with a robot-guided rotating laser-scanner [135, 19].

The last step in model acquisition consists in the registration of the 3-d point model with a single view referred to as the reference view. This view is chosen to correspond to a central view with respect to the encountered camera-

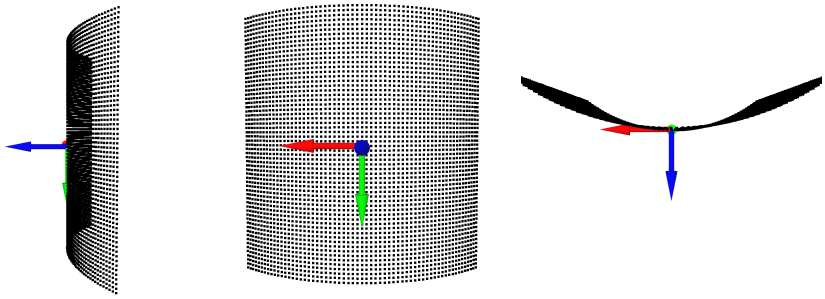

Figure 3.5: The three-dimensional point-set model of a 74° section of a cylinder of radius 46 mm and height 60 mm. From left to right: lateral, frontal, and top view.

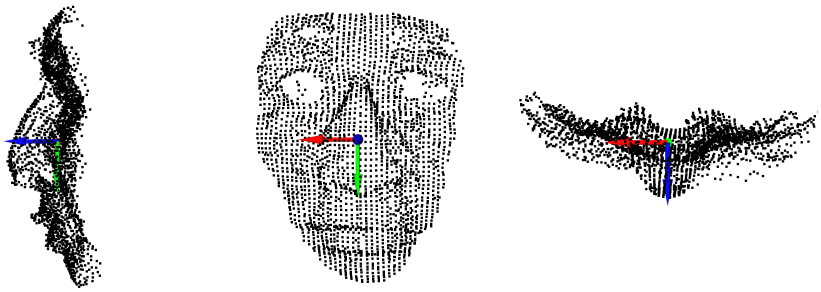

Figure 3.6: The three-dimensional point-set model of a sculpture with an approximate size of 65 mm × 83 mm × 29 mm. From left to right: lateral, frontal, and top view.

to-object poses. Figure 3.7 displays the reference images taken at these poses for the objects in question.

For the objects with an analytic surface description (box and bottle), the registration of the 3-d model with the camera image is performed manually. In this process, the image coordinates of a few surface points are manually determined. Thereafter, the corresponding 6-DoF object pose is estimated through non-linear optimisation of the projection error for these surface points. For the object digitised with the robot-guided laser-scanner (sculpture), the registration of the 3-d model with an eye-in-hand camera is implicitly given by an off-line

Figure 3.7: Reference images for shape-texture based object tracking. Left: box at 0.415m distance. Centre: bottle at 0.31m. Right: sculpture at 0.315m.

hand-eye calibration performed with appropriate tools [122, 134]. Eventually, both registration processes yield a textured three-dimensional point cloud as an object model. Figure 3.8 sketches the steps in the registration process, whereas figure 3.9 shows the obtained textured models.

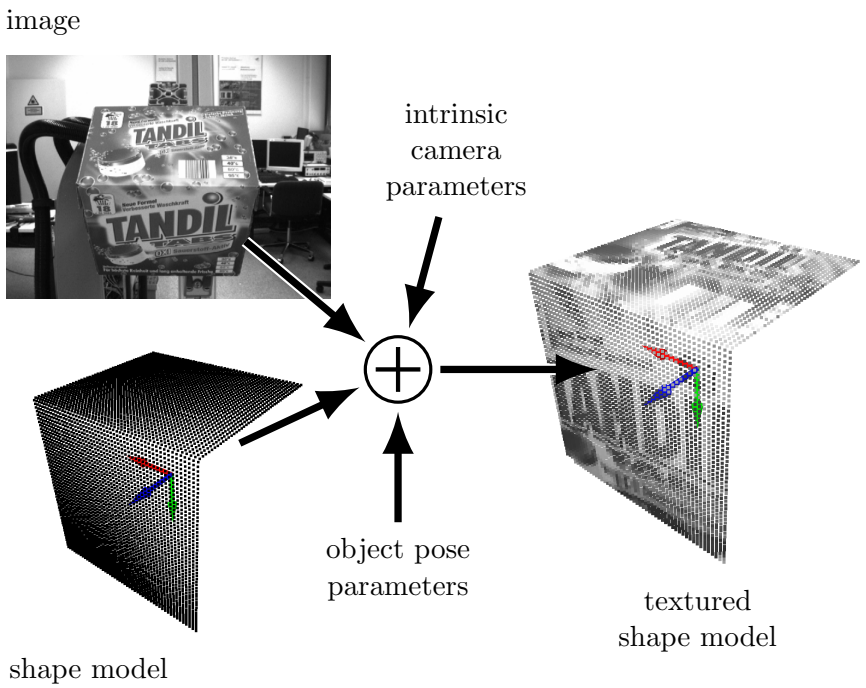

Figure 3.8: Registration of the image data with the shape model to generate a textured 3-d model of a real box object.

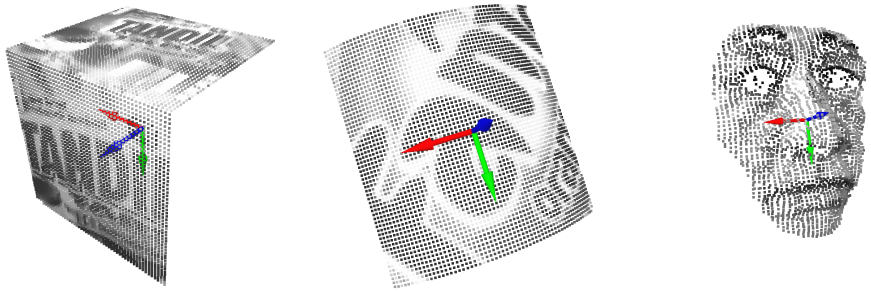

Figure 3.9: Textured shape models. Left: 3-d edge. Centre: 3-d cylinder. Right: 3-d face.

3.4.2 Data Acquisition

Grey-valued image data is acquired with a progressive scan camera at a resolution of 780×580 pixels and a lense aperture of 56° × 43°. The number of

pictures taken and the poses chosen for this purpose aim at efficiently sampling the space of object appearances.

For this purpose, 7 orientations and 8 positions of the object are identified relative to the camera viewpoint. The mentioned orientations comprise a $\pm 25^\circ$ rotation about the x-axis, a $\pm 15^\circ$ rotation about the y-axis, and a $\pm 40^\circ$ rotation around the z-axis. The object positions are chosen to span an obelisk-shaped volume of depth 250 mm and a frontal plane of size 112 mm×68 mm parallel to the camera at a distance of 310 mm. From the camera point of view, the vertices of the obelisk (the 8 positions) project to the upper left (2nd) quadrant of the image plane. Positions in only one quadrant are chosen because symmetric appearance distortions are expected at counterpart positions in the other quadrants.

These orientations and positions are finally combined with each other, yielding a total of 56 different object poses. Figure 3.10 sketches the different object poses employed for image acquisition. In order to guarantee the repeatability of these poses for all three objects, the camera is mounted on a robot. An offline eye-in-hand camera calibration together with an initial object-to-camera registration allow for the robot to realise the desired object poses relative to the camera.

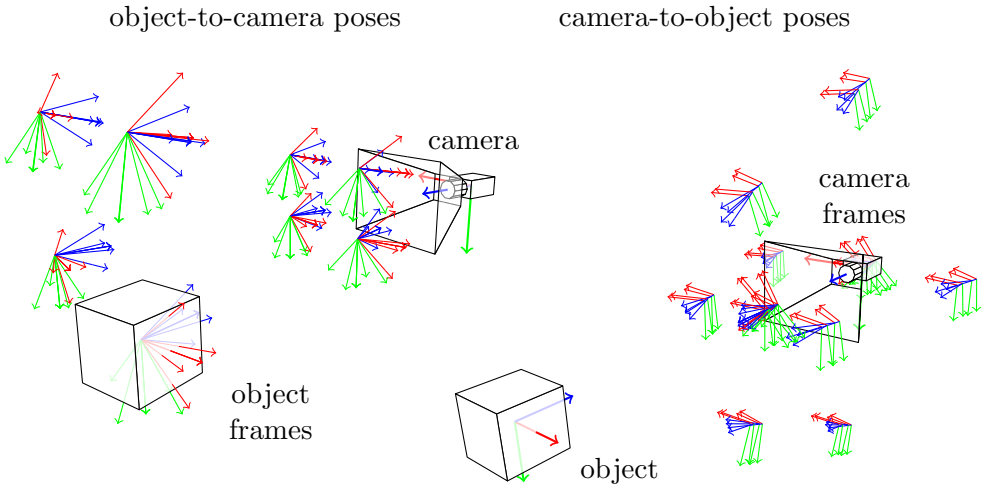

Figure 3.10: Poses for the acquisition of 56 test images. Left: object-to-camera poses. Right: camera-to-object poses.

3.4.3 Properties of the Objective Function and the Minimisation Methods

The objective function 3.35 is inspected for one camera-to-object configuration (see figure 3.11) prior to the identification of radius and speed of convergence. Here, the box object is chosen, with its texture exhibiting high frequency components. This property is reflected in the objective function 3.35 by a narrow valley and fast modulations apart from the global minimum (see figure 3.11). In a real-world scenario, initial pose estimates may differ from the ground-truth in

more than two parameters. Thus, the objective function is expected to actually vary more rapidly than suggested by figure 3.11.

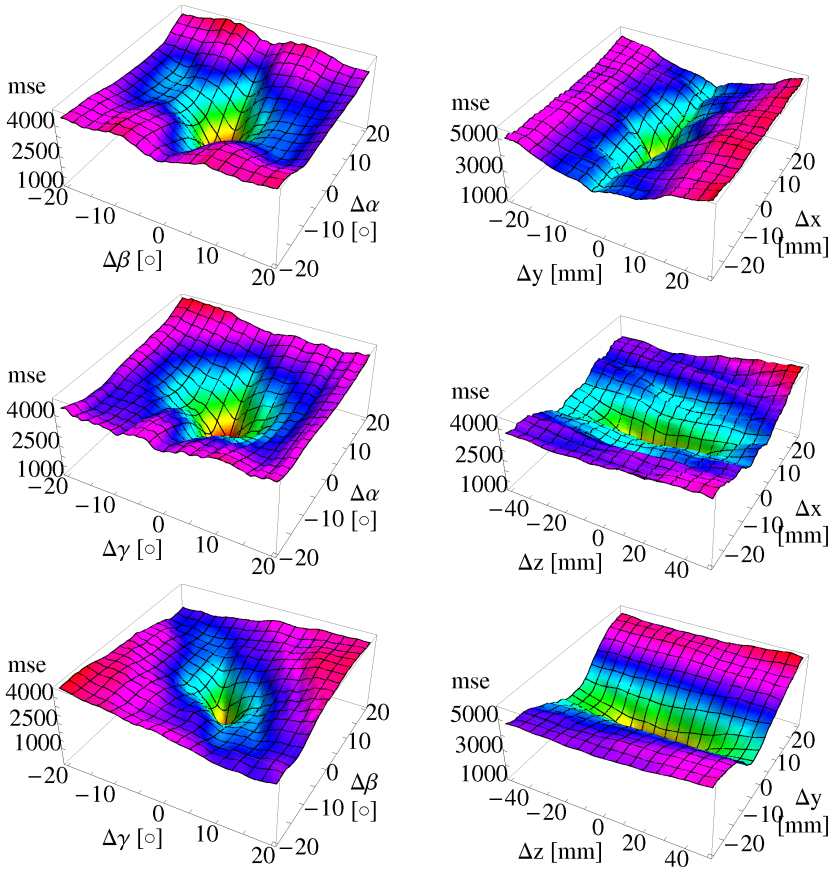

Figure 3.11: Mean squared error (MSE) for motion of the box around the reference pose (46°,−1°,−179°,2 mm,−2 mm,−416 mm). Left: rotation. Right: translation. From top to bottom: with respect to the axes x-y, x-z, and y-z.

Three methods of single-hypothesis tracking have been identified in section 3.2 that minimise the objective function 3.35 and hence, maximise the likelihood function 3.17. These methods are based on the motion Jacobian $\partial_{\delta\boldsymbol{\mu}} I_{\mathbf{x}}(\hat{\boldsymbol{\mu}} \circ \delta\boldsymbol{\mu})$ at $\delta\boldsymbol{\mu} = \mathbf{0}$, which, in turn, depends on the spatial Jacobian $\nabla I_{\mathbf{x}}(\hat{\boldsymbol{\mu}})$. Figure 3.12 illustrates the motion Jacobian showing positive (blue) and negative (red) changes on the assumed surface intensity under pose variation. The arrows indicate the expected optical flow for the surface point projections under pose variation.

Differences between the single-hypothesis methods are best seen in the different assumptions about the motion Jacobian and the expected optical flow. The Jacobian and the corresponding optical flow can vary significantly from view to view, as exemplified in the left and central pictures of figure 3.13. Accordingly, the assumption of a constant Jacobian (cf. section 3.2.5) leads to significant local errors in these cases. By contrast, the prediction of the Jacobian (cf. section 3.2.4) shows negligible differences w.r.t. the true Jacobian

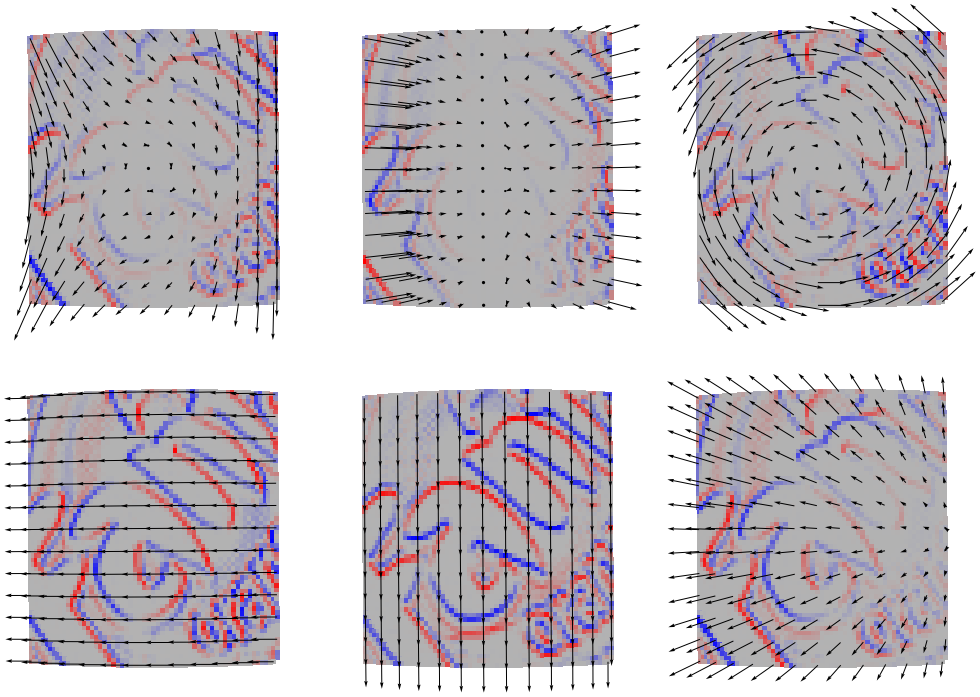

Figure 3.12: Motion Jacobians at the reference pose for rotations (top) and translation (bottom) with respect to the x, y, and z axis (from left to right). Colours illustrate positive (blue) and negative (red) changes while arrows indicate the optical flow for selected model points.

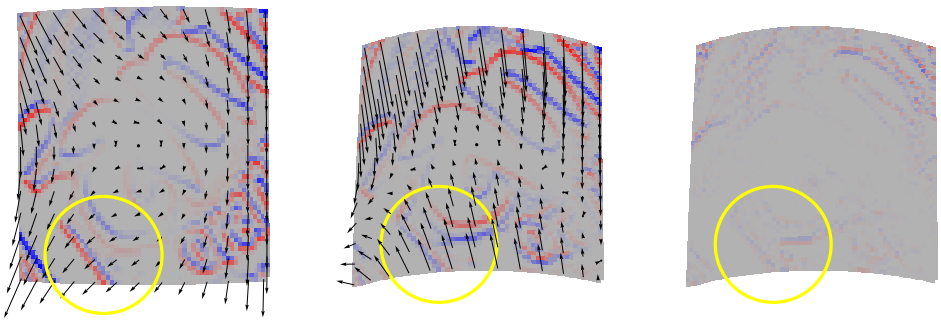

Figure 3.13: Motion Jacobians for rotation around the x-axis computed at the reference pose (left) and at an alternative pose (middle), and the residual between the true Jacobian and its prediction (right).

(compare central and right picture of figure 3.13).

The convergence speed for single-hypothesis methods as well as for the presented multi-hypotheses method does not only depend on the correctness of the underlying assumptions, but also on the computational costs associated with a single iteration of the optimisation method. Table 3.3 reports the averaged time measurements for each method. The computational advantage of predicting the

motion Jacobian (GN-IC, cf. section 3.2.4) instead of computing the derivatives for the current image (GN, cf. section 3.2.1) becomes evident. Further computational improvements are possible assuming a constant Jacobian (GN-IC-R, cf. section 3.2.5). The measurements also show the computational complexity of the multi-hypotheses approach (MCMC, cf. section 3.3.2) for fully-sampled surface models.

	P-4 2.4 GHz	Xeon 2.8 GHz	P-D 2.8 GHz	Core2 2.4 GHz
GN	4675	3785	3608	2389
GN-IC	1100	954	795	515
GN-IC-R	776	661	668	414
MCMC	528851	477657	433846	310575

Table 3.3: Computational costs (time [μs]/iteration) for single-hypothesis (GN, GN-IC, GN-IC-R) and multi-hypotheses (MCMC) approaches. The measurements are performed on Intel-CPU based platforms consistently with 3720 surface model points and 1000 particles in case of the multi-hypotheses approach.

3.4.4 Convergence Properties of the Estimator

Although the methods presented in this chapter aim at tracking objects in consecutive images, no image sequence is actually grabbed for the experiments in this section. Indeed, the ability to track the object pose through a sequence of images without any assumptions on object motion is determined effectively by the region and speed of convergence.

These properties are determined empirically for all of the identified single-hypothesis and multi-hypotheses methods with respect to 56 different viewing poses. For all views, first estimates of the ground-truth object-to-camera poses are obtained with a 6-DoF registration of the object and the camera in the reference view, and the known poses of the robot. The estimates are refined by maximising the log-likelihood of equation 3.18 with the standard Gauss-Newton (GN) approach for the textured model obtained from the reference view.

The convergence behaviour of the methods is identified by repeatedly performing the estimation procedure for increasing offsets in rotation and translation with respect to the ground-truth pose. The approach differs from a Monte-Carlo simulation by the systematical variation of the offsets in order to gain specific information about both the speed and range of convergence.

In particular, three offsets are considered: an offset in rotation along an arbitrary axis of rotation, an offset in translation in the x/y-plane, and an offset in translation along the camera z-axis. The initial object-to-camera pose is set up as a combination of a specific offset in one of the mentioned components and normally distributed errors in the other respective components.

Figures 3.14 and 3.15 report the range of convergence obtained for the bottle, box, and sculpture object. In addition, figure 3.14 exemplifies the speed of convergence for the former object. For each of the 56 views, 50 samples are randomly drawn for each specific offset in rotation or translation. Hence, both

the probability of convergence and the speed of convergence are estimated from as many as 2800 trials for each specific offset. The former statistics expresses the relative frequency of the final estimate lying within a certain ellipsoid around the ground-truth pose. The latter statistics reports the average number of iterations employed by the approach to reach this particular accuracy.

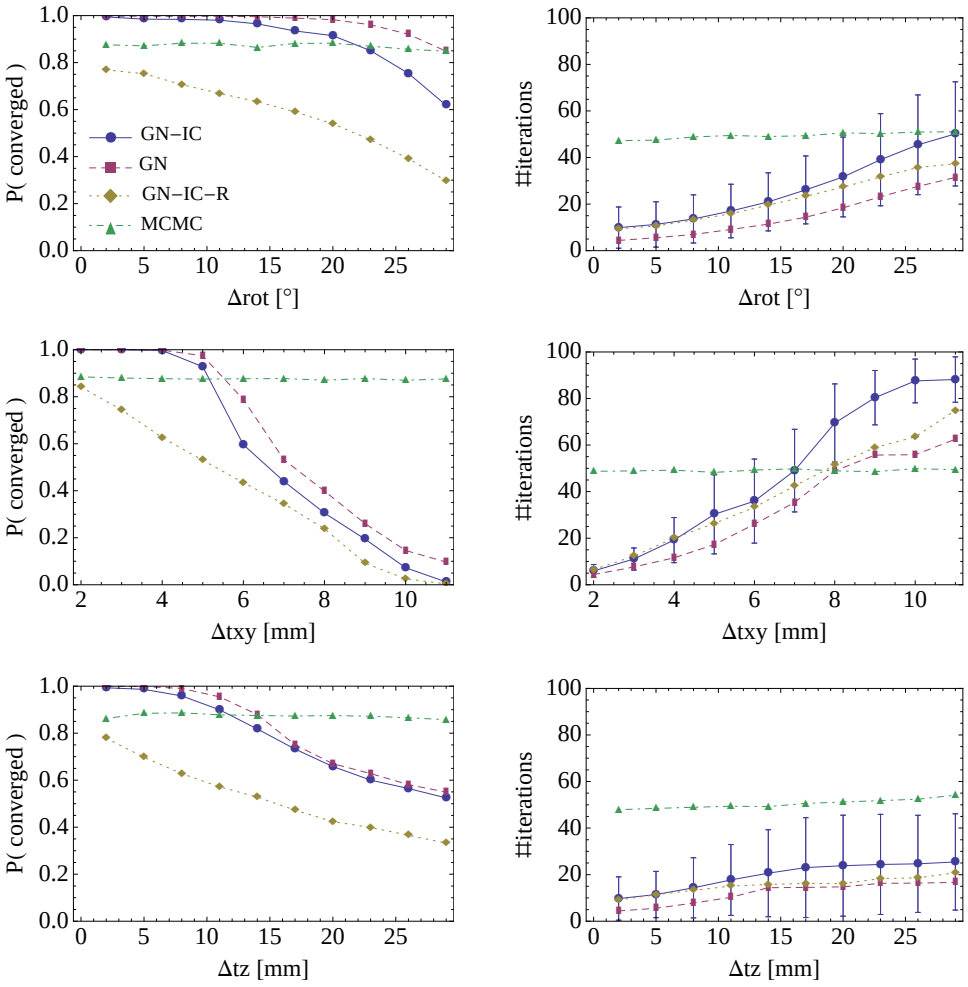

Figure 3.14: Convergence performance for the bottle object. Left: Convergence probability as a function of the initial pose offsets. Right: Speed of convergence measured in iterations.

In these experiments, 100 iterations are allotted to everyone of the single-hypothesis methods. However, multi-hypotheses tracking by means of the Markov-Chain Monte-Carlo algorithm is optimised with regard to execution time in order to equalise the computational resources spent for each trial. Accordingly, 100 iterations of the annealed Markov-Chain Monte-Carlo method are used for the bottle and sculpture objects, while 140 iterations are allotted for the box object. In order to reduce computational time, the surface point-models are sub-sampled by a factor of 40-50 and the number of particles is limited to 300. A convergence threshold higher than the one used for

the single-hypothesis methods is granted to this approach due to its stochastic nature.

The evaluation of the experiments reported in figures 3.14 and 3.15 show that the GN-IC algorithm outperforms the GN-IC-R algorithm in terms of convergence radius. The non-optimised Gauss-Newton (GN) approach exhibits typically an even bigger range of convergence, because it considers the current image for the computation of the Jacobian. The Markov-Chain Monte-Carlo algorithm (MCMC) shows the biggest range of convergence toward the correct pose. If the performance is measured in number of iterations needed for convergence, then the non-optimised Gauss-Newton (GN) approach outperforms all other algorithms.

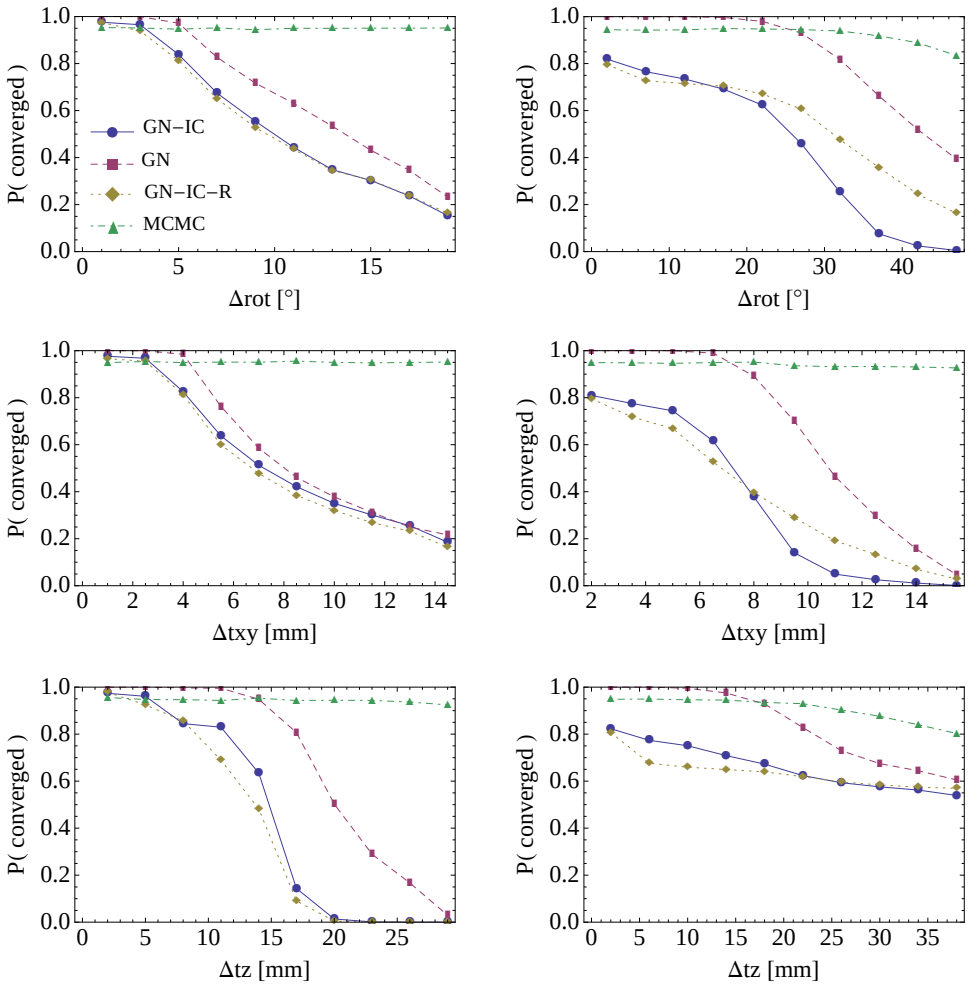

Figure 3.15: Convergence probability as a function of the initial pose offsets for the box object (left) and the sculpture object (right).

The evaluation, however, does not reflect equal estimation accuracies because of the particular threshold granted to the Markov-Chain Monte-Carlo algorithm. This method exhibits higher inaccuracies in the final estimates compared to the single-hypothesis methods as exemplified in figure 3.16. In

the diagrams, the pose error is equal to $0.5\left(\frac{\Delta r\,[^\circ]}{2[^\circ]}\right)^2 + 0.5\left(\frac{\Delta t\,[\mathrm{mm}]}{2[\mathrm{mm}]}\right)^2$, whereas Δt denotes the translation vector and Δr specifies the rotation angle of the angle axis representation of the differential transformation.

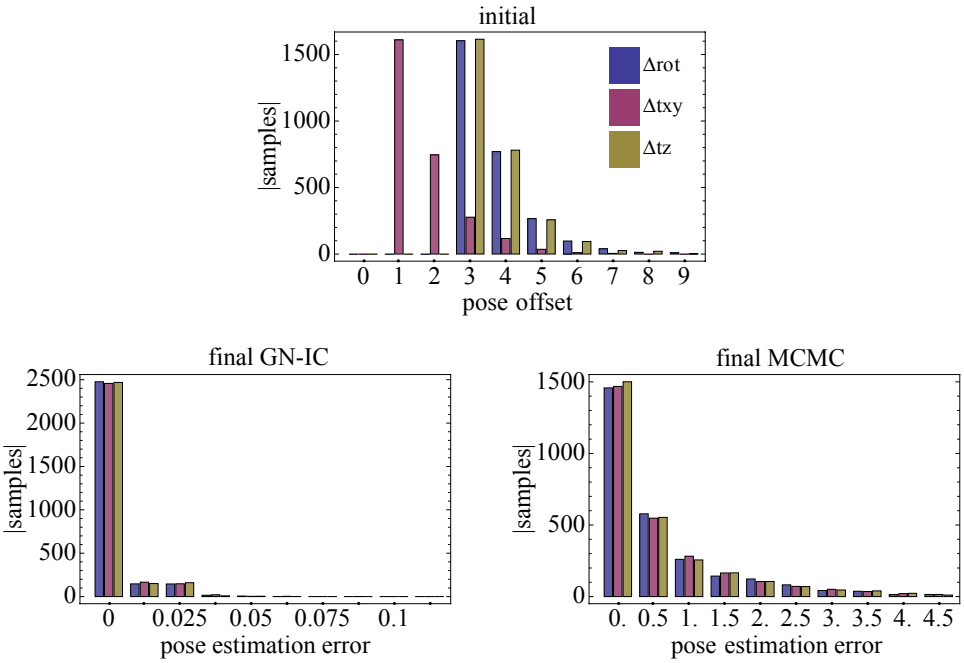

Figure 3.16: Distribution of pose errors for the bottle object. Initial distribution (top), final estimation accuracy for the Gauss-Newton algorithm (lower left), and final accuracies for the annealed Monte-Carlo method (lower right). Note the different scalings on the abscissas. See text for further details.

The ranking of convergence performance changes significantly if the rotational and translational offsets are re-interpreted with respect to the individual processing times. Accordingly, the offsets represent displacements obtained for specific object velocities within the processing period allotted to a particular method. The convergence probabilities for the respective object velocities are shown in figures 3.17 and 3.18. In these figures, the processing times are calculated for 100 (140) iterations of the methods on a Pentium 4 processor at 2.4 GHz (see table 3.3). Tracking with the image-constancy assumption (GN-IC) and the relaxed image-constancy assumption (GN-IC-R) outperform the other methods with respect to the supported speed of object motion in the cases of the bottle and box objects. In terms of the average amount of time needed for convergence to the right pose, tracking with the constant Jacobian (GN-IC-R) yields the best results (see figure 3.17).

In spite of the increased robustness obtained by the prediction of the Jacobian (GN-IC) with respect to the object velocity for the bottle and the box, the experiments reveal a decreased performance for slow motion of the sculpture object (figure 3.18). The loss in performance originates from the problems of the algorithm to converge to the true pose for selected views of the setup (figure 3.15). These failures result from violations of shape and texture assump-

tions. The shape of the sculpture object is not perfectly known but measured. Local inaccuracies can lead to significant deviations from the image-constancy assumption in 3-d for out-of-plane rotations of the object with respect to the camera. The resulting error affects the method twice, namely by the prediction of the motion Jacobian and by non-vanishing residuals of the objective function. In addition, the surface of the sculpture violates the assumption on pure Lambertian, i.e., diffuse, reflection, which accounts for additional deviations from the image-constancy assumption.

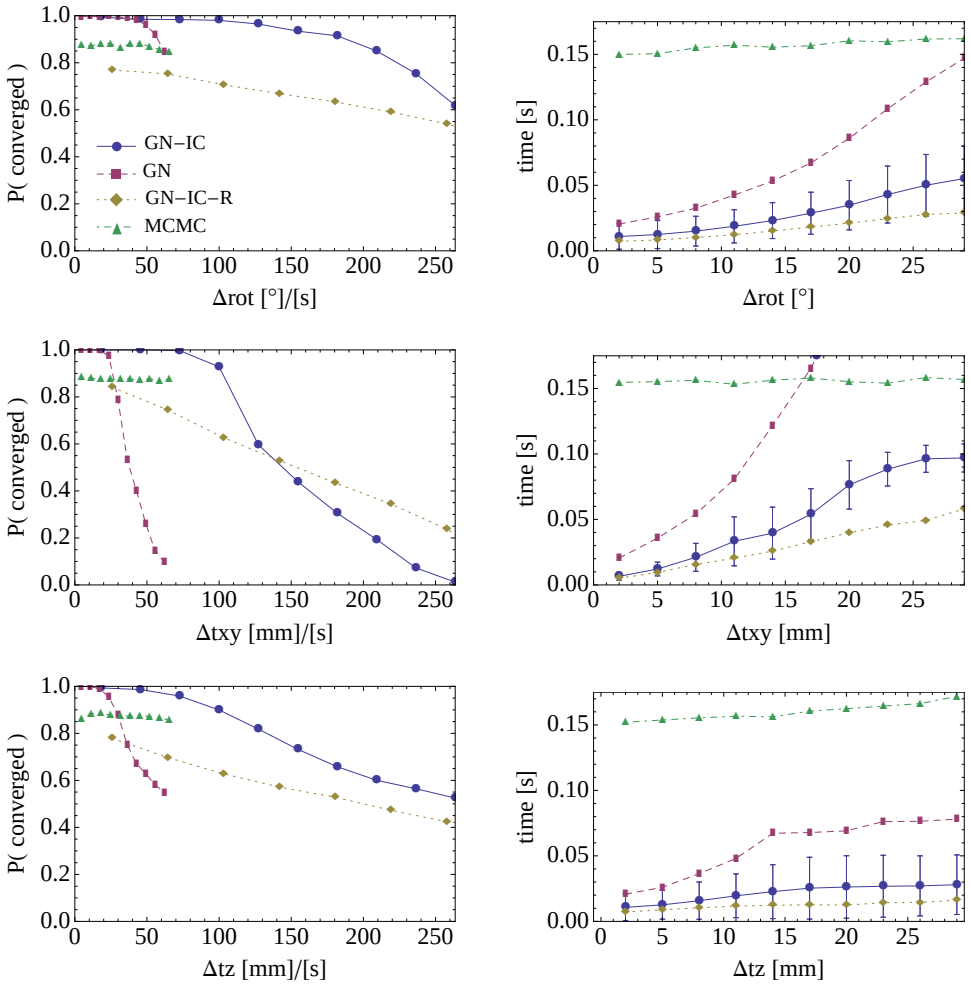

Figure 3.17: Real-time convergence performance for the bottle object on a P-4 2.4 GHz computer. Left: Convergence probability with respect to object velocity. Right: Speed of convergence measured in seconds.

In conclusion, the single-hypothesis methods based on the image-constancy assumption (GN-IC) usually outperform the standard approach (GN) in the most important category, namely the radius of convergence under real-time constraints. Theoretically, the annealed Monte-Carlo method (MCMC) can achieve unlimited radius of convergence if the computational costs are neglected. In practice, however, the resources are limited and both the number of samples

and the number of surface model points have to be reduced significantly. Hence, the method does not obtain by far an accuracy as high as the one of the single-hypothesis methods.

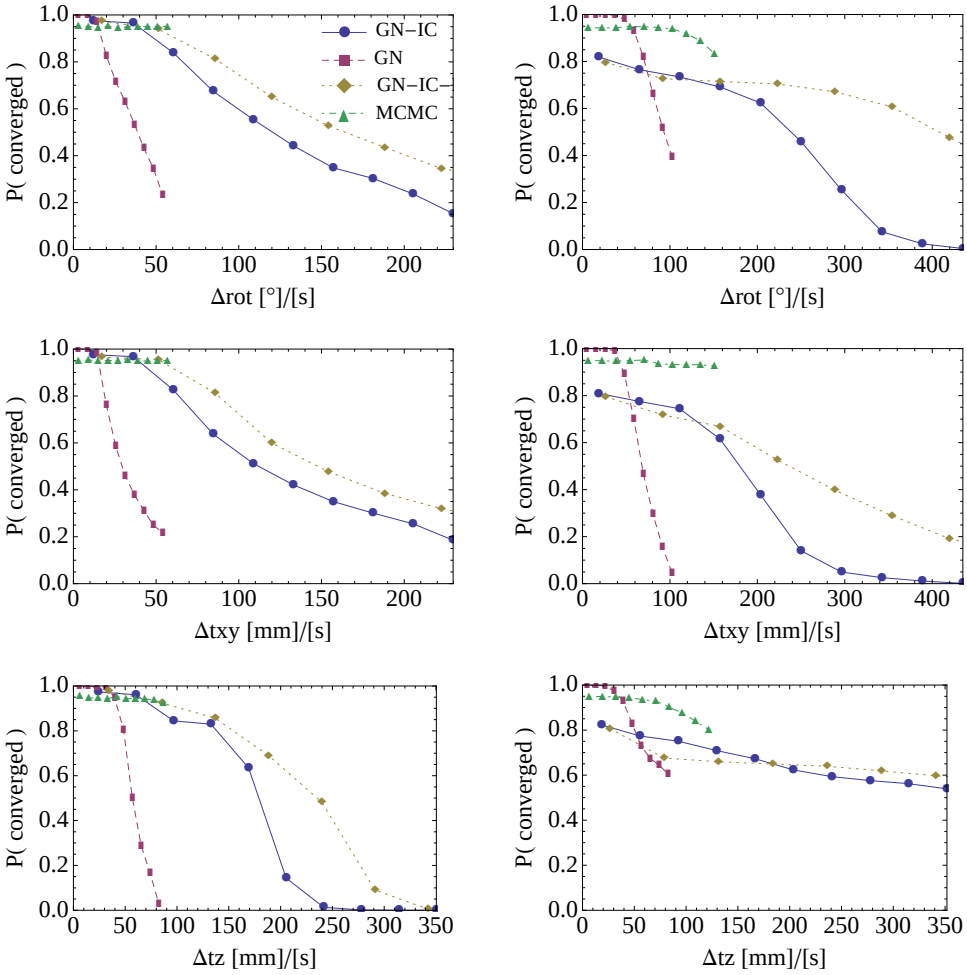

Figure 3.18: Real-time convergence probability with respect to object velocity for the box object (left) and the sculpture object (right). The tests are performed on a P-4 2.4 GHz computer.

4

Object-Luminance Adaptation

Variants and invariants play important roles in computer vision. The appearance of objects on the image plane vary with the pose of the observer, the illumination conditions, the shape of the surface, and its reflectance properties. The task of object classification, for instance, is mainly interested in the invariant properties among the class members, such as shape and reflectance properties. The task of tracking objects in three dimensions, however, aims especially at characteristics of the appearance that vary with the pose of the object. Obviously, these variations are necessary in order to estimate the object pose up to six degrees of freedom (DoF).

The consideration of all possible variations of appearance is usually prohibitive for real-time applications. Hence, it is opportune to assume particular properties to be constant, first and foremost, those attributes intrinsic to the object. These attributes comprehend the object shape, which can be assumed rigid, i.e., non deformable or articulated. Furthermore, intrinsic properties refer to the reflectance characteristics of the object surface.

The reflectance characteristics determines how the light emitted by the surface, termed object luminance or object radiance, relates to the light falling on the surface, which is termed irradiance. In general, this characteristic is a function of the direction and the power of incoming light, as well as the viewers direction as expressed by the bidirectional reflectance distribution function [102]. This function has been established by the National Bureau of Standards[1] and is not necessarily a priori known but can be assumed constant. Of course, this is not a serious restriction of the possible set of objects since the vast majority of objects do not change their reflectance over a reasonable amount of time.

Models in line with the bidirectional reflectance distribution function (BRDF) are used in computer vision as well as in computer graphics. In order to gather

[1] standardisation authority of the USA

the BRDF for a specific material, the reflectance has to be thoroughly measured for possible illumination and viewing directions. No general analytic description of the BRDF exists and thus the function is usually approximated by a basis function network. However, simplifications of the BRDF are possible for several types of materials. The most important and widely used model is the Lambertian reflectance model, which assumes perfect diffuse surfaces. In this case, the surface radiance does not depend on the viewer direction but solely on the angle of the incident light relative to the surface normal.

Given the rigidity assumption and the assumption of unknown but constant reflectance characteristics, the remaining factors for the variations in the appearance are the pose of the observer, the illumination power and direction. The changes in appearance due to a moving object or a moving camera are considered in chapter 3. In that approach, the surface radiance is thought to be independent of the object pose. This assumption is, however, only met if

1. the object has Lambertian reflectance properties and
2. the irradiance on the surface remains constant.

The latter is especially true if the object remains immobile and only the viewer moves without casting shadows on the object. The assumption of constant radiance is relaxed in the following to consider tracking of moving objects. Hence, attention is payed to the variations of the appearance due to illumination.

In the following, three methods are presented, which handle the variations in illumination [120] differently. None of these methods infers changes in object or viewers pose from these variations. Rather, they counteract the effects attributed to changes in illumination. The methods are popular in the computer-vision community and are adapted here to the shape-texture based tracking approaches. Section 4.1 applies intensity normalisation to the surface texture gathered in the current image. The next section 4.2 considers a method that determines the linear texture subspace attributed to variation of illumination conditions. Finally, section 4.3 considers the template-update approach of Matthews et al. [96], which is not tailored specifically to changes in illumination but to all kinds of appearance changes.

4.1 Texture Normalisation

Generally, surface radiance is a non-linear function of light direction even in the case of Lambertian surfaces. Let $f_{\mathbf{x}}(\theta, \psi)$ be the bidirectional reflectance distribution function for a surface point $\mathbf{x}$ with the unit vector θ of the incoming light and the unit vector ψ of the reflected light (cf. [78]). The intensity ${}^{\mathrm{S}}L_{\mathbf{x}}$ measured for surface point $\mathbf{x}$ viewed from direction $\psi_{\mathbf{x}}$ is determined by

$$ {}^{\mathrm{S}}L_{\mathbf{x}} = {}^{t}c\left(\int_{\theta} f_{\mathbf{x}}(\theta, \psi_{\mathbf{x}})\, {}^{\mathrm{S}}E_{\mathbf{x}}(\theta)\, d\theta \right) \tag{4.1} $$

where ${}^{\mathrm{S}}E_{\mathbf{x}}$ denotes the direction-dependent irradiance for surface point $\mathbf{x}$, and ${}^{t}c : \mathbb{R} \rightarrow \mathbb{R}$ represents the camera transfer function. The latter reflects the

typical saturation characteristic of the photosensitive sensor cells and possibly an additional non-linear gamma correction[2]. The transfer function may also vary over time t when camera parameters, e.g., integration time (shutter) and linear amplification (gain), are not kept constant.

Usually, the purpose of automatic shutter, brightness, and gain control is to prevent the saturation of sensor cells and to guarantee good contrast in the image. In theory, this control allows for compensation of brightness changes of the tracked object. However, the control does neither operate individually on a per-pixel or per-texel basis, nor on the basis of a distinct surface. Instead, the control strategy refers to the image as a whole and hence is not suited for tracking distinct objects in the scene.

The same functionality is provided by normalisation of brightness of a selected pattern but with two important differences. First, the set of the observed pixels can be arbitrary and possibly non contiguous. Second, brightness is not adapted in a closed-loop control of shutter and gain but independently at each time instant keeping overall shutter and gain parameters constant. Hence, normalisation operates immediately without a tune-in phase but with the disadvantage that saturation or low contrast at sensor level cannot be prevented. Two normalisation methods are described in the following (cf. [58]).

4.1.1 Intensity-Distribution Normalisation

This approach analyses the first order moment and the central second order moment of the reference pattern and the current pattern for the surface $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ computed respectively by

$$E(\mathrm{I}) = \frac{1}{N} \sum_{\mathbf{x} \in X} I_{\mathbf{x}} \quad \text{and} \quad D^2(\mathrm{I}) = \frac{1}{N} \sum_{\mathbf{x} \in X} \left(I_{\mathbf{x}} - E(\mathrm{I})\right)^2 \tag{4.2}$$

for the surface texture $\mathrm{I} = (I_{\mathbf{x}_1}, I_{\mathbf{x}_2}, \ldots, I_{\mathbf{x}_N})$. Either the reference texture, the current texture, or both of them are transformed so that subsequently all textures coincide in their mean (first order moment) and standard deviation (root of central second order moment).

The rule can be easily integrated into the shape-texture tracking methods of section 3.1.3 thanks to the linearity of the transformation. Let ${}^t\mathrm{I}(\boldsymbol{\mu}) = \left({}^tI_{\mathbf{x}_1}(\boldsymbol{\mu}), {}^tI_{\mathbf{x}_2}(\boldsymbol{\mu}), \ldots, {}^tI_{\mathbf{x}_N}(\boldsymbol{\mu})\right)$ represent the vector of texture values for the image ${}^tI$ and the pose $\boldsymbol{\mu}$ at time t computed according to equation 3.15. The augmented log-likelihood function with the transformation of the current texture then reads for the object pose $\boldsymbol{\mu}$

$$\begin{aligned} \ln {}^tL(\boldsymbol{\mu}) &= -|X| \ln \sqrt{2\pi}\sigma \\ &\quad -\frac{1}{2\sigma^2} \sum_{\mathbf{x} \in X} \left(\frac{D\left({}^0\mathrm{I}({}^0\boldsymbol{\mu})\right)}{D({}^t\mathrm{I}(\boldsymbol{\mu}))} \left({}^tI_{\mathbf{x}}(\boldsymbol{\mu}) - E\left({}^t\mathrm{I}(\boldsymbol{\mu})\right)\right) - \left({}^0I_{\mathbf{x}}({}^0\boldsymbol{\mu}) - E\left({}^0\mathrm{I}({}^0\boldsymbol{\mu})\right)\right) \right)^2 . \end{aligned} \tag{4.3}$$

[2]Gamma correction has been introduced on camera level to compensate for the non-linear response of cathode ray tube (CRT) screens.

This objective function is referred to as *normalised, zero-mean SSD* since it considers both scaling of texture values and shifting the mean texture value to zero.

4.1.2 Intensity-Difference Normalisation

Intensity-distribution normalisation as described in the above section is closely related to the linear brightness adaptation of Lucas and Kanade [91] given by

$$O\big({}^t\boldsymbol{\mu}, {}^ta, {}^tb\big) = \sum_{\mathbf{x}\in X} \big({}^ta\ {}^tI_{\mathbf{x}}\big({}^t\boldsymbol{\mu}\big) + {}^tb - {}^0I_{\mathbf{x}}\big)^2 \ . \tag{4.4}$$

In contrast to the previous approach, scale and shift are considered free parameters and not bound to specific values a priori. Despite the resemblance of the parameters ${}^ta$ and ${}^tb$ to scale and shift of the normalised, zero-mean SSD, these parameters do not necessarily match the corresponding terms at the overall optimum of the above objective function. This follows from the fact that linear brightness adaptation of Lucas and Kanade considers pattern differences, while intensity-distribution normalisation considers each pattern separately. We shall hence refer to the former as intensity-difference normalisation.

In practise, for the estimation of motion, intensity-distribution normalisation is preferred over intensity-difference normalisation because of its higher robustness to badly aligned patterns.

4.2 Complementary-Subspace Mapping

In order to cope more accurately with possible intensity variation of single surface points, the physical laws of light reflection measurement of equation 4.1 are elaborated in more detail before these findings are integrated into the methods for shape-texture tracking.

4.2.1 Illumination Subspace for Lambertian Surfaces

A preliminary and widespread assumption for the simplification of the light reflection measurement models is the linearity of the camera transfer function. More precisely, the transfer function ${}^tc : \mathbb{R} \rightarrow \mathbb{R}$ is thought to correspond to the identity mapping, which leaves the reflected intensities unaltered.

The simplification of the surface radiance of equation 4.1 starts with the irradiance. The amount of light falling on a surface depends on the inclination of the surface with respect to the direction of illumination. Therefore, the surface irradiance ${}^{\mathrm{S}}E_{\mathbf{x}}$ for the surface point $\mathbf{x}$ and illumination direction θ is given by

$${}^{\mathrm{S}}E_{\mathbf{x}}(\theta) = \max\big(\mathbf{n}_{\mathbf{x}}^{\mathrm{T}}\theta, 0\big) \cdot {}^{\mathrm{A}}L_{\mathbf{x}}(\theta) \ , \tag{4.5}$$

where $\mathbf{n}_{\mathbf{x}}$ represents the surface normal and ${}^{\mathrm{A}}L_{\mathbf{x}}$ denotes the direction dependent ambient radiance towards the surface. The non-linear function max truncates incompatible illumination directions. The object irradiance is simplified considering only light sources at infinity and convex shaped objects. Hence, convex

objects do not cast any shadow on themselves, and a single surface element is illuminated by any light source it is facing. The radiance towards the object is therefore considered global and identical for each surface point, that is ${}^{\mathrm{A}}L_{\mathbf{x}}(\theta) = {}^{\mathrm{A}}L(\theta)$. Accordingly, the surface luminance ${}^{\mathrm{S}}L_{\mathbf{x}}$ for a surface point $\mathbf{x}$ measured by the camera in direction $\psi_{\mathbf{x}}$ expands to

$$ {}^{\mathrm{S}}L_{\mathbf{x}} = \int_{\theta} f_{\mathbf{x}}(\theta, \psi_{\mathbf{x}}) \cdot \max\left(\mathbf{n}_{\mathbf{x}}^{\mathrm{T}}\theta, 0\right) \cdot {}^{\mathrm{A}}L(\theta)\, d\theta \; . \tag{4.6} $$

The formula is further simplified by taking exclusively the diffuse reflection characteristic of the surface into account. For Lambertian surface points the above equation reduces to

$$ {}^{\mathrm{S}}L_{\mathbf{x}} = \int_{\theta} \rho_{\mathbf{x}} \cdot \max\left(\mathbf{n}_{\mathbf{x}}^{\mathrm{T}}\theta, 0\right) \cdot {}^{\mathrm{A}}L(\theta)\, d\theta \tag{4.7} $$

where the bidirectional reflectance distribution function $f_{\mathbf{x}}(\theta, \psi_{\mathbf{x}})$ is replaced by a single albedo value $\rho_{\mathbf{x}}$. This equation shows to be the illumination model intensively studied by Belhumeur and Kriegman [13] where the surface texture

$$ {}^{\mathrm{S}}\mathrm{L} = \int_{\theta} \max\left(\mathrm{B}\theta \cdot {}^{\mathrm{A}}L(\theta), \mathbf{0}\right) d\theta \tag{4.8} $$

is expressed in vector form for the surface points $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, whereas the product of surface albedo and normal is aggregated to the $N \times 3$ matrix

$$ \mathrm{B} = \begin{pmatrix} \rho_{\mathbf{x}_1} \mathbf{n}_{\mathbf{x}_1}^{\mathrm{T}} \\ \rho_{\mathbf{x}_2} \mathbf{n}_{\mathbf{x}_2}^{\mathrm{T}} \\ \cdots \\ \rho_{\mathbf{x}_N} \mathbf{n}_{\mathbf{x}_N}^{\mathrm{T}} \end{pmatrix} \; . \tag{4.9} $$

The set of texture images that can be generated for any possible illumination is a convex cone in $\mathbb{R}^N$ [13]. The cone is delimited by up to N extreme rays depending on the number of non-parallel surface normals. These extreme rays correspond to illumination conditions with at least one shadowed surface.

The generative formula for the surface texture becomes linear when the shadowing configurations are not considered. Let $\mathbf{s} \in \mathbb{R}^3$ denote an arbitrary, non-shadowing radiance direction and power, then the so-called illumination subspace is given by

$$ \mathcal{L} = \left\{ {}^{\mathrm{S}}\mathrm{L} | {}^{\mathrm{S}}\mathrm{L} = \mathrm{B}\mathbf{s}, \mathbf{s} \in \mathbb{R}^3 \right\} \; . \tag{4.10} $$

The rank of the subspace depends on the rank of B, which, in turn, depends on the shape of the object. Planar surfaces as well as cylindrical surfaces, for instance, do not exhibit full rank, i.e., 3, but respectively rank 1 and 2.

According to Belhumeur and Kriegman, the basis for the illumination subspace is determined by gathering images of the object for at least 3 different illumination directions that do not cast any shadows. The basis vectors are identified by the left orthonormal vectors corresponding to the non-zero singular values of B. This basis suffices to synthesise any appearances of the Lambertian surfaces not subject to shadows. Since the formulation of Equation 4.10

does not consider specular reflections or shadowed configurations, more than 3 principal components of B are taken into account. However, any increase in dimensionality affects complexity and efficiency of the involved methods.

In order to constrain the dimensionality of the illumination subspace to at most 3, exclusively Lambertian reflectance characteristics are considered in the following. A distinct approach is developed, which determines a basis for the diffuse components of reflection even in the presence of partially specular surface points. It allows for identification of the surface albedos that best describe the approximated Lambertian surface.

In contrast to the work of Belhumeur et al. [13, 62], the three-dimensional model of the object is given a priori. Obviously, this extra information helps in the determination of the basis of the illumination subspace. Let $\mathrm{M} = (\mathbf{n}_{\mathbf{x}_1}, \mathbf{n}_{\mathbf{x}_2}, .., \mathbf{n}_{\mathbf{x}_N})^{\mathrm{T}}$ be the matrix of surface normals. Then, B is defined by

$$\mathrm{B} = \mathrm{diag}(\mathbf{a})\,\mathrm{M}\,, \quad \mathbf{a} = (\rho_{\mathbf{x}_1}, \rho_{\mathbf{x}_2}, \ldots, \rho_{\mathbf{x}_N}) \tag{4.11}$$

where $\mathrm{diag}(\mathbf{a})$ represents the diagonal matrix of corresponding surface albedos. Since the normals are known a priori, the problem of finding the basis of illumination subspace is reformulated to the task of ascertaining the surface albedo and illumination direction. Given ${}^{\mathrm{i}}N$ textures $\{{}^{j}\mathrm{I} \mid j \in 1, 2, \ldots, {}^{\mathrm{i}}N\}$ of the surface under different illumination, then the problem can be formulated as a non-linear least-squares problem

$$O(\mathbf{a}, \mathrm{S}) = \sum_{j=1}^{{}^{\mathrm{i}}N} \left\| \mathrm{diag}(\mathbf{a})\,\mathrm{M}\mathbf{s}_j - {}^{j}\mathrm{I} \right\|^2 \,, \quad \mathrm{S} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{{}^{\mathrm{i}}N}) \,, \tag{4.12}$$

which has to be solved for the vector of surface albedos $\mathbf{a}$ and the unknown illumination directions S. The task can be accomplished by iteratively solving for the illumination directions and the surface albedos as described by the algorithm in Figure 4.1. The basis of the illumination subspace is finally determined by the left orthogonal vectors $\mathrm{U}_{\mathrm{B}}^{\mathrm{T}}$ for the non-zero singular values of B, obtained through a singular value decomposition

$$\mathrm{B} = \mathrm{U}_{\mathrm{B}} \cdot \Sigma_{\mathrm{B}} \cdot \mathrm{V}_{\mathrm{B}}^{\mathrm{T}} \,. \tag{4.16}$$

According to the definition of singular value decomposition, the diagonal matrix Σ_{B} contains the singular values and $(\mathrm{U}_{\mathrm{B}})^{\mathrm{T}}(\mathrm{U}_{\mathrm{B}}) = \mathbb{1}$, $(\mathrm{V}_{\mathrm{B}})^{\mathrm{T}}(\mathrm{V}_{\mathrm{B}}) = \mathbb{1}$.

4.2.2 Shape-Texture Tracking in the Complementary Subspace

The integration of the illumination subspace into the approaches of shape-texture tracking presented in chapter 3 implicates the substitution of the static reference texture with an illumination-dependent texture. If any shadowing effects are excluded or neglected, then any linear combination within the illumination subspace produces a valid surface texture. Accordingly, the log-likelihood function is augmented by the unknown overall illumination vector $\mathbf{s}$ to

$$\ln L(\boldsymbol{\mu}, \mathbf{s}) = -|X| \ln \sqrt{2\pi}\sigma - \frac{1}{2\sigma^2} \left\| \mathrm{I}(\boldsymbol{\mu}) - \mathrm{B}\mathbf{s} \right\|^2 \,, \tag{4.17}$$

Capture ${}^{\mathrm{i}}N$ images and sample corresponding surface textures $\left\{{}^{1}\mathrm{I}, {}^{2}\mathrm{I}, \ldots, {}^{{}^{\mathrm{i}}N}\mathrm{I}\right\}$

Set initial albedo $\mathbf{a} \leftarrow {}^{1}\mathrm{I}$

Do

Determine illumination direction and power $\mathrm{S} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N)$

$$\mathbf{s}_j \leftarrow \underset{\mathbf{s}_j}{\operatorname{argmin}} \left\| \operatorname{diag}(\mathbf{a})\, \mathrm{M}\mathbf{s}_j - {}^{j}\mathrm{I} \right\|^2 \tag{4.13}$$

Determine surface albedo $\mathbf{a} = (\rho_{\mathbf{x}_1}, \rho_{\mathbf{x}_2}, \ldots, \rho_{\mathbf{x}_N})$

$$\rho_{\mathbf{x}} \leftarrow \underset{\rho_{\mathbf{x}}}{\operatorname{argmin}} \sum_j \left(\rho_{\mathbf{x}} \left(\mathbf{n}_{\mathbf{x}}^{\mathrm{T}} \mathbf{s}_j \right) - {}^{j}\mathrm{I}_{\mathbf{x}} \right)^2 \tag{4.14}$$

Re-weight surface albedo

$$\mathbf{a} \leftarrow \frac{\|{}^{1}\mathrm{I}\|}{\|\mathbf{a}\|}\mathbf{a} \tag{4.15}$$

Until convergence of $O(\mathbf{a}, \mathrm{S})$

Figure 4.1: Iterative minimisation of the objective function 4.12 with respect to illumination and surface albedo.

whereas $\mathrm{I}(\boldsymbol{\mu}) = (I_{\mathbf{x}_1}(\boldsymbol{\mu}), I_{\mathbf{x}_2}(\boldsymbol{\mu}), \ldots, I_{\mathbf{x}_N}(\boldsymbol{\mu}))$ represents the vector of texture values for the image I and the pose $\boldsymbol{\mu}$ computed according to equation 3.15. Hager and Belhumeur [62] as well as Baker and Matthews [7] showed that the dependency on the parameters of the linear subspace can be removed from the sum-of-least-squares formulation if the explicit value of the parameters is not requested. The corresponding objective function to be minimised for the unknown pose $\boldsymbol{\mu}$ is constrained by the implicit texture solution in the illumination subspace and corresponds to

$$O(\boldsymbol{\mu}) = \left\| \left(\mathbb{1} - \mathrm{B} \left(\mathrm{B}^{\mathrm{T}} \mathrm{B} \right)^{-1} \mathrm{B}^{\mathrm{T}} \right) \mathrm{I}(\boldsymbol{\mu}) \right\|^2 . \tag{4.18}$$

In the following, the objective function is further simplified to

$$O(\boldsymbol{\mu}) = \left\| \left(\mathbb{1} - \mathrm{U}_{\mathrm{B}} \mathrm{U}_{\mathrm{B}}^{\mathrm{T}} \right) \mathrm{I}(\boldsymbol{\mu}) \right\|^2 \tag{4.19}$$

using the singular value decomposition of $\mathrm{B} = \mathrm{U}_{\mathrm{B}} \cdot \Sigma_{\mathrm{B}} \cdot \mathrm{V}_{\mathrm{B}}^{\mathrm{T}}$. According to the above equation, the squared Euclidean norm is only considered for the components of the surface texture *orthogonal* to the illumination subspace. Note that the reference texture used by Hager and Belhumeur [62] an Baker and Matthews [7] is safely ignored in this formulation since it is supposed to lie completely in the illumination subspace.

The generation of an observation probability density function corresponding to the (log-likelihood) objective function 4.19 is straight forward and allows for shape-texture tracking with the annealed Monte-Carlo Method of section 3.3. Tracking by local optimisation as described in section 3.2, however, deserves further consideration.

Here, a pose estimate $\hat{\boldsymbol{\mu}}^\star$ is iteratively refined by Newton's algorithm through the repeated solution of a linear equation system for the pose variation $\delta\boldsymbol{\mu}$. In case of the illumination subspace method, the equation system reads at the current estimate $\hat{\boldsymbol{\mu}}$

$$\partial_{\delta\boldsymbol{\mu}}\mathrm{I}^{\mathrm{T}} \cdot \left(\mathbb{1} - \mathrm{U_B}\mathrm{U_B^T}\right) \cdot \partial_{\delta\boldsymbol{\mu}}\mathrm{I}\big|_{\hat{\boldsymbol{\mu}}}\, \hat{\delta\boldsymbol{\mu}} = \partial_{\delta\boldsymbol{\mu}}\mathrm{I}^{\mathrm{T}}\big|_{\hat{\boldsymbol{\mu}}} \left(\mathbb{1} - \mathrm{U_B}\mathrm{U_B^T}\right)^{\mathrm{T}} \mathrm{I}|_{\hat{\boldsymbol{\mu}}} \tag{4.20}$$

taking advantage of the idempotence of the projector, i.e.,

$$\left(\mathbb{1} - \mathrm{U_B}\mathrm{U_B^T}\right)^{\mathrm{T}} \left(\mathbb{1} - \mathrm{U_B}\mathrm{U_B^T}\right) = \left(\mathbb{1} - \mathrm{U_B}\mathrm{U_B^T}\right) \ . \tag{4.21}$$

Recall that the efficient formulation of the motion Jacobian $\partial_{\delta\boldsymbol{\mu}}\mathrm{I}$ is based on the image constancy assumption in 3-d, which is now given by

$$I_{\mathbf{y}}(\boldsymbol{\mu}) = \mathbf{b}_{\mathbf{y}}^{\mathrm{T}}\mathbf{s} \ , \quad \mathbf{y} \in \mathcal{X} \subset \mathbb{R}^3 \tag{4.22}$$

for a point $\mathbf{y}$ of the *uncountable* set of scene surface points $\mathcal{X}$ and the position dependent illumination basis $\mathbf{b}_{\mathbf{y}} \in \mathbb{R}^3$. As a consequence, the spatial Jacobian depends not only on the surface point but also on the illumination parameter $\mathbf{s}$. Obviously the illumination parameter has to be determined *beforehand* in order to predict the illumination-dependent Jacobian with the formulas of Section 3.2.4 and hence, computational complexity increases.

After all, Hager and Belhumeur [62] reported that illumination is safely ignored for the computation of a motion Jacobian. Accordingly, the illumination-dependent image-constancy assumption is also replaced here by its illumination-independent counterpart. This allows to efficiently predict the motion Jacobian with the IC algorithm of section 3.2.4. Nevertheless, the solution of the illumination subspace equation 4.20 with this pose-dependent Jacobian is substantially more expensive than the solution of the linear equation system 3.36. By contrast, the approximations employed in the IC-R algorithm of section 3.2.5 can keep the extra computational costs low due to the constant, pose-independent Jacobian.

4.3 Texture Update

The previous sections explored methods using explicit models of illumination and reflection. Generally, their expressive power depends on the degree of detail of the model, which in turn is limited by the available amount of computational resources. Typically, models of reduced complexity are employed, introducing constraints on the light, the object, or the camera in order to meet the demands of real-time applications. The success of the resulting algorithms depends on the degree of correspondence between the real scenery and the combined illumination-reflection model.

In the following approach, illumination is not considered as an isolated phenomenon. Instead, all sources of appearance changes are handled implicitly by updating the appearance over time. In this way, any variation of appearance is addressed, which originate, for instance, from varying illumination or even non-constant reflection properties.

This strategy has a long history in two-dimensional tracking, which requires the compensation of both appearance changes under variation of the camera perspective and appearance changes due to deformation in shape. In this tracking scenario, the object model is represented by a two-dimensional image patch that should be tracked over a sequence of images. By *updating* the reference patch to the current two-dimensional appearance of the object, the method accounts for any potential changes.

Of course, the rate of appearance changes that can be handled is limited. In practise, however, this limitation does not represent a handicap. In fact, the major drawback of the approach is that tracking is subject to *drift* due to slight misregistrations between the reference patch and the current patch. These inaccuracies accumulate over time causing increasing shifts between the initial reference pattern and the continuously updated reference pattern.

4.3.1 Template-Update Method

Recently, Matthews et al. [96] presented a simple solution to the drift problem. Still, the objective function in question is the sum-of-squared differences between the pixel values of the reference image and the pixel values in the current image. Adapted to pose estimation with shape-texture models, the corresponding objective function reads

$$O({}^t\boldsymbol{\mu}) = \left\| {}^t\mathrm{I}({}^t\boldsymbol{\mu}) - {}^0\mathrm{T} \right\|^2 , \tag{4.23}$$

with the N-dimensional reference texture ${}^0\mathrm{T}$ gathered from the reference image with object pose ${}^0\boldsymbol{\mu}$ and the texture ${}^t\mathrm{I}({}^t\boldsymbol{\mu})$ sampled from the current image given the pose hypothesis ${}^t\boldsymbol{\mu}$.

Matthews et al. do not alter the objective function. Instead, they consider it within the context of local minimisation. First- and second-order minimisation techniques tend to reach the minimum closest to the previous pose estimate. Hence, the found minimum depends on the previous estimate and the shape of the objective function.

It can be observed that the more the appearance of the target differs from the reference, the smaller the convergence area of the minimum becomes. To counteract this effect, a reference pattern is used that better reflects the current appearance. Matthews et al. combined the benefits of both the large convergence area of an up-to-date reference pattern and the ground truth represented by the initial reference pattern. At the first step, the current image is matched against the updated pattern. Subsequently, this pose estimate is used as a starting value for the local search of the best match to the original reference pattern.

Formally, let ${}^t\mathrm{T}$ be the up-to-date reference texture at time instant t. Given the current image ${}^tI$ and pose estimate ${}^t\boldsymbol{\mu}$, the outcome of a minimisation Λ

consists in finding a pose variation

$$ {}^{t}\hat{\delta\boldsymbol{\mu}} = \Lambda\big({}^{t}I, {}^{t}\boldsymbol{\mu}, {}^{t}\mathrm{T}\big) \tag{4.24} $$

which locally minimises the objective function $O(\delta\boldsymbol{\mu}) = \big\|{}^{t}\mathrm{I}\big({}^{t}\boldsymbol{\mu} \circ {}^{t}\delta\boldsymbol{\mu}\big) - {}^{t}\mathrm{T}\big\|^2$ for the current reference pattern ${}^{t}\mathrm{T}$. The new pose estimate is used to minimise the objective function 4.23 with respect to the initial reference pattern, formally given by

$$ {}^{t}\hat{\delta\boldsymbol{\mu}}^{\star} = \Lambda\Big({}^{t}I, {}^{t}\boldsymbol{\mu} \circ {}^{t}\hat{\delta\boldsymbol{\mu}}, {}^{0}\mathrm{T}\Big) \ . \tag{4.25} $$

The final pose estimate corresponds to the combination of the initial pose estimate, the pose variation for the updated reference texture, and the pose variation for the initial reference texture, which hence reads

$$ {}^{t}\hat{\boldsymbol{\mu}}^{\star} = {}^{t}\boldsymbol{\mu} \circ {}^{t}\hat{\delta\boldsymbol{\mu}} \circ {}^{t}\hat{\delta\boldsymbol{\mu}}^{\star} \ . \tag{4.26} $$

An additional innovation of Matthews et al. concerns the criteria for updating the reference texture. Here, the template is updated only when minimisation with the initial texture and minimisation with updated texture reach the same optimum. In the contrary case, the template is not altered, and therefore

$$ {}^{t+1}\mathrm{T} = \begin{cases} {}^{t}\mathrm{I}({}^{t}\hat{\boldsymbol{\mu}}^{\star}) & : \quad O_3\Big({}^{t}\hat{\delta\boldsymbol{\mu}}^{\star}\Big) < \epsilon \\ {}^{t}\mathrm{T} & : \quad \text{else} \end{cases} \tag{4.27} $$

where ϵ represents negligible pose variations rated by an appropriate error function $O_3 : \mathrm{SE}(3) \rightarrow \mathbb{R}^+$ on the special Euclidean group SE(3).

4.3.2 Shape-Texture Tracking with Texture Update

The above template-update strategy aims explicitely at sequential minimisation approaches. Hence, the strategy naturally fits with the approaches of single pose-hypothesis tracking adopting local optimisation techniques as proposed in section 3.2.1.

Stochastical sampling approaches, however, are not well suited to the idea of template update as proposed. Multi-hypotheses tracking based on Markov-Chain Monte-Carlo techniques as described in section 3.3 are designed to approximate the posterior pose probability density for a given reference texture. A possibility to integrate a template update in this approach, consists of sampling the probability density separately for an updated texture and the reference texture. Then, the texture would be updated if the difference between the final estimates on both densities is small enough. In practice, however, the estimates might not be accurate and therefore no update would ever be performed.

4.4 Evaluation

In the following, the above concepts are evaluated with respect to their suitability for tracking moving objects. More precisely, the methods of intensity-distribution normalisation, complementary illumination subspace, and the template-update method are confronted with sequential single-hypothesis tracking without any adaptation to illumination changes.

4.4.1 Model Acquisition and Model Representation

The experiments are performed hereafter on a standard Pentium Xeon 1.7 GHz. Images are captured with an analog interlaced camera at a resolution of 768 × 576 pixels (PAL) and a horizontal aperture of 56° and vertical aperture of 48°. The internal parameters of the camera together with the distortion coefficients for a 3rd degree polynomial distortion model are determined offline.

The test set consists of two objects, a bottle and a sculpture as in section 3.4.1. Here, a slightly broader segment of 83.17° of the soda label is chosen, which allows to catch a wider range of illumination changes for the same time instant. Sampling of the cylindrical body of radius 0.046 m leads to 4624 three-dimensional surface points. The sculpture object, instead, is digitised with a hand-guided rotating laser-scanner, which poses are captured in turn by a passive robotic manipulator [135, 19]. Accordingly, 3668 three-dimensional surface points are acquired for the face of the sculpture.

The registration with the reference as well as with illumination-dependent images is achieved as in section 3.4.1 either manually, or automatically via the external pose sensor of the hand-guided scanning device. The illumination dependent textures extracted from these images (see Figure 4.1 and Figure 4.2 for an excerpt) are used to determine the illumination base. In detail, the

Table 4.1: Sample set of images for building the illumination subspace for the bottle surface.

Table 4.2: Sample set of images for building the illumination subspace for the sculpture surface. The illumination direction is changed only slightly not to produce self shadows.

algorithm of Table 4.1 is applied for estimation of the surface albedos. Subsequently, these albedos are used to determine two illumination basis textures for the bottle surface and three illumination basis textures for the sculpture surface by means of a singular value decomposition (see Figures 4.2 and 4.3).

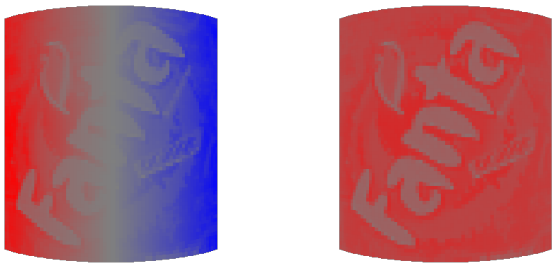

Figure 4.2: Basis of the illumination subspace for the bottle surface.

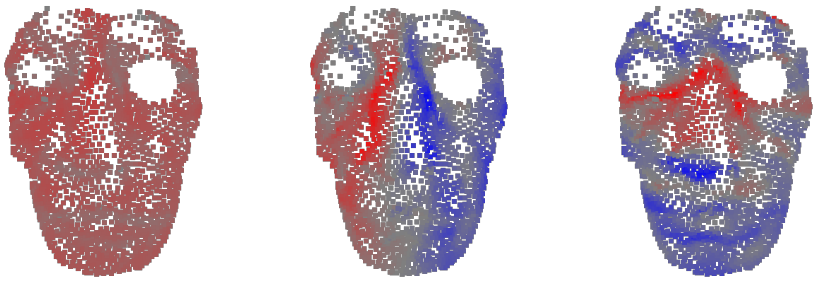

Figure 4.3: Basis of the illumination subspace for the sculpture surface.

4.4.2 Data Acquisition

In order to compare the methods for handling changes in object luminance, the object is required to move freely in three-dimensional space. Hence, three image streams are recorded for each of the objects, i.e., the bottle and sculpture object, moving under different conditions of illumination. The object trajectories start in the vicinity of the reference pose with respect to the camera.

Ground-truth trajectories for both objects are obtained in the first place through shape-texture based registration in the corresponding video data. In detail, maximum likelihood estimation of the object poses is performed on the sequences of slowly moving objects by means of exhaustive single-hypothesis minimisation with the Gauss-Newton method. In the case of the bottle object, the resulting poses are further registered to the trajectories measured by the external optical tracking system smARTtrack2[3]. For this purpose, retro-reflective markers are solidly attached to the body, which was not possible for the sculpture object. The accordingly registered trajectories of the external sensor are considered ground truth.

Virtually accelerated versions of the trajectories are generated for each of the streams through sub-sampling. This procedure ensures a broad coverage of possible combinations of rotational and translational object motion between two images. For each object, all adjacent image pairs (samples) of the accelerated streams are grouped with respect to the amplitude of their rotational and

[3] http://www.ar-tracking.com

translational motion. The histogram over these groups (figure 4.4) shows that some groups of motion occur more frequently than others. In total, approximately 7600 samples are considered for the bottle object in the ranges of $[1, 5]^\circ$ and $[1.5, 5]$ mm, and approximately 6100 for the sculpture object in the ranges of $[2, 5]^\circ$ and $[3, 10]$ mm.

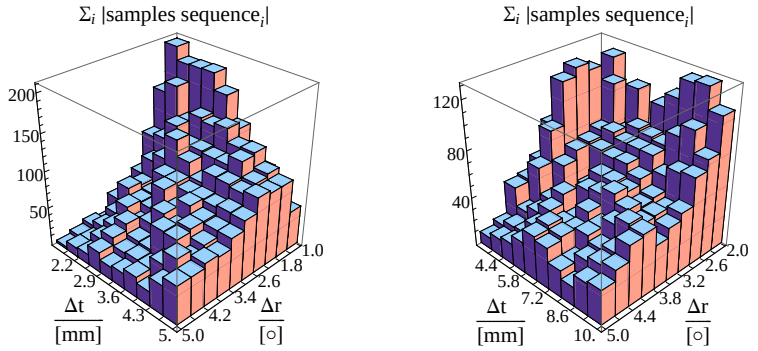

Figure 4.4: Histograms of intra-sequence motion samples taken from three sequences of the bottle object (left) and the sculpture object (right). A motion sample is characterised by the norm of its rotational (Δr) and translational (Δt) components.

4.4.3 Evaluation of Convergence Properties

The performance of the single methods for handling changes in illumination is established with respect to the probability of convergence from an initial pose to the ground-truth pose. This evaluation is only useful in comparison to the respective other methods. Hence, the performance of the method of choice, i.e., efficient Gauss-Newton based tracking with the image-constancy assumption, is opposed pairwise to the extensions devised to handle illumination changes.

In contrast to the evaluation in section 3.4, the probability of convergence is assessed for different combinations of initial rotational and translational offsets. An estimation process is considered converged when its final rotational and translational residuals to the ground truth fall below appropriate thresholds. These thresholds are 1° and 1.5 mm for the bottle object and 2° and 3 mm for the sculpture.

Figure 4.5 exemplifies the gathered information on behalf of the bottle object. Five different methods of single-pose hypothesis tracking are considered, i.e., standard Gauss-Newton minimisation (GN), Gauss-Newton minimisation with the prediction of the Jacobian based on the image-constancy assumption in 3-d (GN-IC), the complementary-subspace extension (GN-IC-CS), the template-update method (GN-IC-TU), and the modification for intensity-distribution normalisation (GN-IC-IN). For a representative comparison of these methods with respect to their real-time performance, a comparable amount of computational time is allotted to the methods. In particular, six minimisation iterations are allowed to the standard Gauss-Newton implementation and 22 iterations to all the methods based on the image-constancy assumption. The

texture-update method performs 14 iterations with an updated texture (obtained here from the starting frame) and eight iterations with the reference texture. This method is not specifically penalised, though additional computations occur when the texture is updated.

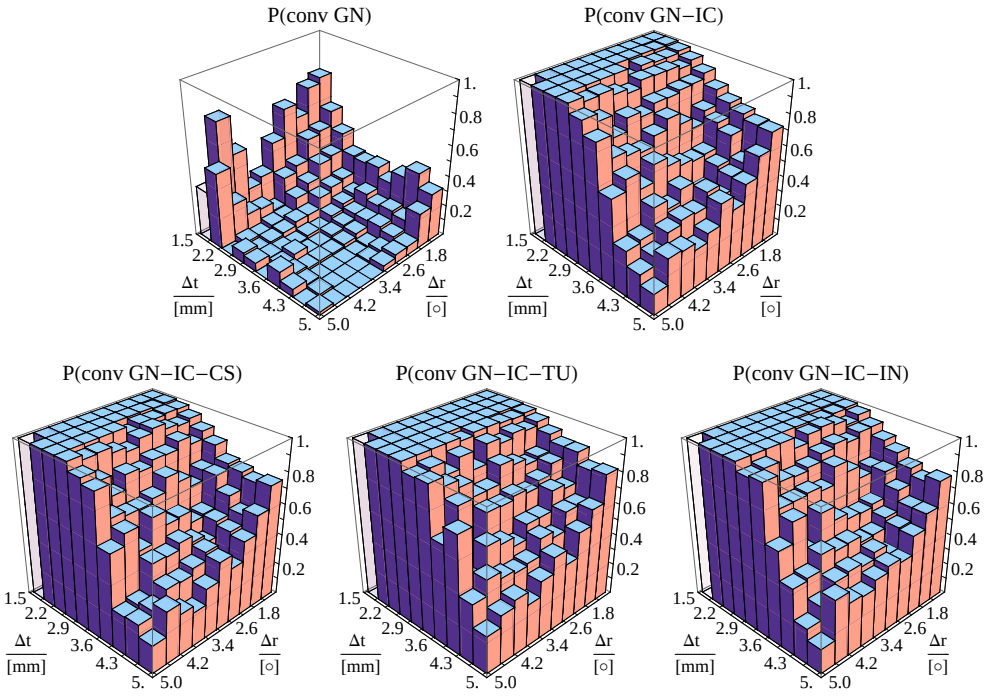

Figure 4.5: Convergence probability over combinations of initial rotational (Δr) and translational (Δt) offsets. All methods have been alloted a comparable amount of time for computation.

The convergence performances of the methods are compared in turn to the method of choice, i.e., the Gauss-Newton method with the prediction of the motion Jacobian (GN-IC). In detail, the convergence probabilities of the methods are compared to the respective probabilities of the GN-IC method. Accordingly, each method exhibits either a gain or a loss in convergence probability with respect to the particular method for each combination of rotation and translation. The histogram over all gains in the considered ranges of rotation and translation offsets as shown in figure 4.6 discloses the relative performances.

Clearly, the standard Gauss-Newton approach (GN) performs badly in tracking moving objects under real-time conditions. This effect was already postulated in section 3.4 showing a reduced range of convergence with respect to the rotational and translational object velocities. This property now becomes more evident.

Surprisingly, the extension to the complementary subspace (GN-IC-CS) does not show a clear gain in performance for any of the objects. A possible interpretation of this fact finds its reason in the motion templates, i.e., the motion Jacobian, being partially contained in the illumination subspace. Hence, convergence would suffer from less clear local derivatives.

The template-update method (GN-IC-TU) provably outperforms the method of choice (GN-IC) and in turn all other methods. Obviously, the two step strategy to track an first updated texture-model and subsequently the original texture-model supports the convergence to the true pose.

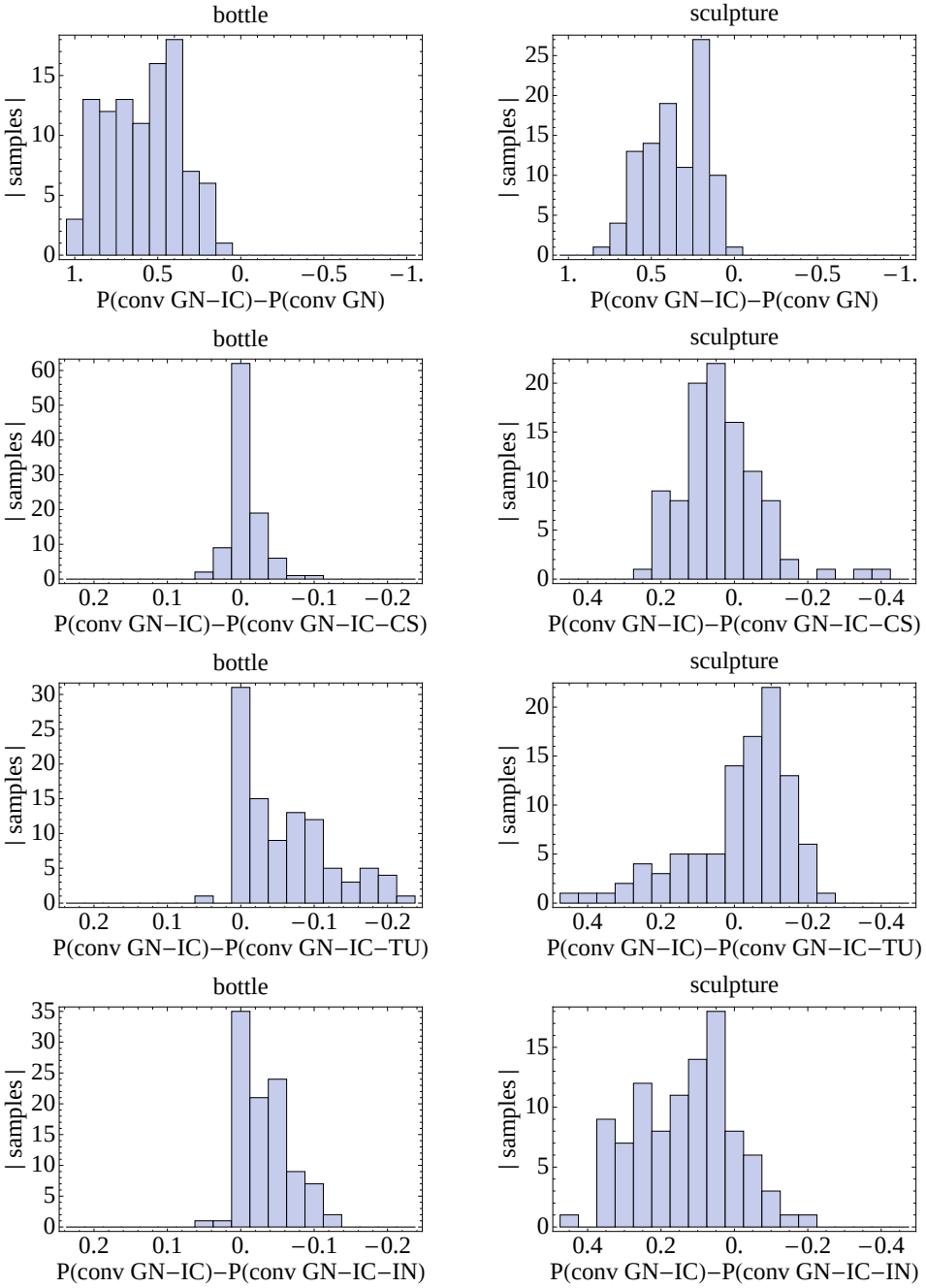

Figure 4.6: Histograms of gains in convergence probability for GN-IC with respect to (from top to bottom): GN, GN-IC-CS, GN-IC-TU, and GN-IC-IN. Left: bottle sequences. Right: sculpture sequences.

The modification for the intensity-distribution normalisation (GN-IC-IN) does not show a clear trend. While for the sequences of the bottle object the method succeeded more often to converge, it is more likely not to converge on the sculpture sequences. The reasons for this behaviour could be found in differences in the lighting conditions of both objects: the bottle sequences show moderate changes in illumination while those for the sculpture exhibit in part challenging illumination conditions.

Finally, the accuracy of pose estimation is assessed for all methods under the above mentioned real-time constraints. Table 4.3 reports the accuracy with respect to those samples for which convergence is attained. Again, the standard

	bottle			sculpture		
	P(conv) [%]	σ_{rot} [°]	σ_{trans} [mm]	P(conv) [%]	σ_{rot} [°]	σ_{trans} [mm]
GN	19.6	0.45	0.67	5.5	1.42	1.47
GN-IC	75.3	0.30	0.61	42.9	1.16	0.94
GN-IC-CS	75.9	0.31	0.61	38.6	1.18	0.78
GN-IC-TU	81.8	0.29	0.60	44.3	0.95	0.83
GN-IC-IN	78.4	0.28	0.61	28.0	1.29	0.80

Table 4.3: Accuracy for estimation procedures converged to the true pose and probability of convergence for all considered samples under real-time constraints.

Gauss-Newton method (GN) shows the worst performance. On average, in the bottle sequences an accuracy of 0.30° and 0.6 mm is obtained for the methods that take advantage of the efficient prediction of the Jacobian (GN-IC), while in case of the sculpture sequences, an accuracy of 1.1° and 0.8 mm is reached. For the former sequences no significant difference can be ascertained among the IC-methods, in contrast to the sculpture images, where the advantage of texture update (GN-IC-TU) becomes evident.

Some snapshots of successful tracking of both the bottle and the sculpture are augmented and displayed in figures 4.7 and 4.8. The tracked point set is shown in yellow and manually outlined for better visualisation. The bars in the lower part of the pictures indicate the degree of rotation around the Cartesian axes (left) together with the degree of translation along these axes (right).

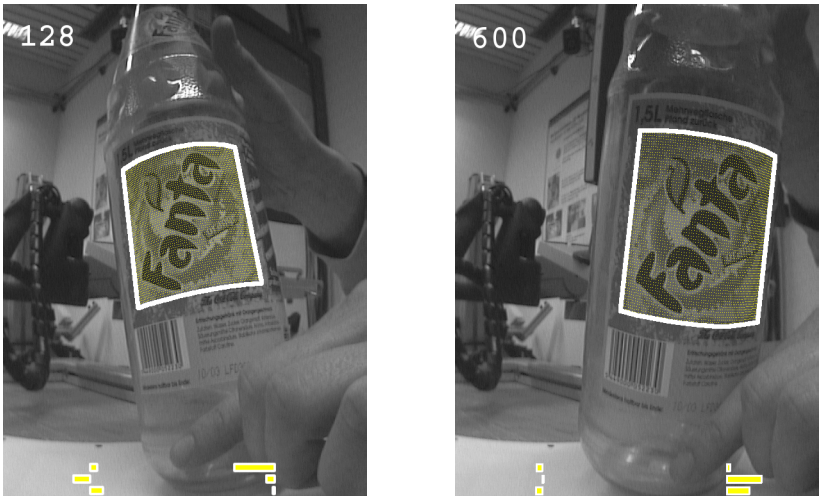

Figure 4.7: Images no. 128 and 600 of a bottle sequence augmented with the tracked 3-d model point cloud and indications of the estimated 6-DoF pose at the bottom. The tracked points are manually outlined for better visualisation.

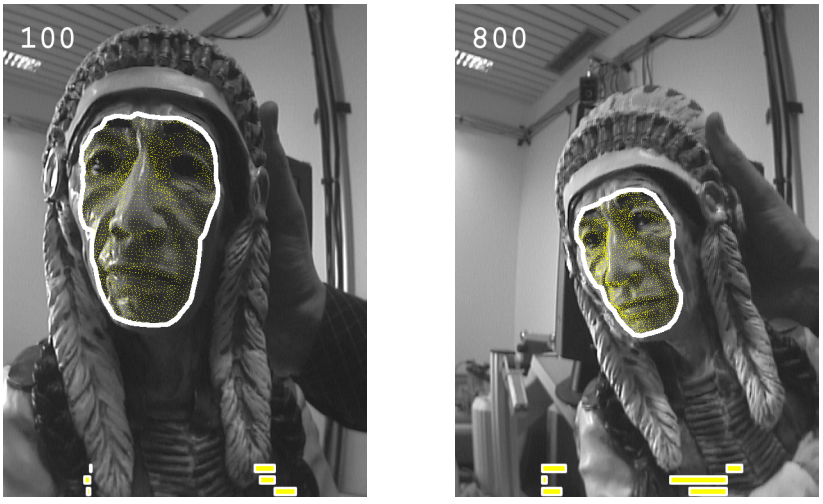

Figure 4.8: Images no. 100 and 800 of a sculpture sequence augmented with the tracked 3-d model point cloud and indications of the estimated 6-DoF pose at the bottom. The tracked points are manually outlined for better visualisation.

5 Hierarchical Visual Tracking

The robustness of tracking moving objects primarily depends on the conformity of the visual appearance with the assumptions employed in the tracking algorithm. Typically, assumptions are set up on the object dynamics, e.g., on the maximal object velocity and on the objects visibility. Obviously, the object gets lost if one of the assumptions is violated, i.e., by to fast motion, by the object leaving the visible volume or the object being obscured by foreign bodies.

Thus, seamless object tracking cannot be guaranteed in uncontrolled environments. Nevertheless, the acceptability of the application that incorporates the tracking method depends on a robust initialisation and the ability to recover from "failure" so that the loss of the object becomes temporary and not permanent. With regard to visual servoing of moving objects, the application is required to perform global pose estimation, irrespective previous object locations in order to be able to (re-)initialise tracking at any time.

Nowadays, global pose estimation in 6 degrees of freedom (DoF) is possible in real-time for feature-based methods [86] under the burden of heavy a priori training. Non feature-based tracking in 6 DoF, however, is still out-of-reach for current personal computer technology. In theory, the search space for this type of tracking is $O(r^d)$ where d denotes the number of degrees of freedom and r represents the search range in each DoF (see Figure 5.1). The computational efforts grow linearly with the search space and, hence, global search in all 6 DoF can be achieved only at comparatively high computational costs.

In the following, an architecture is presented that addresses all the requirements of tracking applications in consideration of limited computing power. The task of global pose estimation in 6 DoF is split into several stages, which allow for seamless transition between initial object detection, tracking, and consecutive re-initialisation of tracking when the object gets lost.

Section 5.1 starts with the theoretical background of tracking at different stages showing the dependencies of the maximal object velocity on hardware constraints, task specifications, and methodological characteristics. These find-

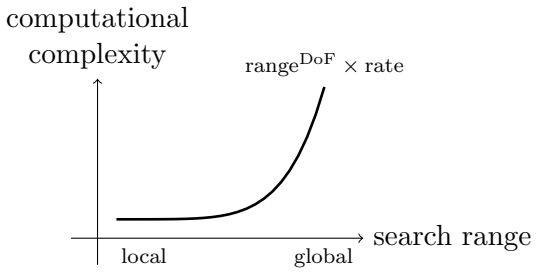

Figure 5.1: Computational complexity of appearance based tracking approaches for a constant number of degrees of freedom (DoF) and constant temporal sampling rate.

ings are used in section 5.2 to identify appropriate object models and sampling strategies for an appearance-based tracking cascade. The rules of activation aiming at switching between the single stages of the cascade are given in section 5.3. The following section 5.4 proposes concrete methods for quick object localisation in all translational degrees of freedom, while the final section 5.5 presents the methods for subsequent refinement and tracking in all translational and rotational degrees of freedom.

5.1 Theory of Multi-Level Tracking

Limitations in the computational power force the restriction of the requirements on global pose estimation in 6 DoF to the most important demands of visual servoing and grasping. The top level demands are identified in the following to be:

requirements of space

- pose estimation in all 6 DoF
- high translational and rotational accuracy

requirements of time

- pose estimation at high temporal sampling rates

The first two sub-items are mandatory for grasping objects in a definite and accurate manner. As such, they are not particular for visual servoing but general for the purpose of grasping. Additionally to these static requirements, the third sub-item introduces a constraints on the temporal behaviour of tracking. It imposes rapid estimation of successive poses in order to minimise the lag of the pose estimation process and to allow for seamless integration in a visual servoing loop. In combination, the above constraints ensure accurate tracking and grasping of moving objects. These necessities are hereafter referred to as *target requirements*.

Though these demands represent the minimal set of characteristics of the servoing application, they still cannot be accomplished "on-the-fly", i.e., without prior information. Less exigent requisites are set up that should be fulfilled at the beginning of the application. The definition of both *entrance* and *target* requirements allows to set up a multi-level tracking application that successively meets increasingly demanding constraints. The minimal requirements that have to be imposed on the first stage of tracking are identified to be:

requirements of space and time

– tracking at high object velocities

This is the only requisite and allows for fast object detection without prior knowledge and is implicitly also suited for re-initialisation of tracking as the target got lost. In the following, this necessity is referred to as *entrance requirement*.

Theoretically, tracking algorithms could simultaneously meet target and entrance requirements. Whether both can be accomplished at the same time depends on exterior circumstances such as the available processing power. The computational resources are considered here inadequate to simultaneously accomplish all the constraints.

The target and entrance requirements are not sufficient to uniquely identify the concrete tracking algorithms neither on the bottom (entrance) or top (target) level nor on intermediate levels. The constraints affect different domains and therefore no direct link is visible between entrance and target requirements. For the purpose of multi-level tracking, the correlation between the constraints in the domains of DoF, accuracy, temporal sampling rate, and object velocity must be determined. Hence, the question arises whether these demands represent concerted or concurrent goals.

To answer the question, the three most common strategies for pixel-based tracking are analysed in more detail. All of these strategies find the most probable object location (and possibly orientation) by comparing the object appearance with the pixel measurements corresponding to a specific location hypothesis. The approaches differ in their strategies to explore the range of possible locations. These strategies perform either

- regular (e.g. correlation based),
- sequential (e.g. gradient-descent based), or
- stochastic (e.g. Monte-Carlo based)

sampling of the object pose space. The methods are characterised by a particular relation of the properties in question, i.e., DoF, accuracy, temporal sampling rate, and object velocity. In the following, the individual relation of the properties of the complete system, i.e., properties of the hardware, the method, and the task, are identified and unfolded to a directed graph. Here, the object velocity, being a property of the task to accomplish, specifies the root node.

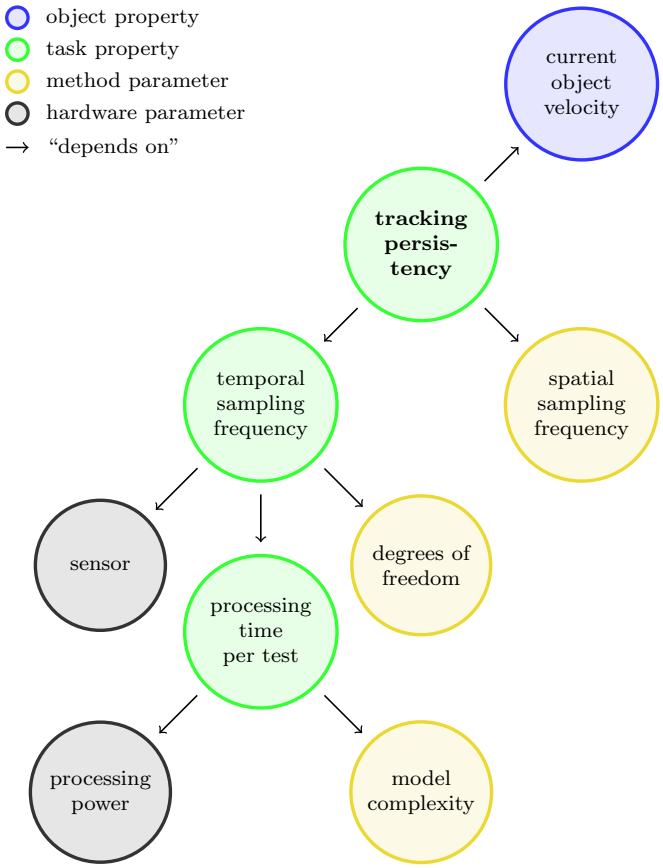

Figure 5.2: Graph of property and parameter dependencies characterising regular sampling. The graph emphasises the persistency of tracking, which implicitly limits the sustained object velocity.

5.1.1 Characteristics of Regular Sampling

The equidistant and equiangular exploration of the pose space is denominated regular sampling. Examples of regular sampling methods are correlation-based approaches for object detection.

The relation of the properties DoF, temporal sampling rate, and object velocity have been identified in section 1.1.3. Accordingly, the object velocity eventually depends on the sustained temporal sampling rate as well as on the spatial sampling rate.

The former is limited by characteristics of the camera hardware, by the processing time per sample, and by the dimensionality of the explored pose space. For given characteristics of the sensor and the computational resources, the remaining properties, i.e., complexity of the object model and the number of DoF, are the only parameters that affect the temporal sampling rate and eventually the maximal object velocity. Hence, increasing demands on the number of DoF or the model complexity result in a lower sampling rate and a lower object velocity for given hardware resources. The dependence graph 5.2 illustrates the relation of the mentioned properties with exception of the

accuracy of object localisation.

The latter is determined by the spatial sampling frequency and by the model complexity. In principle, the more measurements are accumulated for the evaluation of a model, the higher the positional accuracy[1] will be. Since increasing object velocities as well as increasing localisation accuracy demand for more computational resources, both objectives cannot be achieved contemporaneously.

5.1.2 Characteristics of Sequential Sampling

In contrast to regular sampling methods, the sequential exploration of the pose space can eventually achieve arbitrary localisation accuracy because the path of exploration is adapted to the observation at hand. Examples of sequential sampling methods are gradient-descent based approaches for function minimisation.

According to section 1.1.3, the object is tracked as long as the initial pose is contained in the region of hypotheses converging to the true pose. The gap between these two poses depends on the object velocity, the temporal sampling rate, and the convergence rate of the tracking method. The limitation of the supported gap to the radius of convergence constitutes in return a limitation on the supported object velocity. Figure 5.3 shows the dependencies of the most important properties. In comparison to regular sampling, the choice of the minimisation method and the choice of the objective function play a more important role for the efficiency of the tracking approach.

The localisation accuracy, the second property of interest besides the object velocity, is determined by the final gap between the current pose estimate and the true pose. The final gap depends in turn on the gap between the initial pose hypothesis and the true pose and on the temporal convergence rate of the minimisation method. To increase the temporal convergence rate, either more computational resources, less complex models, less degrees of freedom, or faster minimisation methods have to be engaged. In the latter case, for instance, second order minimisation techniques let the pose estimate converge quadratically to the best estimate in contrast to the linear convergence of first order techniques.

5.1.3 Characteristics of Stochastic Sampling

Alternatively, object tracking is accomplished through stochastic sampling of the pose space using multiple, independent pose hypotheses. The sampling process follows a model of the object dynamics formulated as a transition probability function. Hence, the supported object velocity is implicitly specified by the model of dynamics. An example of stochastic sampling are Markov-Chain Monte-Carlo methods such as Particle Filtering.

The persistency of stochastic tracking depends on two properties, the coverage of the state density predicted through the dynamic model by hypotheses

[1] The Nyquist criteria limits the spatial sampling frequency respectively to the frequency components of the object appearance.

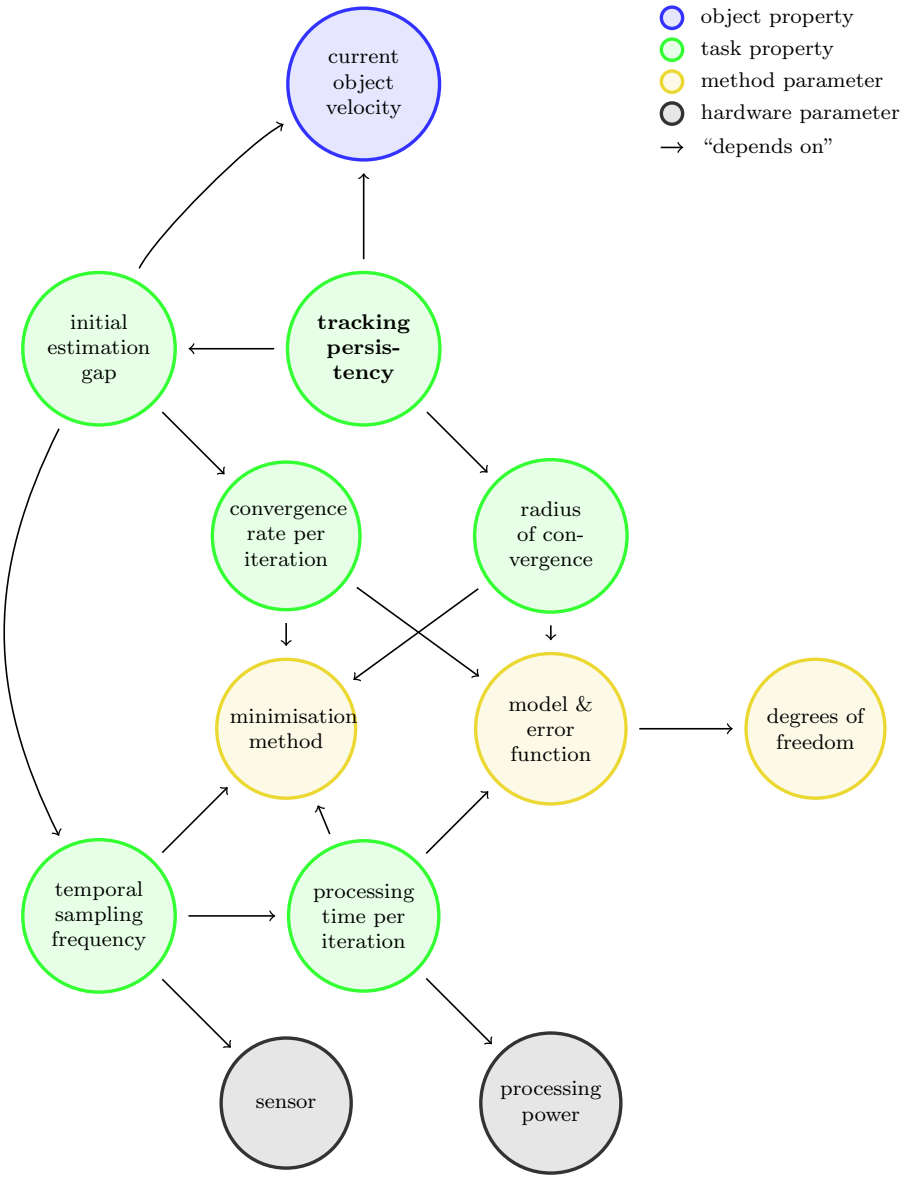

Figure 5.3: Graph of property and parameter dependencies characterising sequential sampling.

(particles), and the broadness of the observation density. Again, the higher the sampling rate, the more compact the predicted density becomes, and the better the density will be covered by particles. Hence, computational resources are freed for handling faster object movements. Figure 5.4 illustrates the major dependencies of the parameters and properties of stochastic tracking.

The coverage of the posterior probability density for object poses by appropriate hypotheses represents the key figure for the accuracy of the estimation process. Doucet et al. [44] assessed a convergence rate of $1/N$ for estimators on bounded functions of the simulated density toward the true estimate, whereas N denotes the number of particles.

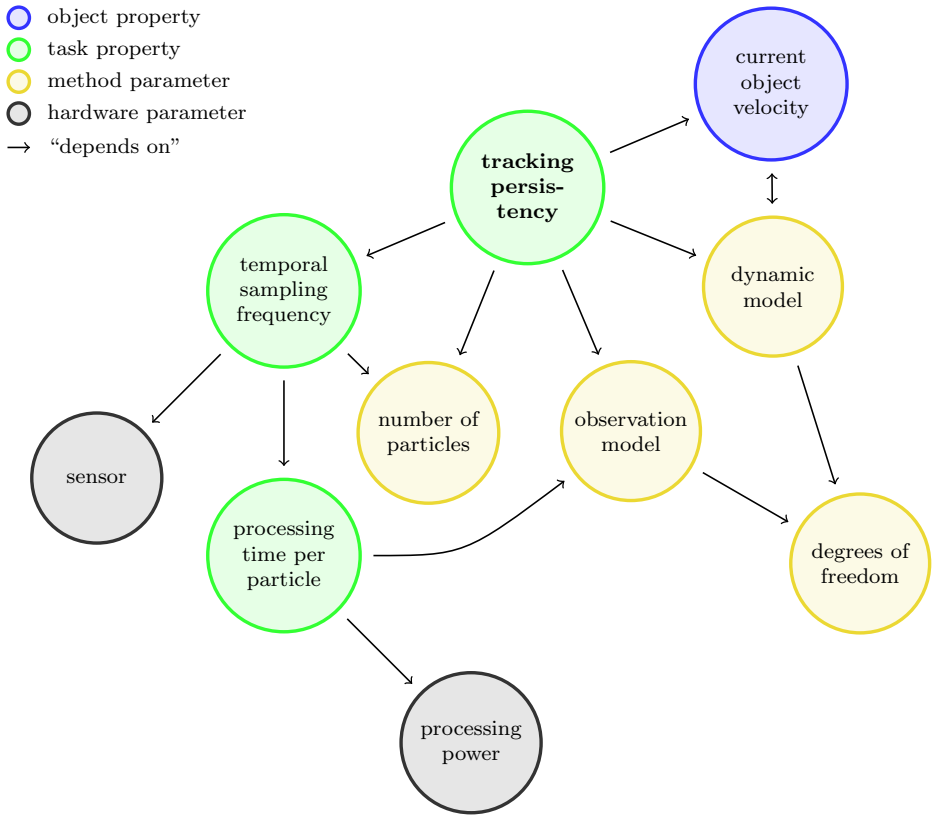

Figure 5.4: Graph of property and parameter dependencies characterising stochastic sampling.

5.1.4 Common Characteristics

The correlation between the requirements in the domains of DoF, accuracy, temporal sampling rate, and the object velocity can be generalised over the considered tracking strategies. The following properties are ascertained:

- temporal sampling frequency $\uparrow \Rightarrow$ supported object velocity $\uparrow$
- number of DoF $\uparrow \Rightarrow$ supported object velocity $\downarrow$
- object model complexity $\uparrow \Rightarrow$ supported object velocity $\downarrow$.

Accordingly, increasing the sampling rates will increase the sustained objects velocities independently of the above mentioned methods. By contrast, the number of DoF and the employed complexity of the object model are related anti-proportionally to the supported velocity. Hence, on one hand, tracking slows down as the employed observation model increases and as more degrees of freedom are considered. On the other hand, objects are localised with increasing accuracy under these variations.

5.2 Appearance-based Multi-Level Tracking

The observations of the previous section are applied in the following to a novel cascade of appearance-based and appearance descriptor-based tracking stages (see Figure 5.5).

The stages are designed for increasing degrees of freedom as well as for increasing model and localisation accuracy from the bottom level to the top level. Objects are first tracked at high velocities and high positional uncertainty at the bottom level. Then, the degrees of freedom are gradually increased along with an increasing refinement of the observation model. At the same time, the sustained object velocity is reduced and the localisation becomes more accurate. The top level is designed to guarantee accurate tracking of objects at high temporal sampling rates and moderate object velocities.

The specific choices for the levels of appearance-based tracking are based on two characteristics, the type of object model and the sampling strategy.

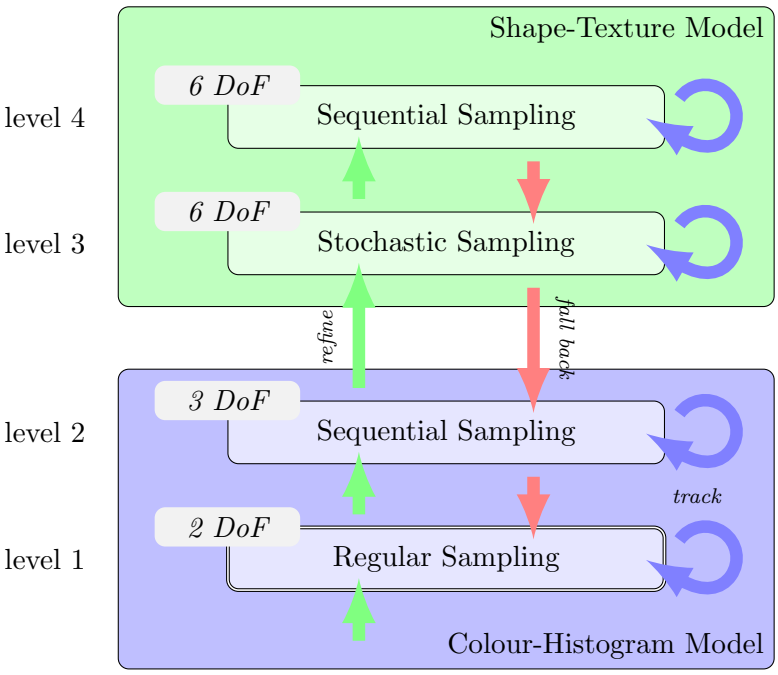

Figure 5.5: Appearance-based multi-level tracking.

5.2.1 Object Model

The first design decision is made with respect to the object model. Obviously, the decision is coupled to the desired accuracy and to the required degrees of freedom. At the top level, the employed object model must allow unambiguous estimation of the object pose in all 6 dof. Obviously, the shape-texture representation of section 3.1.3 models the appearance for different poses in detail. The pose can be estimated precisely, as long as the unity of texture and surface

ensures measurable variation of appearance for arbitrary motion in 6 DoF[2].

At the entry level, no requirements exist with respect to the number of supported degrees of freedom. Accordingly, the object model is expected to be generic enough to cope with variations in the neglected degrees of freedom. In practice, histogram-based models are reasonably robust in reference to changes of the object orientation. Feature matching with gradient location and orientation histogram descriptors, for instance, proved to support out-of-plane rotation of 50°, though with a lower detection rate with respect to the reference view [89, 99]. Colour histograms have been shown to allow tracking of non-rigid objects with affine motion parameters [37, 104].

5.2.2 Sampling Strategy

The completion of the choices on the object model and the DoF with an appropriate selection of the sampling strategy finally allows to uniquely identify the algorithms for the single tracking stages.

As stated at the beginning of the section, the requirements on the top level of the tracking cascade comprise pose estimation in all 6 DoF, accurate pose estimation, and tracking at high sampling rates. Sequential sampling methods based on first order or second order function minimisation are known to allow for accurate parameter estimation. The efficiency of these approaches typically depends on the employed observation model and on the computational costs for the calculation of the corresponding first order derivative. In accordance with the optimisations in the computation of the Jacobian presented in sections 3.2.4, 3.2.5, sequential sampling ensures low computational costs and hence a high sampling frequency when adopted to the Shape-Texture based observation model of section 3.1.3.

In the case of sequential sampling, the sustained object velocity depends on the radius of convergence, the convergence rate per iteration, and the temporal sampling frequency. The optimisations mentioned above address not only the demand for a high sampling rate. Eventually, these optimisations increase the radius of convergence in terms of the object velocity in contrast to unoptimised sequential sampling. Experiments in favour are reported in section 3.4.

Clearly, limitations in the radius of convergence exist, and hence sequential sampling is not suited for the initialisation of the tracking cascade. Accordingly, the limitation is relaxed on the next lower level employing stochastic sampling. In particular, Monte-Carlo methods are not sensitive to local minima as single-hypothesis trackers because multiple hypotheses are considered at once. However, the covered pose space as well as the accuracy of final pose estimation depends on the distribution of the hypotheses. Hence, an increased coverage of the pose space is accompanied either by a lower temporal sampling rate when increasing the number of hypotheses, or by a decreased localisation accuracy when the number of hypotheses is kept constant.

[2]This correlation is pose dependent. For instance, the conditionedness of the estimation of out-of-plane rotation deteriorates as the object moves away from the camera and the full-perspective distortion turns into a weak-perspective projection.

The restriction to fewer degrees of freedoms allows to adopt other sampling strategies at the entry level. Nowadays, regular, correlation-based sampling is feasible for global pose estimation in 2 DoF and thus ideally suited for the bottom level of the tracking cascade. The ability to localise the object without prior knowledge on the location ensures the process to keep up with objects moving arbitrarily fast. However, the rate of this estimation process is not assured. The gap from object localisation in the image plane to pose estimation in 3-d is bridged by adopting a sequential sampling method to the same observation model. Hence, regular sampling in 2-d on the bottom level is succeeded by irregular, observation dependent sampling for pose estimation in 3-d on the next higher level. Figure 5.5 summarises on the methods selected for the tracking cascade.

5.3 Switching Rules for Multi-Level Tracking

Reduced computational power force to activate tracking levels in turn and not in parallel. The transition rules from one level to the next higher or lower level are kept simple in favour of a lean implementation.

Each individual tracking method is based in principle on an observation model and a confidence value. The latter value indicates the degree of consistency of the pose estimate with the current measurement. Accordingly, a pose estimate is linked to a high confidence value if the observation model fits very well with the current measurement for the particular pose. On the other hand, a low confidence indicates that the observation model does not well explain the current measurement for the proposed pose estimate.

Therefore, two confidence bounds are individually chosen for each tracking level. These bounds mark the transition to an adjacent level. Let ${}^{t}\zeta \in 1, 2, \ldots, {}^{\zeta}N$ specify the level active at time t, whereas 1 denotes the bottom layer and ${}^{\zeta}N$ the top layer. Moreover, let $\theta_l^- \in [0, 1]$ and $\theta_l^+ \in [0, 1]$ denote the lower and respective upper confidence bound for level $l \in 1, 2, \ldots, {}^{\zeta}N$. The active process switches from the current tracking level to the next lower level when the confidence value reaches the lower confidence bound, and to the next higher level when the upper confidence bound is reached. Accordingly,

$$ {}^{t+1}\zeta = {}^{t}\zeta + \begin{cases} +1 & \text{if} \quad {}^{t}\zeta < {}^{\zeta}N \quad \wedge \quad \theta_{{}^{t}\zeta}^{+} \leq {}^{t}\gamma \\ -1 & \text{if} \quad 1 < {}^{t}\zeta \quad \wedge \quad {}^{t}\gamma \leq \theta_{{}^{t}\zeta}^{-} \\ 0 & \text{else} \end{cases} \tag{5.1} $$

whereas ${}^{t}\gamma$ denotes the confidence value computed at the level active at time t. The lower and respective upper confidence bound specify different rules on the entry and the target level. If the confidence falls below the lower bound on the bottom level, then the object may not be present at all in the image. Usually, the desired action in this case is to repeat detection on the bottom level until the object is finally found. Likewise, when the confidence exceeds the upper bound on the top level, then observing instances can take appropriate action. In the context of visual servoing and grasping the appropriate control signals are emitted commanding the robot to catch the object.

The above switching rules are implicitly based on some assumptions about the significance of the confidence value. The employed model is thought to be robust to variations induced by parameters other than pose, e.g. illumination parameters. In this situation, the lower confidence bound is only reached when the estimated pose does probably not match the true pose. Conversely, the upper confidence bound is thought to be reached only when the estimated pose matches the true pose accurately enough.

However, unambiguity of local extrema of the optimised function, e.g. the objective function or likelihood function, is not guaranteed. Therefore, the upper and lower confidence bounds have to be chosen conservatively to ensure a proper switching behaviour. The upper confidence bound should be set large enough to distinguish between estimates close to the true pose and estimates in a nearby local minima. Respectively, the lower confidence bound should be set small enough to quickly detect when the object got lost.

5.4 Histogram-based Localisation and Tracking

The first levels of tracking aim at detecting and tracking a fast moving object in a reduced set of degrees of freedom. In the computer vision community, histogram-based models are used for description and matching of single features as well as for tracking moving objects. Especially histograms of colour (e.g. [104, 89]) or gray-level gradient distributions (e.g. [89, 99]) are frequently used due to their robustness with respect to illumination changes and variations of pose.

Here, the focus falls on the colour distributions, in particular on the colour histogram used by Comaniciu et al. [37] as target models as presented in [123, 56]. In contrast to gradient-based object descriptions, colour histograms demand slightly less computational resources because no derivative of the image is computed.

In the following, let ${}^tI(\mathbf{v}) \in \mathcal{C}$ denote the colour value at the position $\mathbf{v} \in \mathbb{N}^2$ of the image at time t in an arbitrary but fixed colour space $\mathcal{C}$. In addition, let the function $h : \mathcal{C} \to \{1, 2, \ldots, {}^{\mathrm{b}}N\}$ map each colour value to the index of a corresponding colour bin and let

$$k(\mathbf{u}, \boldsymbol{\sigma}) = \exp\left(-\frac{1}{2}\mathbf{u}^{\mathrm{T}}\Sigma(\boldsymbol{\sigma})\,\mathbf{u}\right) \tag{5.2}$$

define an anisotropic kernel in $\mathbb{R}^2$ with standard deviation $\boldsymbol{\sigma}$. The diagonal matrix

$$\Sigma((\sigma_{\mathrm{u}}, \sigma_{\mathrm{v}})) = \begin{pmatrix} \sigma_{\mathrm{u}}^2 & 0 \\ 0 & \sigma_{\mathrm{v}}^2 \end{pmatrix} \tag{5.3}$$

allows for individually scaled axes. The probability of occurrence of a specific bin b in the neighbourhood of the position $\mathbf{u} \in \mathbb{R}^2$ at time t is now defined by

$${}^tp_b(\mathbf{u}, \boldsymbol{\sigma}) = \frac{\sum_{\mathbf{v}} k(\mathbf{u} - \mathbf{v}, \boldsymbol{\sigma})\,\delta\big(h\big({}^tI(\mathbf{v})\big) - b\big)}{\sum_{\mathbf{v}} k(\mathbf{u} - \mathbf{v}, \boldsymbol{\sigma})}\,, \tag{5.4}$$

whereas $\delta : \mathbb{R} \to \{0, 1\}$ represents the Kronecker delta function

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{else} \end{cases} \tag{5.5}$$

and where the colour occurrence is weighted according to the spatial distance to the centre by the anisotropic kernel k.

The object to be tracked is recognised by the prior colour probabilities $\mathbf{q} = (q_1, q_2, \ldots, q_{\mathrm{b}N}) \in [0,1]^{\mathrm{b}N}$, $\sum_{i=1}^{\mathrm{b}N} q_i = 1$ obtained from a reference image. The similarity between the current colour distribution ${}^t\mathbf{p} = ({}^tp_1, {}^tp_2, \ldots, {}^tp_{\mathrm{b}N}) \in [0,1]^{\mathrm{b}N}$, $\sum_{i=1}^{\mathrm{b}N} {}^tp_i = 1$ and the prior distribution $\mathbf{q}$ is measured by the Bhattacharyya coefficient

$$\rho\big({}^t\mathbf{p}, \mathbf{q}\big) = \sum_b \sqrt{{}^tp_b \; q_b} \tag{5.6}$$

which takes the value 1 for identical distributions.

5.4.1 2-DoF Histogram-based Localisation

The initial object detection is achieved evaluating the Bhattacharyya coefficients in the image plane on regular grid positions. In order to reduce the occurrence of local minima and to decrease the computational costs, the coefficients are not computed at every image position. Instead, the possible domain of 2-DoF location is sub-sampled.

In this sense, let $\boldsymbol{\sigma} = (\sigma_{\mathrm{u}}, \sigma_{\mathrm{v}})$ determine the elongation of the kernel k. The Bhattacharyya coefficients are sampled at intervals $2\sigma_{\mathrm{u}}$, $2\sigma_{\mathrm{v}}$ assuring good coverage in the image as with Gaussian pyramids [27]. Let $S = \{(2i\sigma_{\mathrm{u}}, 2j\sigma_{\mathrm{v}}) \mid i, j \in \mathbb{N}\}$ be the set of sampling coordinates, then the location ${}^t\hat{\boldsymbol{\mu}} \in \mathbb{R}^2$ of the object in the image space is estimated according to

$${}^t\hat{\boldsymbol{\mu}} = \arg\max_{\mathbf{u} \in S} \sum_b \sqrt{{}^tp_b(\mathbf{u}, \boldsymbol{\sigma}) \; q_b} \tag{5.7}$$

and the confidence value is given by

$${}^t\gamma = \max_{\mathbf{u} \in S} \sum_b \sqrt{{}^tp_b(\mathbf{u}, \boldsymbol{\sigma}) \; q_b} \; . \tag{5.8}$$

The detection is rejected as long as the confidence value is lower than the detection threshold θ_1^+.

5.4.2 3-DoF Histogram-based Tracking

Object tracking by means of sequential optimisation starts immediately after localisation of the object in the image plane. At this stage, the initial two-dimensional position is augmented by an additional scale parameter.

Tracking is performed with the Mean-Shift algorithm, a non-parametric statistical method for sequential determination of the nearest mode of a point sample distribution. The shift is first computed in planar coordinates according to Comaniciu et al. [37]. Here, the initial hypothesis ${}^t\hat{\mathbf{u}}_0 \in \mathbb{R}^2$ of the object location in the image plane at time instant t is gathered from the previously active tracking stage ${}^{t-1}\zeta$ according to the rule

$${}^t\hat{\mathbf{u}}_0 = \begin{cases} {}^{t-1}\hat{\boldsymbol{\mu}} & \text{if } {}^{t-1}\zeta = 1 \\ {}^{t-1}\hat{\mathbf{u}}_{N_2} & \text{if } {}^{t-1}\zeta = 2 \\ p\big(m\big(E(X), {}^{t-1}\hat{\boldsymbol{\mu}}\big)\big) & \text{if } {}^{t-1}\zeta = 3 \end{cases} \; . \tag{5.9}$$

Here, $E(X) = \frac{1}{N}\sum_{\mathbf{x}\in X}\mathbf{x}$ represents the centroid of the surface-model points. Hence, if the active process switched from the detection level (${}^{t-1}\zeta = 1$) to this tracking level (${}^{t}\zeta = 2$), then the hypothesis ${}^{t}\hat{\mathbf{u}}_0$ is gathered from the previously estimated location ${}^{t-1}\hat{\boldsymbol{\mu}}$. If no level switch occurred from time instant $t-1$ to t (${}^{t-1}\zeta = 2$), then the hypothesis is given by the final estimate ${}^{t-1}\hat{\mathbf{u}}_{N_2}$ previously reached after N_2 iterations. If the next higher tracking level (${}^{t-1}\zeta = 3$) was active at $t-1$, then the centroid of the model is projected to the image with the previous position estimate ${}^{t-1}\hat{\boldsymbol{\mu}}$ in order to obtain the hypothesis. After this initialisation step, the location hypothesis is refined in N_2 iterations.

At each iteration i, every position in the neighbourhood of the estimate ${}^{t}\hat{\mathbf{u}}_i$ is weighted according to the relevance of the corresponding colour bin. The relevance of a single bin is determined by the ratio of the frequency of its occurrence in the reference pattern to the corresponding frequency in the current pattern. Hence, the weight at a specific location $\mathbf{u} \in \mathbb{R}^2$ with respect to a position $\mathbf{v}$ reads

$$ {}^{t}w(\mathbf{u},\mathbf{v},\boldsymbol{\sigma}) = \sum_b \sqrt{\frac{q_b}{{}^{t}p_b(\mathbf{u},\boldsymbol{\sigma})}}\,\delta\big(h\big({}^{t}I(\mathbf{v})\big) - b\big) \tag{5.10} $$

for a specific scale $\boldsymbol{\sigma}$. Now, a new object location ${}^{t}\hat{\mathbf{u}}_{i+1}$ is estimated for the current scale estimate ${}^{t}\hat{\boldsymbol{\sigma}}_i$ by

$$ {}^{t}\hat{\mathbf{u}}_{i+1} = \frac{\sum_{\mathbf{v}} k\big({}^{t}\hat{\mathbf{u}}_i - \mathbf{v}, {}^{t}\hat{\boldsymbol{\sigma}}_i\big)\, {}^{t}w\big({}^{t}\hat{\mathbf{u}}_i, \mathbf{v}, {}^{t}\hat{\boldsymbol{\sigma}}_i\big)\, {}^{t}\hat{\mathbf{u}}_i}{\sum_{\mathbf{v}} k({}^{t}\hat{\mathbf{u}}_i - \mathbf{v}, {}^{t}\hat{\boldsymbol{\sigma}}_i)\, {}^{t}w({}^{t}\hat{\mathbf{u}}_i, \mathbf{v}, {}^{t}\hat{\boldsymbol{\sigma}}_i)} \tag{5.11} $$

with the kernel k of 5.2. The initial scale hypothesis ${}^{t}\hat{\boldsymbol{\sigma}}_0$ is set either to the previous scale estimate or to the default scale $\boldsymbol{\sigma} = (\sigma_{\mathrm{u}}, \sigma_{\mathrm{v}})$ used for detection, that is

$$ {}^{t}\hat{\boldsymbol{\sigma}}_0 = \begin{cases} {}^{t-1}\hat{\boldsymbol{\sigma}}_{N_2} & \text{if } {}^{t-1}\zeta = 2 \\ \boldsymbol{\sigma} & \text{else} \end{cases} . \tag{5.12} $$

Subsequently, the correct scale of the object is identified by the evaluation of pixel relevances over $2n+1$ scales relative to a current scale, that are

$$ {}^{t}\boldsymbol{\sigma}_i^{(s)} = {}^{t}\hat{\boldsymbol{\sigma}}_i \cdot a^s \;; \quad -n \le s \le n \tag{5.13} $$

whereas $a > 1$ denotes a constant logarithmic base. A new scale index ${}^{t}\hat{s}_{i+1}$ and scale ${}^{t}\hat{\boldsymbol{\sigma}}_{i+1}$ are estimated according to

$$ {}^{t}\hat{s}_{i+1} = \frac{\sum_s \sum_{\mathbf{v}} k\Big({}^{t}\hat{\mathbf{u}}_{i+1} - \mathbf{v}, {}^{t}\boldsymbol{\sigma}_i^{(s)}\Big)\, {}^{t}w\Big({}^{t}\hat{\mathbf{u}}_{i+1}, \mathbf{v}, {}^{t}\boldsymbol{\sigma}_i^{(s)}\Big)\, s}{\sum_s \sum_{\mathbf{v}} k\Big({}^{t}\hat{\mathbf{u}}_{i+1} - \mathbf{v}, {}^{t}\boldsymbol{\sigma}_i^{(s)}\Big)\, {}^{t}w\Big({}^{t}\hat{\mathbf{u}}_{i+1}, \mathbf{v}, {}^{t}\boldsymbol{\sigma}_i^{(s)}\Big)} \;; \quad {}^{t}\hat{\boldsymbol{\sigma}}_{i+1} = {}^{t}\hat{\boldsymbol{\sigma}}_i \cdot a^{{}^{t}\hat{s}_{i+1}} . \tag{5.14} $$

Note that here simply the kernel k of 5.2 is used instead of a Laplacian of Gaussian or a Difference of Gaussian as proposed by Collins [36].

For the image at time t, the computation of shift and scale are alternated for N_2 iterations. These N_2 optimisation cycles apply to the following image $t+1$, until either the confidence

$$ {}^{t}\gamma = \sum_b \sqrt{{}^{t}p_b({}^{t}\hat{\mathbf{u}}_{N_2}, {}^{t}\hat{\boldsymbol{\sigma}}_{N_2})\; q_b} \tag{5.15} $$

falls below the threshold specified by θ_2^- or the estimation process converged[3], that is, the Euclidean distance $\|{}^t\hat{\mathbf{u}}_{N_2} - {}^t\hat{\mathbf{u}}_0\|$ and $|{}^t\hat{\boldsymbol{\sigma}}_{N_2} - {}^t\hat{\boldsymbol{\sigma}}_0|$ fall below a convergence threshold θ_2^+. As long as tracking continuous on this level, the estimates of location and scale of the last iteration constitute the initial hypotheses for the corresponding measures in the following image.

For each image, the estimates of two-dimensional object location ${}^t\hat{\mathbf{u}}_{N_2} = ({}^t\hat{u}_{N_2}, {}^t\hat{v}_{N_2})$ and scale ${}^t\hat{\boldsymbol{\sigma}}_{N_2} = ({}^t\hat{\sigma}_{N_2}, {}^t\hat{\psi}_{N_2})$ are mapped to three-dimensional Euclidean space at the last iteration considering a full-perspective projection model with known intrinsic camera parameters

$$K = \begin{pmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.16}$$

and the physical extent d of the object. The physical extent is related to the position t_z in direction of the z-axis by the formula $\alpha \cdot d \approx 4\, {}^t\hat{\sigma}_{N_2} \cdot t_z$. Hence, the estimate for the translation of the object is computed according to

$${}^t\hat{\boldsymbol{\mu}} = \begin{pmatrix} {}^t\hat{\mu}_{\mathrm{x}} \\ {}^t\hat{\mu}_{\mathrm{y}} \\ {}^t\hat{\mu}_{\mathrm{z}} \end{pmatrix} = \frac{\alpha d}{4\, {}^t\hat{\sigma}_{N_2}} \begin{pmatrix} \left({}^t\hat{u}_{N_2} - u_0\right)/\alpha \\ \left({}^t\hat{v}_{N_2} - v_0\right)/\beta \\ 1 \end{pmatrix} . \tag{5.17}$$

The active process switches from this extension of Mean-Shift tracking to the lower level, as soon as the confidence falls below the threshold θ_2^-. The next higher level is invoked when convergence is reached within the threshold defined by θ_2^+.

5.5 Shape-Texture Based Tracking

Subsequent to the initial object detection in 2 DoF and tracking in 3 DoF performed at the entry levels, the top tracking levels aim at tracking the object in 6 DoF with Euclidian translation and rotation parameters.

At these two levels, object representation is replaced by the shape-texture model of section 3.1.3. In the first level, multiple pose hypotheses are checked against the visible appearance of the object. On the top level instead, changes in appearance induced by object and/or camera motion are matched to pose variations of the shape-texture model [123, 56].

5.5.1 6-DoF Multi-Hypotheses Tracking

Object tracking in 3 DoF on the lower level is extended to tracking in 6 DoF adopting the annealed Markov-Chain Monte-Carlo filter as described in section 3.3.2. The robustness and the broad area of convergence qualify sequential Monte-Carlo filters for this task. A sequential Monte-Carlo filter propagates a particle set $\{({}^t\boldsymbol{\mu}_i^{(j)}, {}^tw_i^{(j)}) \mid j = 1, 2, \ldots, {}^sN\}$ over time t, whereas here, multiple iterations $i \in \{1, 2, \ldots, N_3\}$ of the filter are performed for each time instant following a continously decreasing process of particle diffusion.

[3] This rule shows in practice to be an effective alternative to 5.1.

The initial distribution $\{({}^t\boldsymbol{\mu}_0^{(j)}, {}^tw_0^{(j)}) \mid j = 1, 2, \ldots, {}^sN\}$ is set up in dependence of the tracking level ${}^{t-1}\zeta$ active at $t-1$. If 3-DoF tracking was previously active (${}^{t-1}\zeta = 2$), then the single position hypothesis ${}^{t-1}\hat{\boldsymbol{\mu}} = ({}^{t-1}\hat{\mu}_\mathrm{x}, {}^{t-1}\hat{\mu}_\mathrm{y}, {}^{t-1}\hat{\mu}_\mathrm{z})$ is augmented first by the object orientation $({}^0\mu_\alpha, {}^0\mu_\beta, {}^0\mu_\gamma)$ in the reference view to a 6-DoF pose hypothesis ${}^t\hat{\boldsymbol{\mu}}_0 = ({}^0\mu_\alpha, {}^0\mu_\beta, {}^0\mu_\gamma, {}^{t-1}\hat{\mu}_\mathrm{x}, {}^{t-1}\hat{\mu}_\mathrm{y}, {}^{t-1}\hat{\mu}_\mathrm{z})$. Subsequently, ${}^sN$ particles are set up with respect to the single estimate according to

$$ {}^t\boldsymbol{\mu}_0^{(j)} \sim \mathcal{N}\big({}^t\hat{\boldsymbol{\mu}}_0, \Sigma_\mathrm{A}\big) \ , \quad {}^tw_0^{(j)} = \frac{1}{{}^sN} \ , \tag{5.18} $$

whereas Σ_A denotes the covariance matrix expressing the initial uncertainty of the hypothesis. The same policy applies to the initialisation of the particle set if the highest level was previously processed (${}^{t-1}\zeta = 4$). If Monte-Carlo tracking was active and no level switch occurred (${}^{t-1}\zeta = 3$), then the previous distribution of particles is kept. Hence, the rules for the level are:

$$ \Big({}^t\boldsymbol{\mu}_0^{(j)}, {}^tw_0^{(j)}\Big) = \begin{cases} {}^t\boldsymbol{\mu}_0^{(j)} \sim \mathcal{N}\big({}^t\hat{\boldsymbol{\mu}}_0, \Sigma_\mathrm{A}\big) \, , \ {}^tw_0^{(j)} = \frac{1}{{}^sN} & \text{if} \quad {}^{t-1}\zeta = 2 \\ {}^t\boldsymbol{\mu}_0^{(j)} = {}^{t-1}\boldsymbol{\mu}_{N_3}^{(j)}, \ {}^tw_0^{(j)} = {}^{t-1}w_{N_3}^{(j)} & \text{if} \quad {}^{t-1}\zeta = 3 \\ {}^t\boldsymbol{\mu}_0^{(j)} \sim \mathcal{N}\big({}^{t-1}\hat{\boldsymbol{\mu}}, \Sigma_\mathrm{A}\big) \, , \ {}^tw_0^{(j)} = \frac{1}{{}^sN} & \text{if} \quad {}^{t-1}\zeta = 4 \end{cases} . \tag{5.19} $$

The particle set is refined in subsequent steps of Markov-Chain Monte-Carlo filtering by means of the annealed SIR approach of section 3.3.2, following the steps of re-sampling, propagation, and update.

In the former step, an existing particle distribution is re-sampled according to the associated, normalised weights $\{{}^tw_i^{(j)} \mid j = 1, 2, \ldots, {}^sN\}$ resulting in a new set of hypotheses $\{{}^t\boldsymbol{\mu}_i'^{(j)} \mid j = 1, 2, \ldots, {}^sN\}$. The following propagation step resembles an inverse diffusion process. A diffusion vector $\mathbf{v}$, perturbing a single re-sampled pose hypothesis, is given by

$$ {}^t\mathbf{v}_i^{(j)} \sim \mathcal{N}(0, \Sigma_\mathbf{v}) \ , \tag{5.20} $$

whereas $\Sigma_\mathbf{v}$ specifies the covariance matrix of the process noise. Then, the re-sampled hypotheses are propagated to the next iteration according to

$$ {}^t\boldsymbol{\mu}_{i+1}^{(j)} = {}^t\boldsymbol{\mu}_i'^{(j)} + b^i \ {}^t\mathbf{v}_i^{(j)} \tag{5.21} $$

with a decay term b^i, $0 < b < 1$, fading with the iterations. The propagated particle distribution is updated to the posterior distribution by the conditional probability of the observation ${}^tI$ given the propagated poses ${}^t\boldsymbol{\mu}_{i+1}^{(j)}$, which reads

$$ {}^tw_{i+1}'^{(j)} = p\Big({}^tI|{}^t\boldsymbol{\mu}_{i+1}^{(j)}\Big) \ . \tag{5.22} $$

for an unnormalised weight ${}^tw_{i+1}'^{(j)}$. Here, the observation probability $p\Big({}^tI|{}^t\boldsymbol{\mu}_{i+1}^{(j)}\Big)$ is determined by equation 3.17 in respect to the shape-texture representation of the object. In contrast to other approaches, the steps of propagation, update, and re-sampling are repeatedly applied to one observation. Hence, the approach resembles a optimisation process with simulated annealing, and allows to gain accuracy for a single image.

In principle, different information can be extracted from the posterior density $p({}^t\boldsymbol{\mu}|{}^tI)$. Here, only the most probable object pose after all N_3 iterations is considered. This pose is obtained from the particle set approximating the posterior density by

$${}^t\hat{\boldsymbol{\mu}} = {}^t\boldsymbol{\mu}_{N_3}^{(k)} , \quad k = \underset{j}{\operatorname{argmax}}\, {}^tw_{N_3}^{(j)} . \tag{5.23}$$

The confidence value for this pose is given by the corresponding weight, reading

$${}^t\gamma = {}^tw_{N_3}^{(k)} , \quad k = \underset{j}{\operatorname{argmax}}\, {}^tw_{N_3}^{(j)} . \tag{5.24}$$

The above annealed particle filter is applied to the sequence of images as long as the confidence value associated to the most probable pose ${}^t\hat{\boldsymbol{\mu}}$ lies in the interval marked by the lower and upper confidence bounds θ_3^- and θ_3^+. In detail, the lower tracking level becomes active when the confidence falls below θ_3^-. Conversely, the next higher tracking level is processed in the following if the confidence exceeds θ_3^+.

5.5.2 6-DoF Single-Hypothesis Tracking

The 6-DoF pose estimate obtained with multi-hypotheses tracking on the previous level is refined on the top level of the tracking cascade. Here, a single pose hypothesis is sequentially optimised according to the Maximum-Likelihood estimator of section 3.2.4.

Depending on the previously active level, the initial 6-DoF hypothesis ${}^t\hat{\boldsymbol{\mu}}_0$ is initialised either with the most probable pose of Monte-Carlo filtering or to the estimate of Gauss-Newton based tracking performed on this level. Hence, the rule reads

$${}^t\hat{\boldsymbol{\mu}}_0 = {}^{t-1}\hat{\boldsymbol{\mu}} \quad \text{if } {}^{t-1}\zeta \in \{3, 4\} . \tag{5.25}$$

The single pose estimate is iteratively refined with respect to the same shape-texture based observation model as used in the previous stage (cf. equation 3.17). By contrast, at this stage the pose is determined that maximises the log-likelihood 3.18 associated to the observation model. The task is accomplished by minimising the negative log-likelihood by means of the Gauss-Newton approach. The linear equation system, which is set up for a single iteration i of the approach (cf. equation 3.36), reads

$$\begin{aligned} \sum_{\mathbf{x}\in X} \partial_{\delta\boldsymbol{\mu}} {}^tI_{\mathbf{x}}\big({}^t\hat{\boldsymbol{\mu}}_i \circ \delta\boldsymbol{\mu}\big)^{\mathrm{T}} \cdot \partial_{\delta\boldsymbol{\mu}} {}^tI_{\mathbf{x}}\big({}^t\hat{\boldsymbol{\mu}}_i \circ \delta\boldsymbol{\mu}\big) \Big|_{\delta\boldsymbol{\mu}=0} {}^t\hat{\delta\boldsymbol{\mu}}_i = \\ \sum_{\mathbf{x}\in X} \partial_{\delta\boldsymbol{\mu}} {}^tI_{\mathbf{x}}\big({}^t\hat{\boldsymbol{\mu}}_i \circ \delta\boldsymbol{\mu}\big)^{\mathrm{T}} \Big|_{\delta\boldsymbol{\mu}=0} \big({}^tI_{\mathbf{x}}\big({}^t\hat{\boldsymbol{\mu}}_i\big) - {}^0I_{\mathbf{x}}\big({}^0\boldsymbol{\mu}\big)\big) \end{aligned} \tag{5.26}$$

and solved for a variation of pose ${}^t\hat{\delta\boldsymbol{\mu}}_i$ with respect to the pose estimate ${}^t\hat{\boldsymbol{\mu}}_i$. Here, the computationally expensive computation of the Jacobian $\partial_{\delta\boldsymbol{\mu}} {}^tI_{\mathbf{x}}$ for each new estimate is replaced by the efficient approximation 3.41. According to the evaluation of section 3.4, this approach features fast and accurate pose estimation in 6 DoF. Though area of convergence is restricted in respect to the

offset of the initial hypothesis to the true pose, the computational efficiency of the optimised minimisation process allows to gain convergence in respect to the object velocity.

The estimate ${}^t\hat{\boldsymbol{\mu}}_{i+1}$ for iteration $i+1$ is determined by the composition of the previous pose estimate ${}^t\hat{\boldsymbol{\mu}}_i$ and the pose variation ${}^t\hat{\delta\boldsymbol{\mu}}_i$ in accordance with equation 3.7, which reads

$$ {}^t\hat{\boldsymbol{\mu}}_{i+1} = {}^t\hat{\boldsymbol{\mu}}_i \circ {}^t\hat{\delta\boldsymbol{\mu}}_i \; . \tag{5.27} $$

The final estimate for the image at time instant t

$$ {}^t\hat{\boldsymbol{\mu}} = {}^t\hat{\boldsymbol{\mu}}_{N_4} \tag{5.28} $$

is determined after N_4 iterations in which the variation of pose is first determined solving the linear equation system, and thereafter combined with the current estimate. The confidence for the pose estimate is easily assessed through the evaluation of the observation probability

$$ {}^t\gamma = p\big({}^tI|{}^t\hat{\boldsymbol{\mu}}_{N_4}\big) \tag{5.29} $$

for the current image.

Tracking at the top level is performed as long as the confidence ${}^t\gamma$ exceeds the lower bound θ_4^-. Once the confidence falls below the threshold, the active process switches to the lower tracking level.

6

Visual Servoing for Grasping Non-Cooperative Objects

Nowadays, the term "robot" is well established in society. Usually, robots are considered machines that operate autonomously, move around and/or manipulate objects. However, this point of view has been promoted primarily by book and film writers and is directed toward possible and less possible future capabilities of robots.

Still, the capabilities of robots are very restricted. Robots in automatisation, for instance, are autonomous but not "intelligent". The actions they perform are taught once and repeated many times. In order to succeed in their task, the objects they manipulate or interact with are static. Moreover, the positions of the objects are known a priori. Researchers have pushed forward this limitation by defining methods to localise certain classes of objects in an unknown but static environment.

A further important improvement is the capability to interact with non-static environments. The stability of control and the responsiveness of the robot to the changes represent the primary issues of this task. The topic has been investigated in a field known as *visual servoing* for many years. Analogously to the problem of object localisation and tracking, one challenge of visual servoing consists in the classes of objects that can be handled. Visual servoing applications have been successfully demonstrated for complex polyhedral objects in 6 degrees of freedom (DoF). So far, however, successful interaction with a moving object has been reported only for either primitive objects or for complex objects with reduced degrees of freedom (DoF).

In the following chapter, visual servoing *and* grasping in 6 DoF are investigated and demonstrated. This is an application of the approaches presented in the previous chapters capable of tracking *free-form* surfaces in 6 DoF. In principle, the shape of the objects to be grasped is not restricted to certain classes, such as planes, spheres, cylinders, or composition of such primitive surfaces.

The constraint imposed by the approaches refers exclusively to the inhomogeneity of the object texture and the unambiguity of the combination of shape and texture.

A successful application demands further investigation of static aspects of robotic systems and dynamic aspects of interaction. From the former point of view, the task of following and grasping the moving object is not possible without a suitable hardware configuration. From the latter point of view, the course of perception, control, and action with respect to the target should allow seamless interaction with the object and eventually with the human moving the object.

Though the choices of robotic hardware are limited, the suitable combination of the sensors and actuators determine the success of the task. Section 6.1 elaborates the static aspects of robotic systems discussing the sensing workspaces and outlining the constraints of robot actuation. The following section 6.2 addresses the spatial arrangement of sensor and actuator, which eventually ensures that the workspaces of the single components overlap as best as possible.

In addition, the flow of robot actions with respect to the target determine the impression of the human involved in the interaction task. Hence, section 6.3 proposes high level robot control rules that fulfil the task appropriately.

The consideration on the robot workspace and high level control are finally combined with the cascade of initial object localisation, pose refinement, and tracking presented in chapter 5. Successful experiments on physical human-robot interaction are reported in the concluding section 6.4.

6.1 Layout of Sensor and Actuator Workspaces

The workspace suited for human-robot interaction is determined by the workspace of both, human and robot. In order to maximise the area suited for interaction, the robot workspace has to be designed appropriately. In particular, the space covered by the sensor and the space reached with the robot actuator have to be analysed. For a given configuration of both system components, the intersection of the single workspaces specifies the volume for visual servoing.

6.1.1 Sensing Workspace

The robot is equipped in the following with a camera as primary sensor, which allows for the contactless inspection of the environment. The sensing range of cameras depends mainly on the lenses in use. Commercially available lenses typically perform either an orthographic projection or a perspective projection of three-dimensional object points to two-dimensional coordinates of the image plane.

Lenses of the former projection type sense a cylindrical volume and guarantee constant spatial sampling of the object irrespective of its distance to the camera. However, the width of sensed area is limited to the radius of the lenses,

which typically vary between 5 mm and 100 mm. This range is not acceptable for visual servoing and grasping, and is thus not considered in the following.

Perspective camera lenses, on the other side, sense a pyramidical volume and show spatial sampling that varies with the object to camera distance. The maximally admissible distance is limited since vision methods cannot deal with infinitesimal small object images. The closest distance is also restricted since usually the visible size of the objects in question is not allowed to exceed the image size. Both the upper and lower distance limits are related to the camera lense aperture.

In the following, the camera lens settings are identified that allow to maximise the working space in accordance with the requirements of the application. For appearance-based pose estimation, the concrete choice is related to the desired lower limit of perspective distortion. The amount of perspective shortening is the main factor for the accuracy of pose estimation under out-of-plane object rotations. The bigger the distance between object and camera, the less reliable the estimation of rotation becomes.

Hereafter, the workspace is uniquely defined by four values: the near and far bound widths of the workspace ${}^{-}W_{\mathrm{x}}\,[\mathrm{m}]$ and respectively ${}^{+}W_{\mathrm{x}}\,[\mathrm{m}]$, and the near and far bound distances ${}^{-}W_{\mathrm{z}}\,[\mathrm{m}]$ and respectively ${}^{+}W_{\mathrm{z}}\,[\mathrm{m}]$. Let $O_{\mathrm{x}}\,[\mathrm{m}]$ denote the known object width[1], and let $C_{\mathrm{u}}\,[\mathrm{px}]$ denote the constant horizontal resolution of the camera image measured in pixel. Then, the near and far bound widths are determined in the following by

$$ {}^{-}W_{\mathrm{x}} = O_{\mathrm{x}} \;, \quad {}^{+}W_{\mathrm{x}} = r_{\min} \cdot C_{\mathrm{u}} \;, \tag{6.1} $$

where $r_{\min}\left[\frac{\mathrm{m}}{\mathrm{px}}\right]$ denotes the minimal required sensor resolution measured in unit length per pixel. The near and far bound distances are computed in accordance with the minimal relative perspective distortion for out of plane rotations around the vertical axis. Suppose, in the following, a full-perspective camera with optical centre $(0,0)$. Then, the relative distortion $\xi_{\min}$ of the object border point $\mathbf{O}_{\mathrm{x}} = (O_{\mathrm{x}}/2, 0, 0)$ perceived at the far bound pose $\boldsymbol{\mu}(\theta) = (0, \theta, 0, 0, 0, {}^{+}W_{\mathrm{z}})$ for a rotation around the perpendicular orientation $\theta = 0$ is specified by

$$ \xi_{\min} = \left. \frac{\left\| \frac{\partial}{\partial_\theta} p(m(\mathbf{O}_{\mathrm{x}}, \boldsymbol{\mu}(\theta))) \right\|}{\| p(m(\mathbf{O}_{\mathrm{x}}, \boldsymbol{\mu}(\theta))) \|} \right|_{\theta=0} = \frac{O_{\mathrm{x}}}{2\,{}^{+}W_{\mathrm{z}}} \;. \tag{6.2} $$

Here, $p : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ denotes the camera projection according to equation 3.13 and $m : \mathbb{R}^3 \times \mathbb{R}^6 \rightarrow \mathbb{R}^3$ corresponds to the rigid-body motion of equation 3.1. Hence, the near and far bound distances can be determined, which read

$$ {}^{+}W_{\mathrm{z}} = \frac{O_{\mathrm{x}}}{2\,\xi_{\min}} \;, \quad {}^{-}W_{\mathrm{z}} = {}^{+}W_{\mathrm{z}} \frac{{}^{-}W_{\mathrm{x}}}{{}^{+}W_{\mathrm{x}}} \;. \tag{6.3} $$

The horizontal lens aperture $\angle C\,[\mathrm{rad}]$ of the camera for the above workspace is finally given by

$$ \angle C = 2\tan^{-1}\left(2\,{}^{+}W_{\mathrm{z}}, {}^{+}W_{\mathrm{x}}\right) \;, \tag{6.4} $$

[1] or object height respectively

where $\tan^{-1}(x, y)$ computes the angle enclosing $(1, 0)$ and (x, y). In conclusion, the sensing workspace (${}^{-}W_{\mathrm{x}}$, ${}^{+}W_{\mathrm{x}}$, ${}^{-}W_{\mathrm{z}}$, ${}^{+}W_{\mathrm{z}}$) and the appropriate lens parameters $\angle C$ are determined with the above rules on the basis of the horizontal camera resolution C_{u}, the minimal required object resolution $r_{\min}$, the object width O_{x}, and the desired minimal relative perspective distortion $\xi_{\min}$.

6.1.2 Dexterous Workspace

The physical arrangement of joints and joint limits determine the ability of the robot to assume different positions and orientations with its end-effector. While the *reachable* workspace is defined by the three-dimensional volume accessible to the robot end-effector, the *dexterous* workspace is characterised by the three-dimensional volume reached by the robot in any orientation.

The workspace of interest, however, might not correspond to the dexterous workspace. Rather, the workspace depends on specific constraints of the task. In addition to the kinematics, these constraints may involve also to the exertion of forces/torques on the environment. While kinematics describes geometric relation between joint values and end-effector pose, the dynamics describes how motor torques affect joint motion. For the particular task of grasping uncooperative objects, the constraints that matter the most apply to the robot dexterous capabilities.

The dextrous workspace reflects the overall capabilities of the system irrespective of the current joint configuration. Hence, dexterity is not related to the effort needed to move the robot between two poses of the workspace. These efforts are reflected by the robot *manipulability*. Many different measures of such efforts have been proposed in literature (see [33] for a survey), where the majority of approaches considers manipulability as a local performance index. The manipulability index based on the condition number of the robot Jacobian is an exception as it has been used to determine global performance measures separately in translational and rotational space [88].

6.2 Combination of Sensor and Actuator Workspaces

The physical configuration of the sensor and actuator determines the volume of the conjoint workspace. Hence, the spatial arrangement between these system components, and furthermore between this components and the human, is of special interest in order to determine the configuration that suites the interaction task the best.

6.2.1 Sensor-Actor Configuration

Mainly, two physical configurations exist relating camera and robot. Either the camera is mounted on the robot end-effector or the camera is fixed relative to the robot base. The former is called *eye-in-hand* configuration, while the latter is known as *eye-to-hand* configuration. In the following, both types are confronted and the benefits and disadvantages are enumerated.

Eye-In-Hand

At the beginning of robotic research, articulated machines were not able to approach accurately desired metric positions. Manufacture tolerances, the heavy weight of the robot in combination with the joint elasticity caused substantial discrepancy between the desired and actual positions of the end-effector. In practice, the true dimensions and alignment of robot cells are often unknown. Both inaccuracies, however, can be alleviated by teaching the joint values for the target positions of the end-effector.

Another possibility to overcome these inaccuracies, is to incorporate sensorial input that guides and aligns the robot with the desired position. Typically, the sensors are rigidly mounted with respect to the end-effector. In this case, end-effector and camera motion are coupled. Two solutions exist that handle the rigid-body transformation between sensor and end-effector, i.e., the displacement between these two systems. Either the transformation is determined in an off-line calibration step, or the transformation is neglected at all by simply learning the sensor-readings for the desired end-effector position.

Eye-in-hand configurations have two major advantages. First, visually guided manipulation can be performed in the complete dexterous workspace since the sensing workspace moves together with the end-effector. Second, the sensor accuracy generally increases as the end-effector approaches the target. This characteristic matches the requirements of typical tasks.

Eye-in-hand setups exhibit also some drawbacks. End-effectors are compact mechatronic devices and additional hardware, such as a camera, demands extra space at the tool-centre-point (TCP). The sensor is usually attached laterally to the end-effector in order not to interfere with the active components of the effector. Such configurations impose particular prudence in collision avoidance for robot motion, and severely restrict the dextrous workspace. Moreover, special attention is required in controlling the trajectory of the TCP to prevent the loss of a moving target from the sensing volume. Image-based visual servoing approaches implicitly address this problem in optimising the path of the visual features in the image-plane as opposed to position-based approaches, which tend to minimise the length of the path in three-dimensional Euclidean space.

Eye-To-Hand

Alternatively, the camera is not attached to the end-effector but is placed at an external position from which the desired workspace can be observed partially or totally. This configuration, known as eye-to-hand or stand-alone configuration, is best known in the animal world, where motion of the extremities is independent of motion of the eyes.

Obviously, this setup increases the autonomy of sensor and end-effector, such that observations are decoupled from the taken actions. On the other hand, if the camera is fixed relative to the robot base, the sensing workspace remains static and thus the volume for conjoint sensing and manipulation is determined a priori by the configuration. In any case, the eye-to-hand configuration requires coordination between hand and eye.

Coordination is achieved either through the contemporaneous observation of both, the hand and the target, or through observations of the target only, while the pose of the manipulator is known or estimated with other sensors. In the domain of robotics, the former approach fulfils end-point closed-loop control, while the latter performs end-point open-loop control. Hence, in contrast to eye-in-hand configurations, the movement of the end-effector does not necessarily affect the information extracted from the image.

The benefits and drawbacks of end-point closed-loop and end-point open-loop control are evident. The former requires the estimation of the object pose as well the estimation of the end-effector pose. Due to the complex appearance of the end-effector and the encountered occlusion, this can be an arbitrarily difficult problem. End-point open-loop control, instead, relies on the exact knowledge about the kinematic chain between the camera and the end-effector and on the accuracy of sensor readings in the kinematic chain.

The absolute position accuracy of robot actuators has improved significantly compared to the beginnings of robotics. The increased accuracy is achieved through higher stiffness, improved position sensors accuracy, as well as improved stiffness of the motors and gears. Hence, end-point open-loop visual servoing became feasible for current robots and robot applications.

The rigid-body transformation between camera and robot constitutes part of the kinematic chain that is needed for end-point open-loop control. Hence, this transformation has to be determined prior to the application through possibly simplified observations of the end-point.

6.2.2 Human-Robot Interaction Configuration

In the following, the eye-to-hand configuration is chosen because the application benefits from an anthropomorphic arrangement of sensor and actuator. A human user quickly recognises the configuration and is willing to interact seamlessly with the robot. A larger and dynamic interaction volume obtained with an eye-in-hand configuration would not be honoured appropriately.

The sensor workspace is optimally aligned with dexterous workspace when a manipulability performance measure over the intersection of both workspace reaches its maximum. In order to avoid local maxima encountered in sequential optimisation, the space of eye-to-robot configurations needs to be sampled exhaustively. Moreover, at every position of the conjoint workspace, the manipulability index should be considered for any orientation of the robot end-effector. In the case of redundant robots, all poses of the kinematic null-space need to be considered in addition.

Here, a pragmatic approach is proposed as an alternative. The sensing workspace is manually aligned with the reachable workspace while accounting for additional constraints on the orientation. This configuration is iteratively refined checking the reachability and manipulability performance of the robot on selected paths in the sensed volume.

6.3 Visual-Servoing Control-Rules

The acceptance of robot assistants is determined by its behaviour, i.e., the course of perception, control, and action with respect to external events. Hence, high-level control rules play an important role for the specific case of human-machine interaction. The actions performed by the robot are designed in the following to communicate a *gentle* and *familiar* behaviour, which favours the acceptance by humans.

The discrimination between gentle and rude behaviour may change with cultural and social interaction habitudes. Nowadays, smooth and slow motions are accepted widely as gentle, whereas fast and unpredictable motions commonly cause discomfort or even anxieties.

Familiarity is a property that depends only to a certain degree on the cultural roots. Personal experiences have a higher impact on the definition of familiarity and unfamiliarity. Here, the complete appearance of the robot as well as its actions are subject to the individual judgements. The familiarity does not only depend on what and how actions are performed, but also which entity is performing.

On the other side, robots are currently regarded as dull and numb and corresponding actions are considered familiar. Hand in hand with the development and dissemination of robots in society, the acceptance of robots as well as the expectations in their capabilites are likely to increase. Not only in future but already nowadays the capabilities of robots and humans are continously confronted. This shows clearly the desire for intelligent robot assistants.

Familiarity depends also on the combination of actions and appearance. Notable results has been achieved in the imitation of human appearance, facial expressions, and actions by building remotely controlled humanoids [71, 103]. In the following, the focus of attention falls on the control rules for physical interaction.

6.3.1 Task-Oriented High-Level Control

The task of grasping an object carried by a human is accomplished in three phases (see figure 6.1): standby, follow, and grasp.

At the beginning of the interaction, the robot waits for the object to be detected (standby). Hence, no pose information is sent to the robot in this state, which, in consequence, remains immobile.

Once image processing detects the object and succeeds in the proper estimation of the object pose, the robot starts moving (*follow*). The robot is continously controlled by the *follow* frame F_{F} relative to the tracked object model. In order for the end-effector not to unintentionally collide with the object, the *follow* frame is not set to a frame suited to immediately grasp the object. Instead, this frame is chosen reasonably close to the object *grasp* frame.

As soon as the end-effector reaches the current *follow* pose within a predetermined distance, the robot approaches the object for grasping (*grasp*). In this period, the end-effector is commanded to move to the *grasp* frame F_{G} relative to the tracked object. The cascade of state switches culminates in sending an

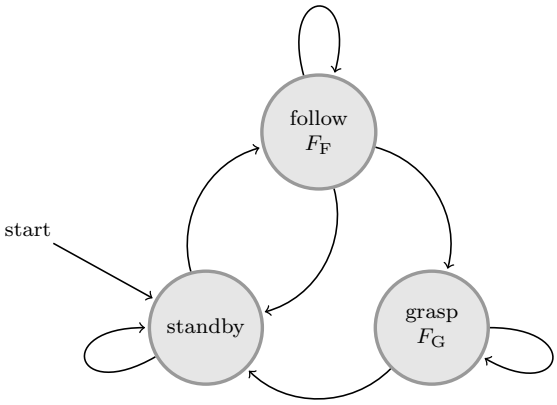

Figure 6.1: Robot state machine for grasping uncooperative objects. Here, F_{F} and F_{G} denote the *follow* frame and *grasp* frame respectively.

appropriate grasp command automatically to the end-effector as soon as the *grasp* pose is reached within a predetermined epsilon neighbourhood.

In general, an uncountable number of stabile grasps and grasp frames F_{G} exist for a certain object. In order to keep the efforts low, a suitable frame is not selected on-line in dependence of the current object pose. Instead, the frame is considered in this work to be determined off-line either by simulation or with a real experiment. In the former case, models of the object and the gripper are needed to plane the appropriate configuration. In the latter case, a grasping experiment is performed and the resulting object pose is determined w.r.t. the end-effector.

Obviously, once the object is caught its position is fixed relative to the end-effector. It is interesting to note, that object motion is restricted already before the grasp is completed. More precisely, the object motion is gradually restricted as the gripper approaches the object because the gripper naturally restricts the free space surrounding the object. This fact is fully taken into account by the states for high-level robot control.

6.3.2 Method-dependent Control

The object pose estimates needs to be mapped appropriately to robot control parameters such for instance motor acceleration, velocity or torque. The levels of the tracking cascade presented in chapter 5 generate pose estimates that differ significantly in dimensionality and accuracy.

In detail, the methods at the entry levels are based on a histogram representation of the target and hence, are not designed to accurately localise the object. Moreover, the first level detects the object on the image plane and not in three-dimensional Euclidean space. Though, the second stage tracks the object in three-dimensions, its estimate is not considered accurate. In consequence, these two stages are not connected to robot control [56]. When either of both stages is active, the robot switches to or remains in the *standby* state.

The top two layers, instead, are based on the shape-texture representation

and generate 6-DoF pose estimates in three-dimensional Euclidean space. These parameters are easily integrated into a visual servoing application (refer to figure 6.2) via position-based control [81]. Here, the robot poses for the *follow* or *grasp* frame (F_{F} and respectively F_{G}) relative to the target (F_{O}) are mapped via an inverse kinematic to the configuration space, e.g., to the robot joint angles (θ'). In view of substantial differences between the current and the desired robot pose, an interpolation module generates intermediate commands either in Cartesian space (F'_{E}) or in configuration space.

Stability of the above visual-servoing application is guaranteed since it implements open-loop control where commands sent to the robot do not affect the object as long as there is no contact.

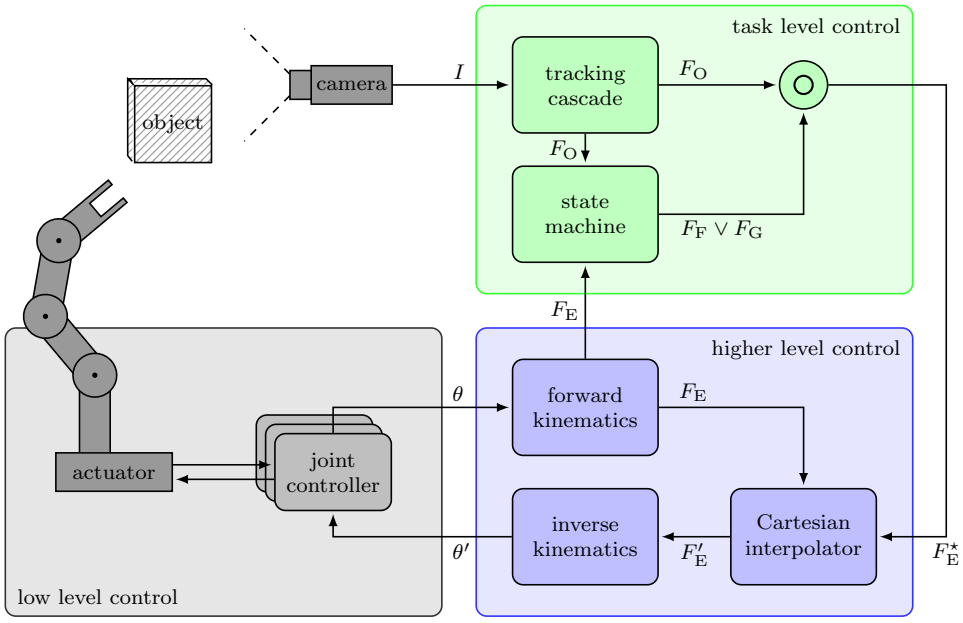

Figure 6.2: Control loop for grasping uncooperative objects consisting of control units at a low level, a path level, and the task level. The estimated object pose F_{O} is combined either with the tracking frame F_{F} or the grasping frame F_{G} to the desired end-effector frame $F^{\star}_{\mathrm{E}}$ feeding the robot controller with the desired end-effector frame. The robot controller supplies each joint controller with the desired angles.

6.3.3 Under- and Over-actuation

The transformation of a desired 6-DoF trajectory in three-dimensional Cartesian space to a trajectory in the configuration space is well posed if exactly one solution to the problem exists. Under- and Over-actuation occur if no solution or respectively more solutions can be found. Redundant robotic arms, e.g., robots with more than 6 DoF, are typically over-actuated with respect to the positioning task. Situations of over-actuation are reflected by redundancy singularities of the task [22]. The inverse kinematics has to take care of

these conditions employing additional constraints, for instance, on the power consumption, on acceleration, or aiming at the avoidance of collision.

Furthermore, the inverse kinematics is required to avoid configuration singularities. These singularities occur when the robot becomes under-actuated, i.e., when the task cannot be accomplished instantaneously.

Thus, in case of human-like 7-DoF robotic arms, the inverse kinematics has to prevent under-actuation and simultaneously manage over-actuation carefully by following additional optimality constraints. In contrast to local manipulation tasks, the application of grasping uncooperatively moving objects poses faces a new problem. The challenge is to prevent configuration singularities without knowing the future trajectory of the target in advance. Approaches that would solve the problem need to embrace two disciplines, which are considered separate til now, i.e., reactive control and motion planning. In this work, no general solution to the problem is proposed.

Instead, typical configurations of the human arm for reaching and grasping tasks are used as a priori knowledge. The human physiology favours particular postures, i.e., the lateral arrangement of the elbow. By contrast, a robot can usually assume a wider range of configurations within the space designated for the man-machine interaction. Within the manifold of configurations, the robot posture that resembles the "elbow" configuration clearly supports the task favouring

- the acceptance by humans,
- a common understanding of robot dextrous space and
- the avoidance of under-actuation.

Accordingly, this configuration aims at supporting object poses accepted by humans. The inverse kinematic does not have to deal with unnatural postures but only with those close to the initial configuration. A bias towards these joint configurations avoids the drift encountered during interaction in the direction of problematic configurations.

6.4 Evaluation

The above design criteria finally allow to integrate and evaluate the tracking cascade presented in chapter 5 in a real human-robot interaction application. The following sections describe first the setup used for this real-world application. Accordingly, the robot hardware is presented, the physical arrangement of the system components is shown, and the procedure is outlined that allows to registers these components with each other (section 6.4.1). Thereafter, the interaction experiments are explained (section 6.4.2), and finally the visual-servoing application is evaluated for the accomplishment of the reference task of grasping a bottle moved by a human being (section 6.4.3).

6.4.1 Hardware Setup

The hardware employed in the demonstrator consists of a light-weight robotic arm, an anthropomorphic robotic hand, and a digital progressive scan camera. These components are briefly described, followed by the description of their physical arrangement. Finally, the procedure for the physical registration of the camera with the robot actuator is outlined.

Components

Visual servoing is evaluated on one of the currently most sophisticated robotic arms, the DLR[2] light-weight robot-2 (LWR-2) [67] with 7 degrees of freedom and a total length of 1024 mm (for further details see section A.1).

In order to handle general objects, especially objects of daily use, the DLR anthropomorphic robotic hand II [28] is used as generic grasping tool in the setup (see section A.2 for technical details). The type of grasp performed with the anthropomorphic hand is chosen off-line according to the desired manipulation task. While pinch or precision grasps allow for object fine manipulation, a power grasp is especially suited for handling heavy objects. The latter results in particularly stable grasps and is therefore chosen in absence of subsequent requirements on manipulation.

The optical sensor is chosen according to the methods of object localisation and tracking. Though the requirements on the sensor are reasonably limited by the sampling frequency that the tracking methods sustain (cf. figures 5.2, 5.3, 5.4), the sensor capabilities are constrained in particular by the available communication resources. The combination of frame-rate, image resolution, and pixel depth dictate the requirements on the communication bandwidth. In addition, the maximal bandwidth is typically limited by the sensor chip to approximately 30 MBit/s. As a consequence, mega-pixel cameras are not considered. Instead, a camera supporting PAL resolution (768×576) images and a frame-rate up to 50 Hz is chosen. The camera models Marlin F-046C and Guppy F-046C of Allied Vision Technologies[3] are identified to fulfil these requirements (see section A.3 for the data sheet).

Workspaces and Arrangement

According to the rules established in section 6.1.1, the sensing workspace is determined based on the object width, minimally required object resolution, and the desired minimal perspective distortion. Here, objects widths of 80 mm are considered. The resolution is set according to the spatial sampling distance of the object models (cf. section 3.4.1) to 1 pixel per mm. With a minimal perspective distortion of 5% at the far bound of the sensing volume, 52° are identified as the required horizontal lens aperture. In respect to this guideline, a nominal focal length of 6 mm is chosen, which establishes a pyramidal sensing space with a horizontal and vertical aperture of approximately 56° and 43°,

[2] Deutsches Zentrum für Luft- und Raumfahrt e.V. (German Aerospace Center)
[3] http://www.alliedvisiontec.com

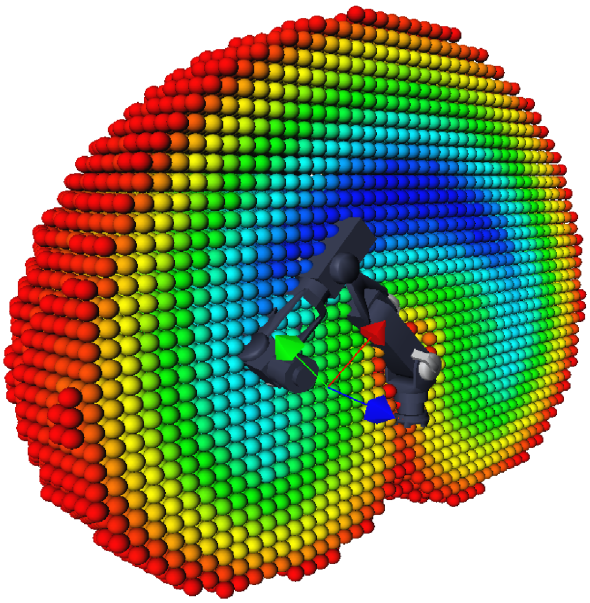

Figure 6.3: Visualisation of the 5-DoF capability map for the LWR-2. The colour scale refers to the robot ability to reach the corresponding point with different orientations. Blue indicates regions of high capability and red indicates regions of reduced capability. Courtesy of Franziska Zacharias [150].

respectively. Hence, the workspace exhibits a depth extension of approximately 660 mm at the theoretical minimal distance of 75 mm. At the far bound, a region of 780 mm $\times$ 580 mm meets the above requirements.

The dextrous workspace of the robot is identified in simulations by means of a capability map [150]. Figure 6.3 shows this map, which allows to locate appropriate regions for interaction.

Finally, the robot arm and robotic hand are combined with the camera into an eye-to-hand setup. Though the robot is mounted on a mobile, holonomic platform (figure 6.4), these additional degrees of freedom are not considered for visual servoing.

Sensor-Actuator Registration

The top two layers of the tracking cascade presented in chapter 5 require an intrinsically calibrated camera in order to accurately match and track the object. In addition, position-based visual servoing relies on an extrinsic calibration of the camera w.r.t. the robot. In an eye-to-hand configuration, this calibration allows to map pose estimates from the camera frame to the robot base frame.

The registration of camera and robot base frame is achieved by attaching a special landmark to the robot tool-centre-point (TCP). The TCP is moved to arbitrary but known positions within the conjoint sensor-actuator workspace while an appropriate module localises the landmark in the camera images. Obviously, the correspondence of the three-dimensional points in the robot frame to the two-dimensional points on the image plane is immediately given. Al-

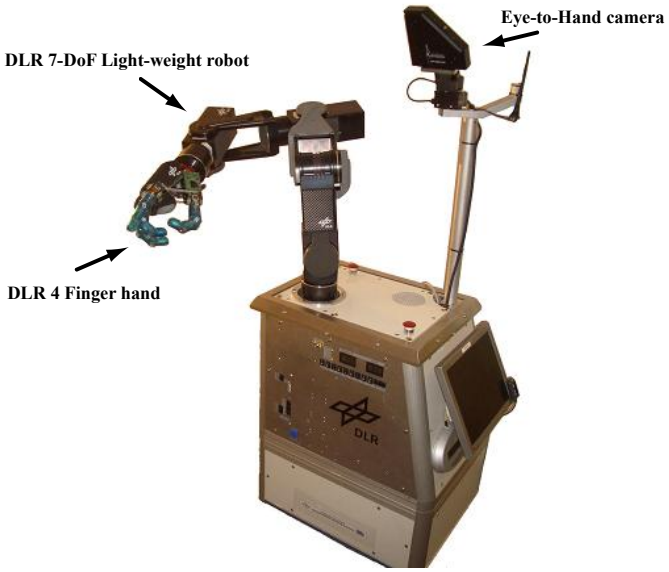

Figure 6.4: The DLR Robutler: A mobile service robot with a 7-DoF light-weight robotic arm, a 13-DoF dextrous hand, and an eye-to-hand camera system.

though a non-linear optimisation process may face the problem of local minima, a rough initial estimation is sufficient to drive minimisation to the global 6-DoF optimum of the rigid-body transformation between robot and camera. This registration process is able to compensate reasonable inaccuracies in the sensed robot pose.

Note, the above registration process can be extended to calibrate both the intrinsic and extrinsic parameters. Hereafter, intrinsic calibration is separated from the extrinsic registration since former parameters can be determined accurately with stand-alone calibration tools [122, 134] at minimal additional cost.

6.4.2 Experiments for Human-Robot Interaction

The following experiments aim at demonstrating the eligibility of the tracking cascade elaborated in chapter 5 for texture-based visual servoing. Moreover the setup is validated giving proof of successful experiments for grasping moving objects.

Accordingly, the focus is on the temporal behaviour of the system. The capability of the top tracking levels with respect to different shapes and textures has been assessed in chapter 3. Hence, the following experiments are performed with a single object suited to demonstrate human-robot interaction, i.e., a bottle object. The procedures to sample part of this object and to register it with a reference-image are accomplished as described in the experimental section of chapter 3.

The experiments make use of the conjoint workspace of robot, camera, and human being as best as possible. The purpose of these tests is to challenge the

robot in catching the object as quickly as possible. Here, a person is designated to take the object, present it to the robot (camera), and to move the object until the robot decides autonomously to fulfil the grasping action. Contemporaneously to the actions performed by the human, the robot is thought to follow a certain procedure. First the robot detects the object. Thereafter, the position and the orientation of the object is determined. The robot approaches the object by following any motion of the object. Finally, it switches to a closer pose in order to finally execute the clasping commands with the anthropomorphic robotic hand.

Ten trials with varying object velocities are carried out to evaluate the performance of tracking and servoing. In contrast to section 6.3.1, here the command to switch from the state *follow* to the state *grasp* is triggered by time in order to be able to better analyse the servoing behaviour.

6.4.3 Evaluation of Workspaces and Tracking Capabilities

The visual-servoing system is assessed by real-world experiments. These experiments are evaluated with respect to three distinct characteristics: the workspace covered by object motion and robot motion, the ability of the tracking cascade to adapt to different object velocities, and the servoing capability of the robotic system.

The experiments are performed for the complete task, that is from initial object detection and localisation, to accurate object tracking with concurrent position-based control of the robot end-effector. The tracking cascade established in chapter 5 is composed in the following of the stages histogram-based object detection and localisation in 2 DoF (section 5.4.1), histogram-based 3-DoF tracking with the Mean-Shift algorithm (section 5.4.2), shape-texture based 6-DoF pose refinement and tracking with an annealed Markov-Chain Monte-Carlo filter (section 5.5.1) and accurate shape-texture based 6-DoF tracking adopting sequential optimisation and the relaxed image-constancy assumption (section 5.5.2 and section 3.2.5).

Sensed and Actuated Workspace

The conjoint workspace for sensing and actuation plays an important role for the physical human-robot interaction. Especially the size and the location of the volume available for the interaction affects the acceptance of the application.

Both properties are measured on the basis of 10 trials of human-robot interaction. Table 6.1 reports the range of sensed object locations transformed to a desired end-effector position and the range of the actually performed robot trajectories. An interaction volume of approx. $0.4\text{m} \times 0.45\text{m} \times 0.3\text{m}$ has been exploited in the experiments. The discrepancy between the desired and actual range results from the low-pass filtering characteristic of robot motion.

This effect becomes apparent in Figure 6.5, which shows single data points of the desired and the actually performed trajectories for each of the trials. Here, the trajectories are transformed to the coordinate system of the principal axes of the desired end-effector positions.

	x			y			z		
	min	max	Δ	min	max	Δ	min	max	Δ
desired	304	736	432	-229	227	456	1031	1472	441
actual	314	698	384	-225	231	456	1163	1454	291

Table 6.1: Range of desired and actual end-effector positions with respect to the robot base frame in mm.

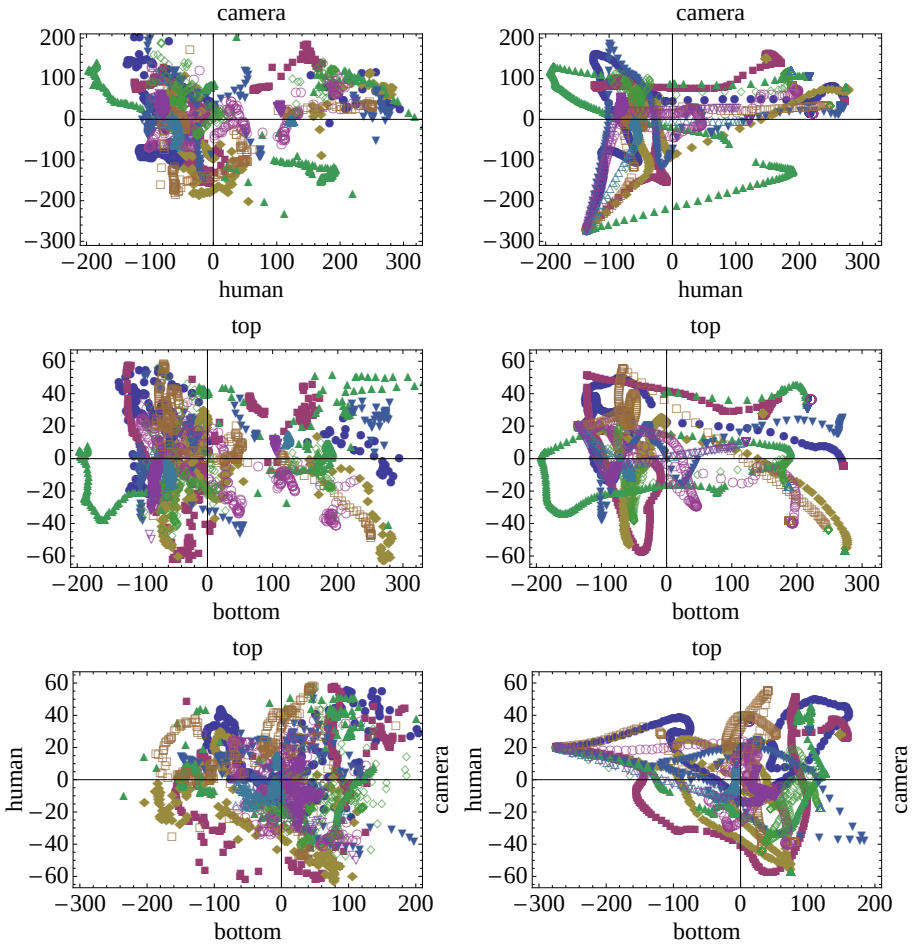

Figure 6.5: Desired (left) and actual (right) end-effector positions in mm from different viewpoints along the principal axes. The data points for 10 interaction trials are shown in a different colour/symbol combinations. From top to bottom: top, frontal, and lateral view.

Capability of Adaptation to Varying Object Velocities

According to the design of cascaded hierarchical tracking, the process of pose estimation switches from one level to the next higher or lower level in dependence on the external conditions, e.g., on object velocity, occlusion, or illumination. This capability is assessed on the basis of 10 trials of human-machine interaction, which vary in their trajectory and in the average object velocity. Figure

6.6 reports the temporal proportion of the three tracking levels of the cascade for each trial. Clearly, the proportion changes with the average velocity of the

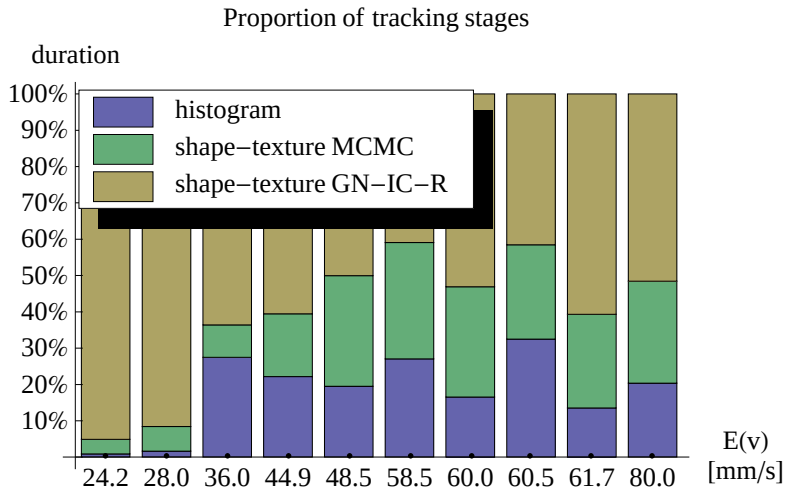

Figure 6.6: Proportion of histogram-based tracking, shape-texture based tracking with Monte-Carlo methods, and shape-texture based tracking following Gauss-Newton optimisation for each interaction trial. The trials are sorted according to the average translational velocity encountered in the shape-texture based tracking stages.

object. In particular, shape-texture based tracking by means of the Gauss-Newton algorithm is active for a shorter period of time as the object velocity increases. This observation is perfectly in line with the design of the tracking cascade since failures at the top level are compensated by the annealed Monte-Carlo approach in the first place followed by histogram-based tracking as the next fall-back level.

This tendency is clearly seen in the comparison of two particular trials, the trajectory with an average object velocity of 46 mm/s (reported in figure 6.7) and the trajectory with an average velocity of 24 mm/s (displayed in figure 6.8). While shape-texture based tracking dominates the object moving at 24 mm/s, frequent switches to histogram-based tracking are required for a higher object velocity.

Servoing Capability

The servoing capability is best observed by opposing the trajectory of the tracked object with the trajectory of the end-effector. Figure 6.7 reports the trajectories for the trial with an average object velocity of 46 mm/s, while figure 6.8 shows the trajectories for the experiment with an average velocity of 24 mm/s. In each case, the robot remains immobile for 4 seconds and subsequently starts moving as soon as shape-texture based tracking becomes active. From then on, the robot end-effector approaches the *follow* frame within the position-based control loop. After 15 seconds from the beginning, the target frame switches to

the *grasp* frame and the end-effector executes the grasp command as soon as the residual distance from the desired pose falls below a threshold.

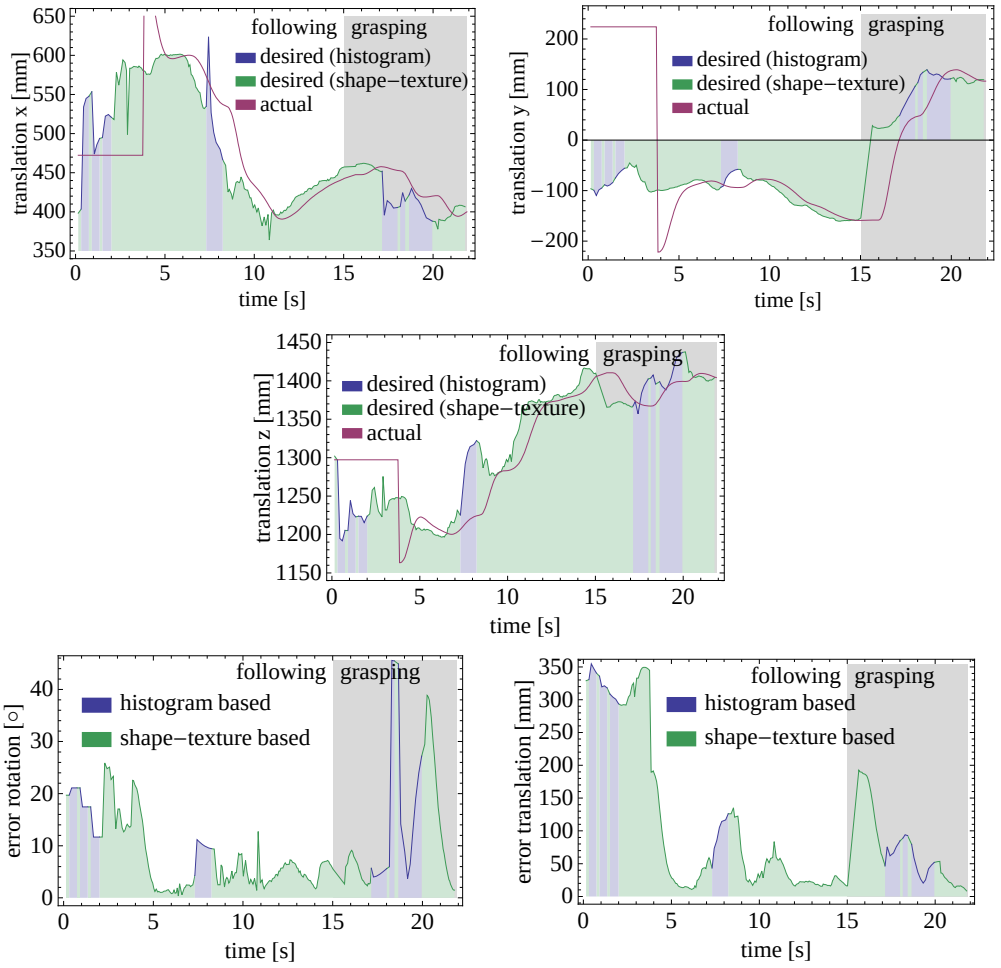

Figure 6.7: Trajectories of an exemplified visual servoing session with an average object velocity of 46 mm/s. Top & Middle: Desired and actual absolute positions in x,y and z. Bottom: Rotational and translational error between the desired and actual end-effector poses.

Both figures show that the trajectory performed by the end-effector is affected by low-pass filtering and hence, introduces a phase lag. Low-pass filtering is inherent to the system due to the robot inertia[4] and the delay of pose estimation caused by the tracking stages. The phase appears in particular during the periods of histogram-based tracking. While high-level control switches to standby and no new target poses are emitted, the robot still tends to the most recent pose.

In the phases of shape-texture based tracking, the residual rotation as well as the residual translation between the desired and the actual robot pose depend on the current object acceleration. For constant object motion, the robot control

[4]The robot inertia is a combined value of mass and underlying control.

loop is able to diminish the residuals (see figure 6.8). During the phases of acceleration, however, the residuals are growing. Eventually, the robot grasps the object when it is not accelerating.

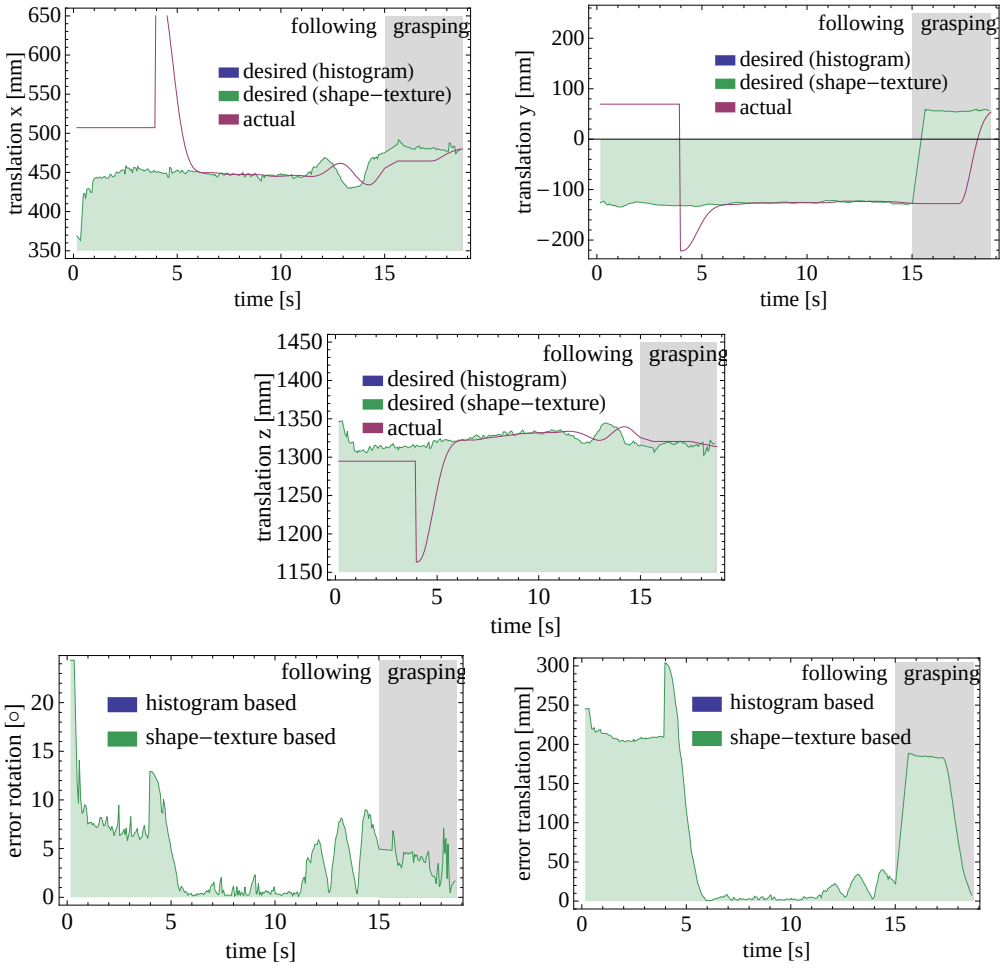

Figure 6.8: Trajectories of an exemplified visual servoing session with an average object velocity of 24 mm/s. Top & Middle: Desired and actual absolute positions in x,y and z. Bottom: Rotational and translational error between the desired and actual end-effector pose.

The final proof for the validity of the servoing approaches is given by the capability not only to follow the object movements but also to catch the object. In the set of 10 interaction trials with an adequate maximal object velocities of 80 mm/s, the application succeeded in this task.

Finally, some pictures are reported documenting the course of interaction seen from the camera point of view. Figure 6.9 shows the anthropomorphic hand approaching the bottle and finally grasping it while still in motion.

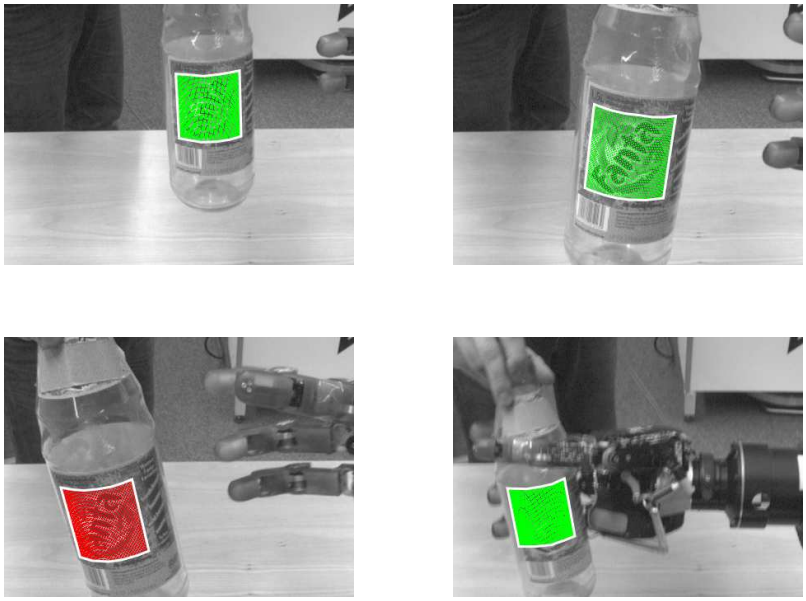

Figure 6.9: Augmented screen-shots of successful visual servoing and grasping of a bottle. The tracked model points are outlined in green when the tracking stage GN-IC-R is active, and red when the annealed Particle filter is tracking.

7 Conclusion

The thesis addressed the problem of tracking and grasping a moving object with a robotic actuator following a novel object representation model. In contrast to feature-based approaches, this representation allows for free-form and non-homogeneously textured surfaces. The work elaborated methods for real-time tracking and evaluated different techniques to cope with variations of the illumination. A tracking cascade was proposed that allows for the initial localisation of the object in 2 DoF and subsequent refinement of the pose estimate to 6 DoF. Physical configurations of the sensor and the robot were discussed with respect to their suitability for the task. Finally, an appropriate control framework for visual servoing was devised and all components are evaluated in real-world interaction experiments.

In this concluding chapter, the proposed methods and attained results are discussed (section 7.1), and motivations for following research are given (section 7.2).

7.1 Discussion

The discussion of the methods, setups, and evaluations presented in the thesis follows the order of occurrence respectively to the chapters.

7.1.1 Shape-Texture Based Tracking

The central methods of shape-texture based tracking presented in chapter 3 use a novel object model that, in its universality, has not yet been used for appearance-based tracking. In combination with the considered full-perspective projection, 6-DoF motion can be estimated for any potential non-homogeneous object or scene. Especially, the novel method for predicting the motion Jacobian increases robustness under restricted availability of computational resources. This feature is very important for real-time applications such as visual servoing.

Inherently to the used object model, self-occlusions and changes of illumination are not explicitely handled. However, self-occlusions can be trivially handled in the case of convex objects by looking at the orientation of the surface normal with respect to the camera. This technique is known in computer graphics as back-face culling. A more complex object description is needed for non-convex surfaces. Here, wire-frame models represent a possible extension. Errors in the model, i.e., errors in the surface shape, texture, or initial registration, naturally affect the tracking performance. These errors can compromise robustness especially when the motion Jacobian is predicted, and not measured, in the current image. However, this effect only occurs at viewing angles substantially different from the reference view.

In this thesis, we evaluated the methods very generally and application unspecific. The tests comprehend planar, curved, and free-form shapes as well as both richly textured and low textured surfaces. Moreover, the tests consider a large variation of object poses in the visible region of the camera. The performance is finally assessed with respect to the probability of convergence (to the true pose) under real-time conditions, i.e., with respect to the speed of object motion and the available computing power. Hence, the experiments provide a fair benchmark for the tracking performance.

7.1.2 Object-Luminance Adaptation

Shape-texture tracking is extended in chapter 4 by additional support on varying illumination conditions. In particular, three different strategies are considered: estimation based on a concurrent model of motion and illumination (section 4.2), estimation based on an adaptation to illumination effects (sections 4.1 and 4.3), and estimation based on the assumption of constant conditions of illumination (sections 3.2.1 and 3.2.4).

The evaluation showed clearly that the template-update technique (section 4.3) improves tracking performance with respect to the tracking method that uses the predicted motion Jacobian (section 3.2.4). In addition to the findings of chapter 3, the experiments show that the latter tracking method outperforms in turn the non-optimised Gauss-Newton approach (section 3.2.1) under real-time constraints, even in conditions of changing illumination. Surprisingly, neither the complementary illumination subspace approach (section 4.2) nor the adaptation with intensity-distribution normalisation (section 4.1) change the tracking performance significantly.

The evaluation is based on real trajectories with no intentional motion biases. Furthermore, the illumination in the experiments changes from moderately homogenous to strongly directed. Despite all efforts, the data-set can be only considered a subset in the domain of all appearances and appearance changes. Accordingly, the robustness and accuracy of the methods can change in relation to biases on the object or the illumination conditions.

7.1.3 Hierarchical Visual Tracking

The successful application of the tracking methods depends on the additional ability to determine the initial pose without prior estimates and to recover from occlusions or failures. Due to real-time limitations, the accomplishment of this goal requires the appropriate partitioning of the problem.

In chapter 5 the dependencies of different estimation strategies are analysed, and a tracking cascade is presented in line with the found dependencies. The adjustment of the threshold parameters inherent to the tracking cascade is left to the user. In principle however, these parameters can be determined on the basis of appropriate training data. Furthermore, at the transition between 3-DoF histogram-based localisation and tracking stage to the stage of 6-DoF shape-texture based tracking, the large range of object orientations needs to be explored, which can be an computationally expensive task. In practice, the application benefits from the actual preferences of humans on the object orientation during interaction. This knowledge lowers the computational constraints significantly.

The tracking cascade is evaluated in chapter 6 based on real-world interaction trials. The experiments validate the design criterias and give evidence of suitability for the approach.

7.1.4 Visual Servoing for Grasping

Finally, the tracking cascade is integrated in chapter 6 into an appropriate robotic demonstrator. Physical layouts of the sensor (camera) with respect to the actuator (robot) and the human are investigated for their suitability to human-robot interaction. In general, robots exhibit a limited dexterous workspace. Their dexterity typically diminishes both towards the robot base and towards the border or of their workspace. Hence, the devised configuration is specific to the particular task of human-robot interaction for grasping.

Real-world interaction experiments show the validity of the devised configuration, the control loop, the tracking cascade, and its components for the desired task of grasping a moving object from the hand of the user. In the end, these experiments prove that the task based on pure appearance-based object models is possible. This approach differs from the popular contour feature-based methods, which typically lack of support for textured objects.

Obviously, the smoothness of interaction is determined by the degree of cooperation of the participants. The duration of the interaction til completion cannot be determined a priori due to the unpredictability of the movements of the human. Interestingly, the trajectories seem to depend on the familiarity of the human with robots. However, further research is necessary to prove this affirmation.

7.2 Prospective Questions

In spite of these achievements, potential improvements can be identified and new questions arise.

The experiments of shape-texture based tracking allow to conjecture that the size of the convergence area depends on the frequency components of the surface-texture. Thus, in order to extend the area of convergence, high frequency components on the surface texture should be suppressed (cf. typical approaches based on Gaussian-pyramids). Accordingly, image processing should start at a course image resolution and terminate at the native sensor resolution. Another possibility consists of the combination of local appearance cues (features) with the presented global-appearance representation. In this way, surface point correspondences could be established over increased distances in the image. Moreover, the simultaneous use of sparse local appearance cues (features) and global appearance representation in an integrated tracking method would not only improve robustness, but would also join the originally disjunct object classes separately supported by each method.

The appearance-based methods are capable of tracking arbitrarily shaped but known objects. However, shape models could also be acquired on-line through dense stereo methods. Furthermore, the models can also be potentially constructed from a monocular image sequence following structure from motion approaches. It would be very interesting to investigate the impact of the image-constancy assumption (cf. sections 3.2.2 and 3.2.3) on an efficient method for the simultaneous and direct estimation of both structure and motion.

With respect to the illumination compensation, new and promising findings on illumination invariants exist. They have to be investigated for suitability to the appearance-based methods. Generally, and in contrast to the existing off-line computation of the illumination subspace, it could be convenient to determine both the illumination base and its parameters on-line without additional efforts. The robust handling of occlusions is not addressed in the presented approaches but can be easily accomplished with M-estimators [98]. In addition, it would be very informative to investigate to which extent M-estimators affect the area of convergence of any tracking method.

Deliberately in the present work, no physical model of object dynamics is considered in order to avoid wrong assumptions about object motion. However, the adoption of conservative assumptions can increase the maximal sustained object velocity in the average case.

It is almost impossible to track the object until the end of the grasping task because the employed gripper or the employed dextrous anthropomorphic hand often obscure the object. Moreover, the object can slip during grasping, and the commanded tool-centre-point pose might vary from the true one due to the potentially flexible robot structure. All these effects cause the actual final object pose to deviate from the estimated pose. It would be interesting to quantify the final discrepancies through external measurement systems.

In the end, this successful application can be extended to an increased number of degrees of freedom, for instance, the pan-tilt degrees of freedom of the stand-alone camera could be extended, or additional degrees of freedom in the robot base could be added in order to allow for mobile visual-servoing applications.

A Technical Data

The hardware employed in the demonstrator consists of DLR light-weight robot-2 (LWR-2), DLR anthropomorphic robotic hand, and a digital progressive scan camera. These components are briefly described in the following.

A.1 DLR Light-Weight Robotic Arm (LWR-2)

The arm [67] exhibits 7 degrees of freedom (DoF), hence, the robot is redundant for the task of assuming a certain pose with its end-effector. This redundancy can be used to optimise the robot trajectory, e.g., with respect to power consumption or velocity. The robot consists of a serial configuration of links and revolution joints with a total length of 1.024 m. A single joint consists of a electrical brushless DC motor with a Harmonic Drive[1] gear, a force-torque sensor on the output side, position encoders on the motor side and at the link side, a power-supply, and all the signal-processing components needed to locally control the position or the torque applied by the joint.

In total, the LWR-2 has a weight of 18kg and is able to manipulate up to 7kg. The maximal joint speed is 187°/s, which sufficiently resembles human capabilities. See table A.1 for further details.

A.2 DLR Robotic Hand II

This anthropomorphic robotic hand [28] features four identical fingers whereas one finger assumes the role of a thumb. Each finger is composed of three serial links with three DoF, two for the finger base and one for dependent motion of the last two links. In addition, the palm can be contracted in order to support clasping of objects. In total, the robotic hand exhibits 13 DoF. Each finger is

[1] http://www.harmonicdrive.de

overall	DoF	7
	weight	18 kg
	length	1024 mm
	max. payload	7 kg
joint	motors	7 brushless DC
	gears	7 Harmonic Drive
	max. speed	187°/s
	sensors	2 position sensors & 1 torque sensor per joint
	brake	electromagnetic safety brake
	power supply	48V DC, 20 kHz AC
	electronics	fully integrated electronics
	control	position, torque, impedance control

Table A.1: Technical data of the DLR light weight robot 2 (LWR-2).

equipped with three position and three force-torque sensors, one for each degree of freedom, and a six-dimensional force-torque sensor at the finger tip.

All electrical motors and all the electronic communication components are integrated into the dextrous hand. As a consequence the hand is roughly 1.5 times bigger than a human hand at a total weight of 1.8 kg. However, it can be attached to every robotic arm by connecting a minimal set of cables. Table A.2 summarises the technical information.

overall	number of fingers	4
	DoF	13
	weight	1.8 kg
finger	motors	3 brushless DC
	gears	3 Harmonic Drives
	DoF	3
	max. speed	360°/s
	active force	30 N
	sensors	1 position sensors & 1 torque sensor & 1 6-DoF force-torque sensor
	electronics	fully integrated electronics
palm	DoF	1

Table A.2: Technical data of the DLR hand 2.

A.3 Digital Cameras AVT Marlin F-046C and Guppy F-046C

The camera models Marlin F-046C and Guppy F-046C of Allied Vision Technologies[2] features a IIDC-DCAM protocol over an IEEE-1394a interface. The data-sheet for the former camera is reported in table A.3.

[2]http://www.alliedvisiontec.com

hardware	sensor	1/2" progressive scan CCD
	resolution	up to 780 × 582 (format 7)
	colour	colour Bayer-pattern
	A/D	10 bit
	trigger	extern/intern
	frame-rate	up to 53 Hz (format 7)
firmware	protocol	DCAM v1.30
	shutter	manual/auto (11 μs - 67 s)
	gain	manual/auto (0-16 dB)
	white-balance	manual/auto
	additional features	real-time shading correction, programmable look-up-table

Table A.3: Technical properties of the IIDC-DCAM 1.30 compliant camera Marlin F-046C of Allied Vision Technologies.

Bibliography

[1] Peter K. Allen, Aleksandar Timcenko, Billibon Yoshimi, and Paul Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Transactions on Robotics and Automation*, 9(2):152–165, 1993. 13

[2] Helder Araújo, Rodrigo L. Carceroni, and Christopher M. Brown. A fully projective formulation to improve the accuracy of Lowe's pose-estimation algorithm. *Computer Vision and Image Understanding*, 70(2):227–238, 1998. 25

[3] Klaus Arbter and Hans Burkhardt. Ein Fourier-Verfahren zur Bestimmung von Merkmalen und Schätzung der Lageparameter ebener Raumkurven. *Informationstechnik it*, 33(1):19–26, 1991. 23, 27, 29, 30

[4] Klaus Arbter, Jörg Langwald, Gerd Hirzinger, Guo-Qing Wei, and Patrick Wunsch. Proven techniques for robust visual servo control. In *Proceedings of the IEEE International Conference on Robotics and Automation 1998, ICRA 1998*. IEEE, 1998. 18, 19, 31

[5] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb 2002. 54, 56

[6] K. S. Arun, Thomas S. Huang, and Steven D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987. 24

[7] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2001*, Kauai, HI, USA, Dec 8-14 2001. IEEE Computer Society. 13, 14, 21, 22, 26, 27, 28, 29, 79

[8] Simon Baker and Iain Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004. 39, 40

[9] Eric Bardinet, Laurent D. Cohen, and Nicholas Ayache. Tracking medical 3d data with a deformable parametric model. In *Proceedings of the 4th European Conference on Computer Vision, ECCV 1996*, volume 1, pages 317–328, London, UK, 1996. Springer-Verlag. 23, 25, 28, 29

[10] Sumit Basu, Irfan A. Essa, and Alex P. Pentland. Motion regularization for model-based head tracking. In *Proceedings of the 13th International Conference on Pattern Recognition 1996, ICPR 1996*, volume 3, pages 611–616. International Association for Pattern Recognition (IAPR), Aug 1996. Vienna, Austria. 23, 25, 28

[11] Paul A. Beardsley, Andrew Zisserman, and David W. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3):235–259, 1997. 25, 31

[12] Peter N. Belhumeur and Gregory D. Hager. Tracking in 3d: Image variability decomposition for recovering object pose and illumination. *Pattern Analysis & Applications*, 2(1):82–91, April 1999. 23, 26, 27, 28, 30

[13] Peter N. Belhumeur and David J. Kriegman. What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision*, 28(3):245–260, 1998. 14, 30, 77, 78

[14] Selim Benhimane and Ezio Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2004, IROS 2004*, volume 1, pages 943–948, Sendai, Japan, Sep 28 – Oct 2 2004. IEEE/RSJ. 13, 23, 26, 27, 28, 29, 32

[15] Selim Benhimane, Ezio Malis, Patrick Rives, and Jose Raul Azinheira. Vision-based control for car platooning using homography decomposition. In *Proceedings of the IEEE International Conference on Robotics and Automation 2005, ICRA 2005*, pages 2161– 2166, Barcelona, Spain, Apr 18-22 2005. IEEE. 13, 18, 20, 32

[16] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 24

[17] Michael J. Black and Yaser Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proceedings of the 5th International Conference on Computer Vision, ICCV 1995*, pages 374–381, Cambridge, Massachusetts, USA, 1995. IEEE Computer Society. 22, 26, 32

[18] Volker Blanz, Sami Romdhani, and Thomas Vetter. Face identification across different poses and illuminations with a 3d morphable model. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, FGR 2002*, pages 192–197, Washington, DC, USA, May 20-21 2002. IEEE Computer Society. 23, 26, 28, 29, 30

[19] Tim Bodenmüller, Wolfgang Sepp, Michael Suppa, and Gerd Hirzinger. Tackling multi-sensory 3d data acquisition and fusion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2007, IROS 2007*, pages 2180–2185, San Diego, CA, USA, Oct 29–Nov 2 2007. IEEE/RSJ. 60, 83

[20] Bjørn Braathen, Marian Stewart Bartlett, Gwen Littlewort-Ford, and Javier Movellan. 3-d head pose estimation from video by nonlinear stochastic particle filtering. In *Proceedings of the 8th Joint Symposium on Neural Computation*, La Jolla, California, USA, May 19 2001. 23, 25

[21] Matthew Brand. Morphable 3d models from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2001*, volume 2, pages 456–463, Kauai, HI, USA, Dec 8-14 2001. IEEE Computer Society. 13, 23, 25, 26, 28, 29, 32

[22] Herman Bruyninckx. Serial robots. In *The Robotics WEBook (http://www.roble.info)*. Herman Bruyninckx, John Hallam, Aug 19 2005. 117

[23] José Miguel Buenaposada and Luis Baumela. Real-time tracking and estimation of plane pose. In *Proceedings of the 16th International Conference on Pattern Recognition, ICPR 2002*, volume II, pages 697–700. International Association for Pattern Recognition (IAPR), Aug 11-15 2002. Quebec, Canada. 23, 26, 27, 28, 29, 32

[24] José Miguel Buenaposada, Enrique Muñoz, and Luis Baumela. Efficient appearance-based tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Worskhop 2004, CVPRW 2004*, volume 1, page 6, 2004. 23, 26, 27, 28, 29, 30

[25] Darius Burschka and Gregory D. Hager. V-GPS(SLAM): vision-based inertial system for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation 2004, ICRA 2004*, volume 1, pages 409–415, New Orleans, LA, USA, Apr 26 - May 1 2004. IEEE. 23, 25

[26] Darius Burschka, Ming Li, Russell Taylor, and Gregory D. Hager. Scale-invariant registration of monocular stereo images to 3d surface models. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2004, IROS 2004*, pages 2581–2586, Sendai, Japan, Sep 28 – Oct 2 2004. IEEE/RSJ. 23, 25

[27] Peter J. Burt and Edward H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983. 102

[28] Jörg Butterfaß, Markus Grebenstein, Hong Liu, and Gerd Hirzinger. DLR-Hand II: Next generation of a dextrous robot hand. In *Proceedings of the IEEE International Conference on Robotics and Automation 2001, ICRA 2001*, volume 1, pages 109–114, Seoul, Korea, May 21-26 2001. IEEE. 119, 133

[29] Marco La Cascia, Stan Sclaroff, and Vassilis Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000. 13, 23, 26, 27, 28, 29, 30

[30] Bruno Cernuschi-Frias, David B. Cooper, Yi-Ping Hung, and Peter N. Belhumer. Toward a model-based bayesian theory for estimating and recognizing parameterized 3-d objects using two or more images taken from different positions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1028–1052, 1989. 13, 23, 26, 27, 28, 32

[31] François Chaumette and Seth Hutchinson. Visual servo control part I: Basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, Dec 2006. 19

[32] François Chaumette and Seth Hutchinson. Visual servo control part II: Basic approaches. *IEEE Robotics & Automation Magazine*, 14(1):109–118, Mar 2007. 19

[33] Ying chun Liu and Yue qing Yu. Survey of robot manipulability indices. In *International Conference on Intelligent Manipulation and Grasping 2004, IMG 2004*, pages 79–84, Genova, Italy, Jul 1-2 2004. University of Genova. 112

[34] Dana Cobzas and Martin Jagersand. 3d SSD tracking from uncalibrated video. In *Workshop on Spatial Coherence for Visual Motion Analysis, SCVMA 2004*, pages 25–37, 2004. 14, 23, 26, 27, 32

[35] Dana Cobzas and Martin Jagersand. Tracking and predictive display for a remote operated robot using uncalibrated video. In *Proceedings of the IEEE International Conference on Robotics and Automation 2005, ICRA 2005*, pages 1847– 1852, Barcelona, Spain, Apr 18-22 2005. IEEE. 23, 26, 27, 30

[36] Robert Collins. Mean-shift blob tracking through scale space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2003,CVPR 2003*, volume 2, pages 234–240, Madison, WI, USA, Jun 16-22 2003. IEEE Computer Society. 22, 103

[37] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2000, CVPR 2000*, volume 2, pages 142–149, Hilton Head Island, SC, USA, Jun 13-15 2000. IEEE Computer Society. 99, 101, 102

[38] Andrew. I. Comport, Danica Kragic, Éric Marchand, and François Chaumette. Robust real-time visual tracking: Comparison, theoretical analysis and performance evaluation. In *Proceedings of the IEEE International Conference on Robotics and Automation 2005, ICRA 2005*, pages 2841–2846, Barcelona, Spain, Apr 18-22 2005. IEEE. 23, 25, 28, 29, 31

[39] Andrew I. Comport, Éric Marchand, and François Chaumette. Robust model-based tracking for robot vision. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2004, IROS*

2004, volume 1, pages 692–697, Sendai, Japan, Sep 28 – Oct 2 2004. IEEE/RSJ. 13, 25, 31

[40] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. 22, 26, 27, 28, 30

[41] Timothy F. Cootes, Gavin V. Wheeler, Kevin N. Walker, and Christopher J. Taylor. View-based active appearance models. *Image and Vision Computing*, 20(9-10):657–664, Aug 2002. 22, 27, 28, 30

[42] Jason Corso, Darius Burschka, and Gregory Hager. Direct plane tracking in stereo images for mobile navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation 2003, ICRA 2003*, volume 1, pages 875– 880, Taipei, Taiwan, Sep 14-19 2003. IEEE. 23, 26, 27, 28, 29, 30

[43] Harald Craméer. *Mathematical methods of statistics*. Number 9 in Princeton mathematical series. Princeton University Press, Princeton, NJ, 1946. 40

[44] Dan Crisan and Arnaud Doucet. A survey of convergence results on particle filtering methods forpractitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746, Mar 2002. 96

[45] Frank Dellaert, Chuck Thorpe, and Sebastian Thrun. Super-resolved texture tracking of planar surface patches. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 1998, IROS 1998*, volume 1, pages 197–203, Victoria, BC, Canada, Oct 13-17 1998. IEEE/RSJ. 23, 26, 27, 28, 29, 32

[46] David Demirdjian and Trevor Darrell. Motion estimation from disparity images. In *Proceedings of the 8th IEEE International Conference On Computer Vision, ICCV 2001*, volume 1, pages 213–218, Vancouver, British Columbia, Canada, Jul 7-14 2001. IEEE Computer Society. 23, 25, 28, 29

[47] Jonathan Deutscher, Andrew Blake, and Ian Reid. Articulated body motion capture by annealed particle filtering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2000, CVPR 2000*, volume 2, pages 126–133, Hilton Head Island, SC, USA, Jun 13-15 2000. IEEE Computer Society. 57

[48] Norbert Diehl. *Methoden zur allgemeinen Bewegungsschätzung in Bildfolgen*. Number 92 in 10: Informatik/Kommunikationstechnik. VDI-Verlag, Düsseldorf, 1988. 27

[49] Fadi Dornaika, Franck Davoine, and Mo Dang. 3d head tracking by particle filters. In *Proceedings of the 5th International Workshop on Image Analysis for Multimedia Interactive Services*, Lisboa, Portugal, Apr 21-23 2004. 23, 26, 28, 29, 30, 32

[50] Tom Drummond and Roberto Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946, 2002. 23, 25, 28, 29, 31

[51] Éric Marchand and François Chaumette. Features tracking for visual servoing purpose. In Henrik Christensen Danica Kragic, editor, *Advances in Robot Vision - From Domestic Environments to Medical Applications*, page 1020, Sendai, Japan, Sep 2004. 20, 31

[52] Olivier D. Faugeras and Martial Hebert. The representation, recognition, and locating of 3-d objects. *International Journal of Robotics Research*, 5(3):27–52, 1986. 24

[53] Ronald A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London*, 222:309–368, 1922. 40

[54] Daniel Freedman and Matthew W. Turek. Illumination-invariant tracking via graph cuts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2005, CVPR 2005*, volume 2, pages 10–17, San Diego, CA, USA, Jun 20-26 2005. IEEE Computer Society. 26, 31

[55] Udo Frese, Berthold Baeuml, Steffen Haidacher, Günter Schreiber, Ingo Schäfer, Matthias Hähnle, and Gerd Hirzinger. Off-the-shelf vision for a robotic ball catcher. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2001, IROS 2001*, volume 3, pages 1623–1629, Maui, Hawaii, Oct 29–Nov 3 2001. IEEE/RSJ. 13

[56] Stefan Fuchs. Autonomes Greifen bekannter Objekte in freier Bewegung. Diplomarbeit, Technische Universität Berlin, 2006. 57, 101, 104, 116

[57] Sadaoki Furui. 50 years of progress in speech and speaker recognition. In *Proceedings of the 10th International Conference on Speech and computer, SPECOM 2005*, pages 1–9, Patras, Greece, Oct 17–19 2005. 12

[58] Christoph Gräßl, Timo Zinßer, and Heinrich Niemann. Illumination insensitive template matching with hyperplanes. In *Proceedings of the 25th Pattern Recognition Symposium, DAGM 2003*, number 2781 in Lecture Notes In Computer Science, pages 273–280, Magdeburg, Germany, Sep 10-12 2003. Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), Springer Verlag. 75

[59] Martin Gröger, Tobias Ortmaier, Wolfgang Sepp, and Gerd Hirzinger. Tracking local motion on the beating heart. In Seong K. Mun, editor, *Proceedings of SPIE Medical Imaging 2002: Visualization, Image-Guided Procedures, and Display*, volume 4681, pages 233–241, San Diego, USA, Feb 2002. The International Society for Optical Engineering (SPIE). 22

[60] A. H. Abdul Hafez and C. V. Jawahar. Target model estimation using particle filters for visual servoing. In *Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006*, volume 4, pages 651–654, Hong Kong, China, Aug 20-24 2006. International Association for Pattern Recognition (IAPR). 23, 25, 28, 29, 30

[61] Gregory D. Hager and Peter N. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1996, CVPR 1996*, pages 403–410, San Francisco, CA, USA, Jun 18-20 1996. IEEE Computer Society. 21, 26, 27, 28, 30

[62] Gregory D. Hager and Peter N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998. 14, 21, 26, 27, 28, 30, 78, 79, 80

[63] Robert Hanek and Michael Beetz. The contracting curve density algorithm: Fitting parametric curve models to images using local self-adapting separation criteria. *International Journal of Computer Vision*, 59(3):233–258, 2004. 23, 25, 28, 29, 30

[64] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, March 2004. 25

[65] Michael Harville, Ali Rahimi, Trevor Darrell, Gaile G. Gordon, and John Woodfill. 3d pose tracking with linear depth and brightness constraints. In *Proceedings of the 7th IEEE International Conference on Computer Vision, ICCV 1999*, volume 1, pages 206–213, Kerkyra, Corfu, Greece, Sep 20-25 1999. IEEE Computer Society. 23, 27, 28, 29, 31

[66] Ulrich Hillenbrand, Christian Ott, Bernhard Brunner, Christoph Borst, and Gerd Hirzinger. Towards service robots for the human environment: the Robutler. In *Proceedings of the International Conference on Mechatronics & Robotics, MechRob 2004*, pages 1497–1502, Aachen, Germany, Sep. 13–15 2004. IEEE Industrial Electronics Society. 13

[67] Gerd Hirzinger, Jörg Butterfaß, Max Fischer, Markus Grebenstein, Matthias Hähnle, Hong Liu, Ingo Schäfer, Norbert Sporer, Manfred Schedl, and Ralf Koeppe. A new generation of light-weight robot arms and multifingered hands. In *Proceedings of the Seventh International Symposium on Experimental Robotics, ISER 2000*, volume 271 of *Lecture Notes in Control and Information Sciences*, pages 569–570, Honolulu, Hawaii, USA, Dec 10–13 2000. Springer Verlag. 12, 119, 133

[68] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4:629–642, Apr 1987. 24

[69] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981. 22

[70] Michael Isard and Andrew Blake. CONDENSATION – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998. 32, 54

[71] Hiroshi Ishiguro. Android science: Toward a new cross-interdisciplinary framework. In *Proceedings of the 12th International Symposium of Robotics Research, ISSR 2005*, San Francisco, CA, USA, Oct 12–15 2005. 115

[72] Mitsuru Ito, Takeshi Sugimura, and Jun Sato. Recovering structures and motions from mutual projection of cameras. In *Proceedings of the 16th International Conference on Pattern Recognition, ICPR 2002*, volume 3, pages 676–679. International Association for Pattern Recognition (IAPR), Aug 11-15 2002. Quebec, Canada. 25

[73] Hailin Jin, Paolo Favaro, and Roberto Cipolla. Visual tracking in the presence of motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2005, CVPR 2005*, volume 2, pages 18–25, San Diego, CA, USA, Jun 20-26 2005. IEEE Computer Society. 21, 26, 27, 28, 32

[74] Biing-Hwang Juang and Lawrence R. Rabiner. Automatic speech recognition–a brief history of the technology. In *Encyclopedia of Language and Linguistics*. Elsevier, second edition edition, 2005. 12

[75] Frédéric Jurie and Michel Dhome. A simple and efficient template matching algorithm. In *Proceedings of the 8th IEEE International Conference On Computer Vision, ICCV 2001*, volume 2, pages 544–549, Vancouver, British Columbia, Canada, Jul 7-14 2001. IEEE Computer Society. 23, 26, 27, 28, 29, 32

[76] Fredrik Kahl and Didier Henrion. Globally optimal estimates for geometric reconstruction problems. In *Proceedings of the 10th IEEE International Conference on Computer Vision, ICCV 2005*, pages 978–985, Beijing, China, Oct 17-20 2005. IEEE Computer Society. 25

[77] Junhwan Kim, Vladimir Kolmogorov, and Ramin Zabih. Visual correspondence using energy minimization and mutual information. In *Proceedings of the 9th IEEE International Conference on Computer Vision, ICCV 2003*, volume 2, pages 1033– 1040, Nice, France, Oct 14-17 2003. IEEE Computer Society. 31

[78] Reinhard Klette, Karsten Schlüns, and Andreas Koschan. *Computer Vision: Three-Dimensional Data from Images*. Springer Singapore, 1998. 74

[79] Reinhard Koch. 3-d surface reconstruction from stereoscopic image sequences. In *Proceedings of the 5th International Conference on Computer Vision, ICCV 1995*, pages 109–114, Cambridge, Massachusetts, USA, 1995. IEEE Computer Society. 23, 27, 28, 29, 31

[80] Reinhard Koch. 3-d modeling of human heads from stereoscopic image sequences. In *Proceedings of the 18th DAGM Symposium Mustererkennung, DAGM 1997*, pages 169–178, Heidelberg, Germany, Sep 11-13 1996. Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), Springer Verlag. 23, 27, 28, 29, 31

[81] Danica Kragic and Henrik I. Christensen. Survey on visual servoing for manipulation. Technical Report ISRN KTH/NA/P–02/01–SE, University of Stockholm, Department of Numerical Analysis and Computing Science, 2002. 117

[82] Danica Kragic and Ville Kyrki. Initialization and system modeling in 3-d pose tracking. In *Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006*, pages 643–646, Hong Kong, China, Aug 20-24 2006. International Association for Pattern Recognition (IAPR). 18, 20, 23, 25, 29, 30

[83] Björn Krebs, Peter Sieverding, and Bernd Korn. Correct 3d matching via a fuzzy ICP algorithm for arbitrary shaped objects. In *Proceedings of the 18th DAGM Symposium Mustererkennung, DAGM 1997*, pages 521–528, Heidelberg, Germany, Sep 11-13 1996. Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), Springer Verlag. 24

[84] Ville Kyrki and Danica Kragic. Integration of model-based and model-free cues for visual object tracking in 3d. In *Proceedings of the IEEE International Conference on Robotics and Automation 2005, ICRA 2005*, pages 1554–1560, Barcelona, Spain, Apr 18-22 2005. IEEE. 25, 30

[85] Sung-Woo Lee, Bum-Jae You, and Gregory Hager. Model-based 3-d object tracking using projective invariance. In *Proceedings of the IEEE International Conference on Robotics and Automation 1999, ICRA 1999*, volume 2, pages 1589–1594, Detroit, Michigan, USA, May 1999. IEEE. 22, 25, 28, 29, 30

[86] Vincent Lepetit, Pascal Lagger, and Pascal Fua. Randomized trees for real-time keypoint recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2005, CVPR 2005*, pages 775–781, San Diego, CA, USA, Jun 20-26 2005. IEEE Computer Society. 23, 25, 28, 29, 31, 91

[87] Peihua Li, François Chaumette, and Omar Tahri. A shape tracking algorithm for visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation 2005, ICRA 2005*, pages 2847–2852, Barcelona, Spain, Apr 18-22 2005. IEEE. 18, 19, 21, 25, 28, 29, 30, 32

[88] Xinjun Liu, Jinsong Wang, Feng Gao, and Zhenlin Jin. Design of a serial-parallel 7-dof redundant anthropomorphic arm. *China Mechanical Engineering*, 2(13):101–104, 2002. 112

[89] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 99, 101

[90] Chien-Ping Lu, Gregory D. Hager, and Eric Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000. 25

[91] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, 1981. 21, 26, 27, 28, 30, 44, 76

[92] Ezio Malis. Improving vision-based control using efficient second-order minimization techniques. In *Proceedings of the IEEE International Conference on Robotics and Automation 2004, ICRA 2004*, volume 2, pages 1843–1848, New Orleans, LA, USA, Apr 26 - May 1 2004. IEEE. 18, 20, 23, 25

[93] Ezio Malis and François Chaumette. 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1):79–97, June 2000. 23, 25, 28, 30

[94] Lucie Masson, Michel Dhome, and Frédéric Jurie. Robust real time tracking of 3d objects. In *Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004*, volume 4, pages 252–255, Cambridge, UK, Aug 23-26 2004. IInternational Association for Pattern Recognition (IAPR), IEEE Computer Society. 23, 25, 28, 29, 32

[95] Iain Matthews and Simon Baker. Active appearance models revisited. *Internationl Journal of Computer Vision*, 60(2):135–164, 2004. 22, 26, 27, 28, 30

[96] Iain Matthews, Takahiro Ishikawa, and Simon Baker. The template update problem. In Richard Harvey and Andrew Bangham, editors, *Proceedings of the British Machine Vision Conference 2003, BMVC 2003*, Norwich, UK, Sep 9-11 2003. British Machine Vision Association. 31, 74, 81

[97] Gérard G. Medioni and Bastien Pesenti. Generation of a 3-d face model from one camera. In *Proceedings of the 16th International Conference on Pattern Recognition, ICPR 2002*, volume 3, pages 667–671. International Association for Pattern Recognition (IAPR), Aug 11-15 2002. Quebec, Canada. 25

[98] Peter Meer, Doron Mintz, Azriel Rosenfeld, and Dong Yoon Kim. Robust regression methods for computer vision: a review. *International Journal of Computer Vision*, 6(1):59–70, 1991. 31, 132

[99] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, Oct 2005. 99, 101

[100] Takayuki Moritani, Shinsaku Hiura, and Shinsaku Sato. Real-time object tracking without feature extraction. In *Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006*, pages 747–750, Hong Kong, China, Aug 20-24 2006. International Association for Pattern Recognition (IAPR). 23, 26, 27, 28, 29, 32

[101] Enrique Muñoz, José Miguel Buenaposada, and Luis Baumela. Efficient model-based 3d tracking of deformable objects. In *Proceedings of the 10th IEEE International Conference on Computer Vision, ICCV 2005*, pages 877–882, Beijing, China, Oct 17-20 2005. IEEE Computer Society. 23, 26, 27, 28, 29, 30

[102] Fred E. Nicodemus. Reflectance nomenclature and directional reflectance and emissivity. *Applied Optics*, 4:767–773, 1970. 73

[103] Shuichi Nishio, Hiroshi Ishiguro, and Norihiro Hagita. Geminoid: Teleoperated android of an existing person. In Armando Carlos de Pina Filho, editor, *Humanoid Robots: New Developments*, chapter 20, pages 582–591. I-Tech Education and Publishing, Vienna, Austria, 2007. 115

[104] Katja Nummiaro, Esther Koller-Meier, and Luc J. Van Gool. A color-based particle filter. In A.E.C. Pece, editor, *Proceedings of the First International Workshop on Generative-Model Based Vision, GMBV 2002*, volume 2002/01, pages 53–60. Datalogistik Institut, Kobenhavns Universitet, Jun 2 2002. 22, 27, 32, 99, 101

[105] Katja Nummiaro, Esther Koller-Meier, and Luc J. Van Gool. Object tracking with an adaptive color-based particle filter. In *Proceedings of the 24th Pattern Recognition Symposium, DAGM 2002*, number 2449 in Lecture Notes In Computer Science, pages 353–360, Zurich, Switzerland, Sep 16-18 2002. Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), Springer Verlag. 22, 27, 32

[106] Harry Nyquist. Regeneration theory. *Bell System Technical Journal*, 11:126–147, 1932. 4

[107] Giorgio Panin, Alexander Ladikos, and Alois Knoll. An efficient and robust real-time contour tracking system. In *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems, ICVS 2006*, page 44, Washington, DC, USA, 2006. IEEE Computer Society. 23, 25, 28, 29, 30, 32

[108] Seong Kee Park and In-So Kweon. Robust and direct estimation of 3-d motion and scene depth from stereo image sequences. *Pattern Recognition*, 34(9):1713–1728, Sep 2001. 27, 31

[109] James A. Paterson and Andrew W. Fitzgibbon. 3d head tracking using non-linear optimization. In Richard Harvey and Andrew Bangham, editors, *Proceedings of the British Machine Vision Conference 2003, BMVC 2003*, volume 2, pages 609–618, Norwich, UK, Sep 9-11 2003. British Machine Vision Association. 23, 26, 29, 30, 31

[110] Karl Pauwels and Marc M. Van Hulle. Robust instantaneous rigid motion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2005, CVPR 2005*, pages 980–985, San Diego, CA, USA, Jun 20-26 2005. IEEE Computer Society. 25

[111] Soo-Chang Pei and Lin-Gwo Liou. Tracking a planar patch in three-dimensional space by affine transformation in monocular and binocular vision. *Pattern Recognition*, 26(1):23–31, 1993. 21, 27, 28, 29, 30

[112] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition edition, 1992. 26

[113] Muriel Pressigout and Éric Marchand. Real time planar structure tracking for visual servoing: a contour and texture approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2005, IROS 2005*, pages 251–256, Edmonton, Alberta, Canada, Aug 2–6 2005. IEEE/RSJ. 19, 23, 25, 27, 28, 31

[114] Nicholas A. Ramey, Jason J. Corso, William W. Lau, Darius Burschka, and Gregory D. Hager. Real-time 3d surface tracking and its applications. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop 2004, CVPRW 2004*, volume 3, page 34, Washington, DC, USA, 2004. IEEE Computer Society. 23, 26, 27, 30

[115] James M. Rehg and Andrew P. Witkin. Visual tracking with deformation models. In *Proceedings of the IEEE International Conference on Robotics and Automation 1991, ICRA 1991*, volume 1, pages 844–850. IEEE, Apr 9-11 1991. Sacramento, CA, USA. 22, 26, 27, 32

[116] Sami Romdhani and Thomas Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *Proceedings of the 9th IEEE International Conference on Computer Vision, ICCV 2003*, volume 1, pages 59– 66, Nice, France, Oct 14-17 2003. IEEE Computer Society. 13, 23, 26, 27, 28, 29, 30, 31

[117] Carsten Rother and Stefan Carlsson. Linear multi view reconstruction and camera recovery. In *Proceedings of the 8th IEEE International Conference On Computer Vision, ICCV 2001*, volume 1, pages 42–50, Vancouver, British Columbia, Canada, Jul 7-14 2001. IEEE Computer Society. 25

[118] Carsten Rother and Stefan Carlsson. Linear multi view reconstruction with missing data. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Proceedings of the 7th European Conference on Computer Vision, ECCV 2002*, volume 2 of *Lecture Notes in Computer Science*, pages 309–324, Copenhagen, Denmark, May 28-31 2002. Springer. 25

[119] Tomokazu Sato, Masayuki Kanbara, Naokazu Yokoya, and Haruo Takemura. 3-d modeling of an outdoor scene by multi-baseline stereo using a long sequence of images. In *Proceedings of the 16th International Conference on Pattern Recognition, ICPR 2002*, volume 3, pages 581–584. International Association for Pattern Recognition (IAPR), Aug 11-15 2002. Quebec, Canada. 25

[120] Wolfgang Sepp. A direct method for real-time tracking in 3-d under variable illumination. In *Proceedings of the 27th Pattern Recognition Symposium, DAGM 2005*, number 3663 in Lecture Notes In Computer Science, pages 246–253, Vienna, Austria, Aug 30 - Sep 2 2005. Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), Springer Verlag. 53, 74

[121] Wolfgang Sepp. Efficient tracking in 6-dof based on the image-constancy assumption in 3-d. In *Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006*, Hong Kong, China, Aug 20-24 2006. International Association for Pattern Recognition (IAPR). 43, 50, 52

[122] Wolfgang Sepp, Stefan Fuchs, and Klaus Arbter. DLR Calibration Detection Toolbox, CalDe 2005. http://www.dlr.de/rm/callab, 2005. 62, 121

[123] Wolfgang Sepp, Stefan Fuchs, and Gerd Hirzinger. Hierarchical featureless tracking for position-based 6-dof visual servoing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2006, IROS 2006*, pages 4310–4315, Beijing, China, Oct 9–15 2006. IEEE/RSJ. 57, 101, 104

[124] Wolfgang Sepp and Gerd Hirzinger. Featureless 6 dof pose refinement from stereo images. In *Proceedings of the 16th International Conference on Pattern Recognition, ICPR 2002*, volume 4, pages 17–20. International Association for Pattern Recognition (IAPR), Aug 11-15 2002. Quebec, Canada. 23, 26, 27

[125] Wolfgang Sepp and Gerd Hirzinger. Real-time texture-based 3-d tracking. In *Proceedings of the 25th Pattern Recognition Symposium, DAGM 2003*, number 2781 in Lecture Notes In Computer Science, pages 330–337, Magdeburg, Germany, Sep 10-12 2003. Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), Springer Verlag. 52

[126] Ali Shahrokni, Tom Drummond, and Pascal Fua. Texture boundary detection for real-time tracking. In Tomás Pajdla and Jiri Matas, editors, *Proceedings of the 8th European Conference on Computer Vision, ECCV*

2004, volume 2 of *Lecture Notes in Computer Science*, pages 566–577, Prague, Czech Republic, May 11-14 2004. Springer. 23, 25, 30, 32

[127] Ying Shan, Zicheng Liu, and Zhengyou Zhang. Model-based bundle adjustment with application to face modeling. In *Proceedings of the 8th IEEE International Conference On Computer Vision, ICCV 2001*, volume 2, pages 644–651, Vancouver, British Columbia, Canada, Jul 7-14 2001. IEEE Computer Society. 25

[128] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1994, CVPR 1994*, pages 593–600, Seattle, WA, USA, Jun 21-23 1994. IEEE Computer Society. 22

[129] Heung-Yeung Shum and Richard Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *Proceedings of the 6th IEEE International Conference on Computer Vision, ICCV 1998*, pages 953–956, Bombay, India, Jan 4-7 1998. IEEE Computer Society. 26, 27

[130] Stefano Soatto, Ruggero Frezza, and Pietro Perona. Motion estimation via dynamic vision. *IEEE Transactions on Automatic Control*, 41(3):393–413, March 1996. 23, 25, 28, 29, 30, 32

[131] Jay Stavnitzky and David Capson. Multiple camera model-based 3-d visual servo. *IEEE Transactions on Robotics and Automation*, 16(6):732–739, 2000. 18, 20, 23, 25, 28, 29, 30

[132] Gideon P. Stein and Amnon Shashua. Model-based brightness constraints: On direct estimation of structure and motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):992–1015, 2000. 27, 28, 29, 31

[133] Klaus H. Strobl and Gerd Hirzinger. Optimal hand-eye calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2006, IROS 2006*, pages 4647–4653, Beijing, China, Oct 9–15 2006. IEEE/RSJ. 19

[134] Klaus H. Strobl and Christian Paredes. DLR Calibration Laboratory, CalLab 2005. http://www.dlr.de/rm/callab, 2005. 62, 121

[135] Michael Suppa, Simon Kielhöfer, Jörg Langwald, Franz Hacker, Klaus H. Strobl, and Gerd Hirzinger. The 3d-modeller: A multi-purpose vision platform. In *Proceedings of the IEEE International Conference on Robotics and Automation 2007, ICRA 2007*, pages 781–787, Rome, Italy, Apr 10-14 2007. IEEE. 60, 83

[136] Omar Tahri and François Chaumette. Application of moment invariants to visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation 2003, ICRA 2003*, volume 3, pages 4276–4281, Taipei, Taiwan, Sep 14-19 2003. IEEE. 19, 27, 28, 29, 30

[137] Omar Tahri and François Chaumette. Image moments: generic descriptors for decoupled image-based visual servo. In *Proceedings of the IEEE International Conference on Robotics and Automation 2004, ICRA 2004*, volume 2, pages 1185–1190, New Orleans, LA, USA, Apr 26 - May 1 2004. IEEE. 13, 19, 27, 28, 29

[138] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91132, Carnegie Mellon University School of Computer Science, Pittsburgh, April 1991. 22

[139] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. 25

[140] Kentaro Toyama and Gregory D. Hager. Incremental focus of attention for robust visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1996, CVPR 1996*, pages 189–195, San Francisco, CA, USA, Jun 18-20 1996. IEEE Computer Society. 14

[141] Bill Triggs. Factorization methods for projective structure and motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1996, CVPR 1996*, pages 845–851, San Francisco, CA, USA, Jun 18-20 1996. IEEE Computer Society. 25

[142] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms, ICCV 1999*, volume 1883 of *Lecture Notes In Computer Science*, pages 298–372, London, UK, 2000. Springer-Verlag. 25

[143] Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1385–1391, 2004. 23, 25, 28, 29, 31

[144] Paul Viola and William M. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, Sep 1997. 31

[145] Stefan Winkler, Patrick Wunsch, and Gerd Hirzinger. A feature map approach to real-time 3-d object pose estimation from single 2-d perspective views. In *Proceedings of the 19th DAGM Symposium Mustererkennung, DAGM 1997*, pages 129–136, Braunschweig, Germany, Sep 15-17 1997. Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM), Springer Verlag. 26, 29, 32

[146] Patrick Wunsch and Gerd Hirzinger. Registration of CAD-models to images by iterative inverse perspective matching. In *Proceedings of the 13th International Conference on Pattern Recognition 1996, ICPR 1996*,

pages 78–83. International Association for Pattern Recognition (IAPR), Aug 1996. Vienna, Austria. 23, 25, 28, 29, 31

[147] Patrick Wunsch and Gerd Hirzinger. Real-time visual tracking of 3-d objects with dynamic handling of occlusions. In *Proceedings of the IEEE International Conference on Robotics and Automation 1997, ICRA 1997*, volume 4, pages 2868–2873, Albuquerque, NM, USA, Apr 20-25 1997. IEEE. 13

[148] Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. Real-time combined 2d+3d active appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2004*, volume 2, pages 535 – 542, Washington, DC, USA, Jun 27 - Jul 2 2004. IEEE Computer Society. 13, 23, 26, 27, 28, 29, 30

[149] Yiannis Xirouhakis and Anastasios Delopoulos. Least squares estimation of 3d shape and motion of rigid objects from their orthographic projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):393–399, 2000. 25

[150] Franziska Zacharias, Christoph Borst, and Gerd Hirzinger. Capturing robot workspace structure: representing robot capabilities. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 2007, IROS 2007*, pages 3229–3236, San Diego, CA, USA, Oct 29–Nov 2 2007. IEEE/RSJ. 120