Automated Visual Inspection of Assemblies from Monocular Images

Dirk Stößel

Dipl.-Inform. Dirk Stößel
AG Angewandte Informatik
Technische Fakultät
Universität Bielefeld
email: dstoesse@techfak.uni-bielefeld.de

Abdruck der genehmigten Dissertation zur Erlangung
des akademischen Grades Doktor-Ingenieur (Dr.-Ing.).
Der Technischen Fakultät der Universität Bielefeld
am 07.03.2007 vorgelegt von Dirk Stößel,
am 21.06.2007 verteidigt und genehmigt.

Gutachter:

Prof. Dr.-Ing. Gerhard Sagerer, Universität Bielefeld
Prof. Dr.-Ing. Rainer Ott, Daimler Chrysler AG, Ulm

Prüfungsausschuss:

Prof. Dr. Holger Theisel, Universität Bielefeld
Prof. Dr.-Ing. Gerhard Sagerer, Universität Bielefeld
Prof. Dr.-Ing. Rainer Ott, Daimler Chrysler AG, Ulm
Dr. rer. nat. Robert Haschke, Universität Bielefeld

Gedruckt auf alterungsbeständigem Papier nach ISO 9706

Automated Visual Inspection of Assemblies from Monocular Images

Der Technischen Fakultät der Universität Bielefeld

zur Erlangung des Grades

Doktor-Ingenieur

vorgelegt von

Dirk Stößel

Bielefeld – März 2007

Acknowledgments

Writing and handing in a thesis are two different things. The former task I was able to start all alone. But to reach a final state wouldn't have been possible without the support of many others.

Most of all, I have to thank my fiancée, Johanna. It was she who had to endure my moods whenever experiments utterly failed, or whenever I got stuck writing some particularly nasty passage of this thesis. She helped me out of some very tight corners and was always there for me in times of despair. Writing this thesis would also have been impossible without the other members of my family. They have backed up all my decisions and given me strength to keep on going. Finally, much moral support was given by all my friends, many of which know me since my childhood days. A close friend, Hamudi Hlihel, has also been a great help in the effort of improving the illustrations and was my most important advisor in questions of layout.

On the professional level, I first want to thank my advisor Gerhard Sagerer. I'm not sure whether I would have re-entered university in any other work group but his. Furthermore, I would like to thank Prof. Dr. Rainer Ott who agreed to be the second reviewer. His critical and encouraging remarks were a great help in improving the presentation of the key ideas within this thesis. Concerning my colleagues, I am deeply grateful for the proof-reading support of Sven Wachsmuth who was tirelessly commenting everything I was throwing at him. Also, my thanks go to Christian Thurau, Volker Wendt, Marc Hanheide and Sebastian Wrede for being the nicest office mates I can imagine, and to all my other colleagues, for being the nice bunch of lads and ladies that they are.

Abstract

Industrial part assembly has evolved significantly throughout the last decades. Together with more elaborated methods of part assembly, automated visual inspection has been refined as well and plays an important role in contemporary quality assurance efforts. Nevertheless, one of the key issues in automated visual inspection, the exact localization of objects under inspection, has so far seen little progress for the case of articulated assemblies with more than two or three rigid parts. This thesis proposes a system for the inspection of assemblies consisting of multiple rigid subparts. The system is envisioned to be part of a highly automated industrial manufacturing environment. In an offline stage, the system prepares models of rigid subparts and assemblies, given CAD data. Online, the system uses a novel kernel particle filter to localize all assembly subparts that are observed within images taken by a monocular camera.

Contents

1 Introduction

> `Industry`, *n. (...) (Polit. Econ.) Human exertion of any kind employed for the creation of value, and regarded by some as a species of capital or wealth; (Webster's 1913 Dictionary)*

Industrial manufacturing has shaped our life. Take modern vehicles like cars, for instance: We really appreciate their benefits, though our enthusiasm occasionally suffers a bit when we're stuck in a traffic jam. Nevertheless, one can hardly imagine a life without cars anymore. So, within our everyday life we really depend on modern inventions and all of us use them day by day but only few ask the question of how these creations come into existence. This thesis takes part in asking the "how" question. It is concerned with taking the industrial assembly of complex products one step ahead, by devising a visual inspection system that measures whether assemblies have been put together according to given plans.

1.1 Automated Visual Inspection in the Context of Quality Assurance

The industrial process of manufacturing has brought with it a number of problems. From an engineering point of view, problems related to product quality are among the most challenging ones. One may start by asking the simple question what quality really is. In plain words, one has achieved high quality when it's the customers returning to the shop and not the merchandise. This indicates that quality correlates with properties people generally desire, e.g. durability, maintainability or safety. However, a more exhaustive specification of properties really depends on the item whose quality is being discussed and the way it is perceived by its customers.

Once a definition of the term quality is at hand, one might ask how it can be guaranteed that the production outcome is of the desired quality. This is where engineering problems really start. Accordingly, many systematic approaches have been developed which are known in the literature as approaches of quality management[1]. They generally affect a

[1] An extensive review of past and contemporary quality management strategies is given in [Bec98].

company or organization as a whole, from management down to the shop floor level. However, within this thesis only a specific part of quality management activities will be of concern: The ones collecting evidence that quality requirements have been met, i.e. *quality assurance* procedures.

Traditional quality assurance relies on statistical quality control, in the following named SQC. As the term ”statistical” indicates, SQC is based on selecting samples from produced items which are then inspected. Inspection is carried out at a few points along an assembly line. Here, defective items are removed and eventually reworked. SQC suffers from the problem that any erroneous operational unit might be far away from the next point of inspection. As a consequence, it might be very difficult to hunt down and eliminate the error source. What is more, sampling from produced items only rationalizes the inspection task. Apart from rationalization it does not help to improve quality.

Contemporary quality assurance methods try to overcome the problems of SQC. An illustrative example is the approach proposed by Shigeo Shingo [Shi86]. With regard to terminology, he distinguishes between defects and errors. Defects are unacceptable deviations from quality requirements. They arise when errors are made within production processes which are not corrected later on. His approach ultimately aims to prevent errors from being made at all and thus to reach a defect-free production level. Shingo tries to reach this goal by re-organizing inspection. By performing self-checks, i.e. by having an operational unit inspect each item it has just worked on, errors might be discovered much faster and the cause rapidly removed. Furthermore, the defect level can be improved by employing source inspection. This means to inspect production conditions even before a new step is carried out. Production is pursued only for the case of proper preconditions. Otherwise, the underlying problem must first be removed before work is resumed. Both types of proposed checks, self-checks and source inspection, make use of Poka-Yoke devices. Poka-Yoke[2] is the synonym for a mechanism that either prevents a mistake from being made or else reveals it at a glance. In general, the term refers to mistake-proofing quality assurance procedures.

Today, Poka-Yoke approaches are employed extensively. Instead of simply sorting out produced items of bad quality, priority is now given to maintaining proper production conditions. And in the same way that quality assurance changed, inspection activities evolved, too. Traditional inspection equaled fault detection. Today, this task extends to gathering and processing information about the whole production environment and about the in- and output of operational units [BA83]. Consider for example an ignition plug (outlined in red on the very left side of Fig. 1.1) and an operational unit at which a worker is supposed to attach the plug to a connector (outlined in blue in Fig. 1.1). Imagine further that at the beginning of a new work cycle, the worker is supposed to lay out all parts needed for the next step in a special box. Inspection in this context could mean to first detect whether the special box is initially empty - if it is not, the worker

[2]The term Poka-Yoke can be directly translated to ”Defect=0”.

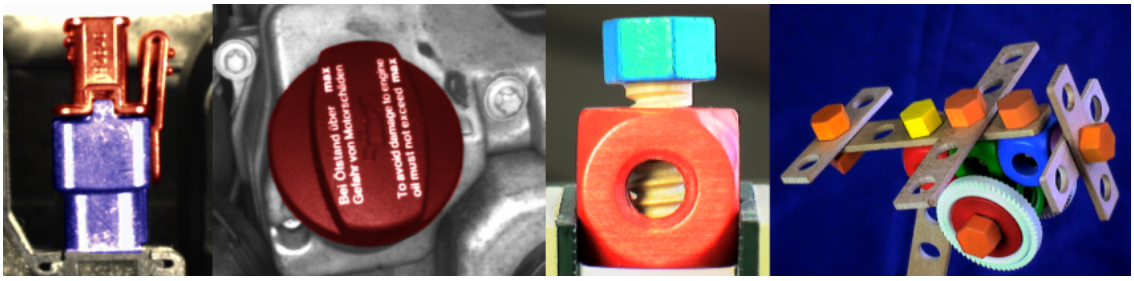


Figure 1.1: Industrial and toy assemblies. From left to right: Ignition plug (red) and connector (blue), Car engine with oil cap (highlighted in red), Toy screw and block, Toy plane. Two leftmost pictures courtesy of DaimlerChrysler AG

might have forgotten to attach the remaining parts during the last work cycle. Inspection could further include localizing the connector in order to verify that it is in the right place before the worker starts attaching the ignition plug. Finally, inspection could mean to localize the plug and the connector and to classify whether they have been put together satisfactorily.

The ignition plug example illustrates three common subtasks of inspection: Part detection, localization, and classification. Usually they are carried out by performing manual visual inspection which means to employ human operators and to rely on their ordinary vision. However, since visual inspection is a highly repetitive task it tends to exhaust humans quite fast which might in turn fail to recognize faults. Manufacturers consequently put great effort into automating visual inspection and much progress has been achieved for a variety of cases. For instance, automated systems can reliably inspect assemblies composed from two or three rigid parts (cf. Chap. 2). The ignition plug or the screw-block assembly shown in Fig. 1.1 are good examples of this simple assembly type. Nevertheless, much needs to be done with respect to assemblies composed from more than two or three parts which will be addressed by this thesis. Figure 1.1 shows examples of this more complex type of assembly, too.

Automated visual inspection, to which this thesis contributes, is a very promising and active field of research. Progress in this area, in combination with advances in management, might help to increase the overall product quality of industrial manufacturing. A very good example of the huge potential that remains to be tapped with respect to product quality is provided, again, by the automotive industry: In its annual report for 2005, the German Kraftfahrt-Bundesamt reports that about 1.4 million cars were recalled in Germany for fixing minor to critical safety problems [Imm05]. The report gives no estimate of the cost incurred by recalls but one might easily imagine that reworking cars isn't cheap. Hence, statistics like these stress the importance of rigorous quality management and assurance efforts to which this thesis contributes.

1.2 Scope and Contribution of this Thesis

This thesis proposes a system for assembly inspection from computer vision. A first overview of its architecture is illustrated in figure 1.2. The proposed system consists of two main parts which are related to the design phase and the manufacturing phase of industrial production cycles. Within the *design phase*, Computer-Aided Design (CAD) and Computer-Aided Engineering (CAE) techniques generate construction plans and prototypical realizations of new assemblies. The results are used to establish steady production processes during the *manufacturing phase*. The two distinct parts of the proposed system are designed to be integrated into the two production phases. During the design phase, the system learns from construction plans what a proper assembly is. The design information thus serves as a reference for later quality measurements. Within the manufacturing phase, the system uses the gathered knowledge in order to localize assemblies from images that are presented to it. Afterwards, a classification module is envisioned to decide whether the set of localized parts is complete and was assembled correctly. It is important to note, however, that the prototypical realization of the proposed system currently doesn't include a classification module. If, at any time, some parts of inspected assemblies change in shape or in the way they are put together, the system can be adapted to updated design plans. Concerning the sensors used for assembly inspection, the system processes images taken from industrial monocular CCD cameras which have the major advantage of being cheap and standardized and thus easy to replace in case of malfunctioning hardware.

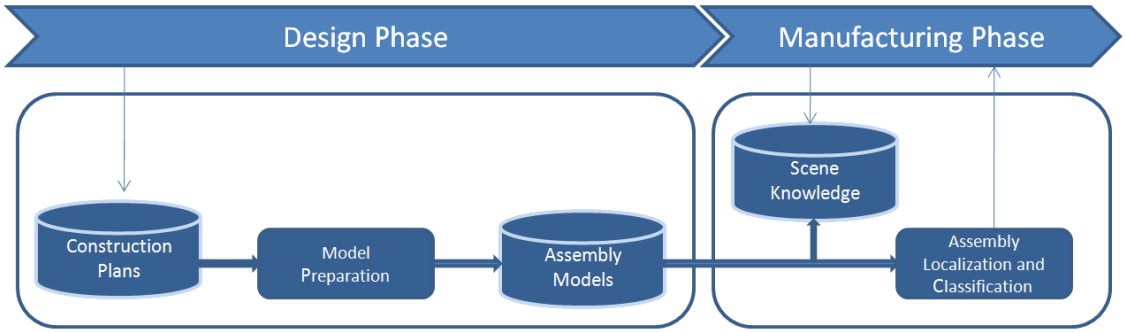


Figure 1.2: Architectural overview of the proposed automated visual inspection system

The scope of this thesis covers the techniques that were developed and implemented in order to extract part and assembly representations, in the following named *part models* and *assembly models*, from design phase information. In scope are further all methods developed and implemented for the localization of assembly parts from single images. Furthermore, the topic of classification is addressed in order to describe, how the information obtained from the localization module can be employed for the purpose of

classification. Considerations concerning viewpoint and illumination planning are restricted to the discussion of related work in Chap. 2.3 because both fields have already been extensively covered in recent publications.

Past work in the field of automated visual inspection has usually examined some aspects of either the detection, localization or classification of single rigid objects. The major contribution of this thesis is a new system which covers the full process of localization of assemblies composed from multiple rigid parts. Its design phase module employs a versatile model feature extraction stage that automatically generates part models from CAD descriptions. This thesis contributes a model feature optimization stage that extends the automatic model feature extraction by filtering out all features that don't contribute to solving the assembly pose localization task. The optimized part models can then be combined to powerful assembly models which efficiently and accurately represent feature visibility under perspective occlusion. For the assembly localization, a novel kernel particle filter (KPF) is developed. A KPF is a recent approach for recursive Bayesian Filtering that was introduced by Chang & Ansari [CA03] and Schmidt et al. [SKF06] for the purpose of visual tracking. This thesis extends kernel particle filtering such that it can be used for assembly pose localization from single monocular images. The proposed system will be thoroughly evaluated with regard to localization accuracy and precision and shown to be competitive to state-of-the-art systems that have been designed to deal with objects composed from up to three rigid parts.

1.3 Organization of this Thesis

This thesis is organized in the following way: The next chapter explains inspection subtasks in more detail. Afterwards, the related work on all issues that are identified as relevant in the context of this thesis is reviewed. It is illustrated, too, what state-of-the-art inspection systems are currently capable of. The strengths of these systems are analyzed together with their shortcomings. In Chap. 3, it is shown how part models can be generated automatically from design phase data. It is further detailed how part models can be optimized with respect to storage and put together to form assembly models. The next chapter then provides an overview of the system part responsible for assembly inspection subtasks. Here it is described in detail how the new KPF is designed and in which respect it goes beyond previous work. Furthermore, concepts for the classification of assembly pose integrity and part completeness are presented. Subsequently, the overall system performance in terms of measurement accuracy and precision is evaluated in Chap. 5. The results are compared to state-of-the-art systems. Finally, a summary of the achieved results and an outlook to future work conclude this thesis.

2 Related work on Automated Visual Inspection

The first chapter presented the general context of automated visual inspection in manufacturing. It explained why this topic is an important and active field of research. It further provided an outline of the assembly inspection system that will be presented and defined the major contribution of this thesis. However, except for some general examples, the task of automated visual inspection has not been clearly described so far and the term ”assembly” is still undefined, too. Assemblies are characterized in the following and inspection tasks are specified in more detail. Afterwards, it is explained which of the presented tasks and topics are of high relevance within the context of this thesis. The literature on relevant issues is then reviewed.

According to the literature, there are at least three important characteristics of *assemblies*. On a most general level, they are identified as man-made objects, e.g. by Bauckhage [Bau02, p.11], with typical examples such as tools, furniture or vehicles. Bauckhage also refers to a second important characteristic by observing that all these examples have been created with an inherent utility or purpose anticipated by their human designers. The notion of purpose sets assemblies apart from arbitrary pieces of work. It thus provides a means to distinguish complete from partial assemblies (the latter have not been put together far enough to fullfill a specific task). A third important aspect of assemblies that is apparent from the literature is their composition from *parts* [RW91, RP96]. Parts might be decomposable into subparts but only to a certain limit: Decomposition reaches a level where parts are atomic such that further division would yield irreversible destruction. This categorization can be simplified by regarding all parts as rigid or solid objects, as it is modeled by Requicha & Whalen [RW91]. Hence, for the remainder of this thesis we identify parts as rigid or solid and use the terms ”part”, ”rigid part”, and ”solid part” interchangeably.

In this thesis, assemblies are defined in the restricted sense of *articulated objects* as proposed by Hauck & Stöffler [HLZ97] or Byne & Anderson [BA98], i.e. objects composed from rigid parts that are connected by joints introducing internal degrees of freedom (DOF). A major motivation for this choice is the fact that only few computer vision systems have so far addressed visual inspection of articulated objects. Furthermore, inspecting assemblies in general would have meant to consider parts with a higher flexibility than the DOF defined by joints. Unfortunately, localization of flexible parts has so far been accomplished only for specific subtypes. For example, Ellenrieder [Ell05] recently

developed an approach for the inspection of near-arbitrary flexible objects like cables and tubes of a car engine. His approach could principally be incorporated to the inspection system proposed by this thesis. However, in order to keep the complexity of the task at hand from growing further, we decided that such an undertaking is more promising in the context of future work.

Automated visual inspection of assemblies can be broadly divided into six subtasks which are illustrated in Fig. 2.1. The figure shows that the subtasks can be grouped into two dependency levels. The subtasks of the upper level are known from the computer vision literature as object detection, recognition, localization, and classification. The figure visualizes that these four tasks depend strongly on the tasks of the lower level, namely model preparation and inspection planning. Regarding the upper level tasks, *Object detection* is concerned with deciding whether something important is present in an image[1]. It is typically employed early within a computer vision system as a means to focus the system's attention on important events. Object detection stages thus help to avoid wasting computational power, e.g. by filtering out irrelevant images or image regions. *Object recognition* aims to determine what objects can be seen in an image and is sometimes also termed object identification. A large body of work in the literature is dedicated to this task and the complementary problem of how to learn to recognize new objects. The task is further strongly related to determining the position and orientation of objects, i.e. the task of *object localization*. The latter is often also termed *pose estimation*. As will be shown later in this chapter, published techniques mostly understand objects to mean rigid parts that do not possess any internal variability. However, some work also addresses assemblies. The task of *classification* in the context of visual inspection is related to distinguishing unwanted items from nominal ones. It subsumes a variety of activities such as the testing of shape and dimensional accuracy, surface inspection, or checks of completeness and integrity.

As illustrated in Fig. 2.1, the performance of object detection, recognition, localization, and classification techniques depends strongly *model preparation*, i.e. on the generation of appropriate object representations within a computer vision system. On the lowest system level, objects are often explicitly described by models consisting of sets of *features*[2]. Regarding this term Ji & Marefat [JM97, p.266] note that "there is no universally accepted definition of features. In fact, this has been one of the difficulties researchers have faced in this area". The difficulty remains. In this thesis, we follow their suggestion and understand features as characteristic topological entities that together can be used to unambiguously represent an object within a certain domain and application. The definition must obviously be refined in order to really implement a computer vision system. For example, it must be specified from which source of information features will be taken. One

[1]Due to the computer vision context, this thesis only considers signals acquired by cameras. In general, any kind of computer readable signal obtained from some sensor could be used.

[2]The system presented in this thesis relies on explicit object knowledge. Consequently, approaches that encode objects implicitly will not be discussed in the following.

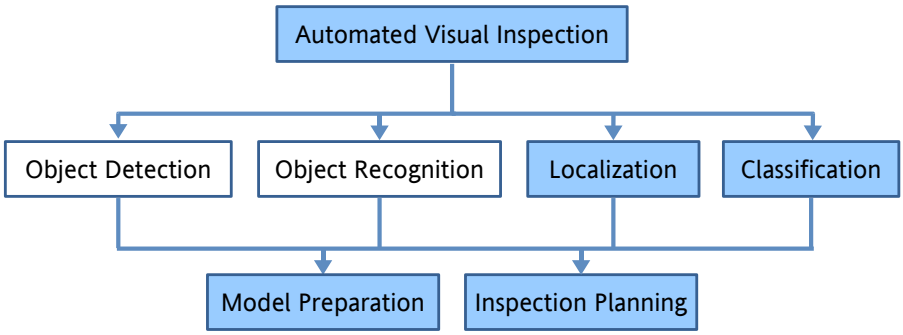


Figure 2.1: A coarse overview of automated visual inspection subtasks. Only the topics highlighted in blue are addressed in this chapter

must further describe how a system can extract features from the information presented to it. Especially in manufacturing environments, automatic feature extraction techniques are preferred to manual solutions. Once a certain feature set has been determined it is further necessary to specify how features are grouped to part and assembly models. All these topics will be dealt with in the following two sections. Finally, robust visual inspection depends on the selection of appropriate observation viewpoints and illumination conditions. Techniques that address this issue are known from the literature as methods of *inspection planning* and are discussed in this chapter, too.

Object detection, recognition, localization, and classification are carried out every time a new sensor measurement is available. In contrast to this, feature extraction, assembly modeling, and inspection planning results generally do not have to be updated whenever new observations arrive. They can thus be computed in advance which increases the performance of the remaining tasks. Consequently, precalculatable activities are categorized in the following as *offline tasks* while object detection, recognition, localization, and classification will be referred to as *online tasks*. The following sections provide a systematic review of work related to offline and online tasks. Regarding online activities, the introduction already declared that only localization and classification will be considered. The reason for this specialization is that the design of the proposed system is based on two assumptions: First, it is assumed that the system processes only images that contain assemblies. Second, it is assumed that the observed assemblies are known in advance. Based on the first assumption, the proposed system does not provide any object detection capabilities. These are considered unnecessary because within the target manufacturing environment detection tasks are usually solved by employing simple and robust devices such as photoelectric relays. Based on the second assumption, the proposed system does not address object recognition issues either. The assumption is reasonable in any environment where assemblies are presented to a camera in a highly controlled manner, e.g. with robot manipulators. In summary, ”Automated Visual Inspection” within the title of this thesis refers to the localization and classification of assemblies from monocular ima-

ges. Classification in this context means checks for part completeness and configuration integrity. Other classification activities like surface inspection are omitted in the following because they are typically carried out before parts are assembled to more complex objects.

2.1 Part Model Features

In order to give an overview of features that might be used for an assembly inspection system, this section initially presents a variety of features that have been successfully used to accomplish a diverse range of computer vision tasks. It also presents some past work on modeling feature visibility. When inspecting multi-part assemblies, feature visibility is an important topic because individual assembly parts very often occlude each other. Finally, this section considers how recent computer vision systems have acquired part models.

In order to discuss features in detail, two important categorizations are introduced in the following. The first concerns the scope which can either be global or local. *Global features* arise from objects as a whole. Due to their dependence on entire objects, most global features are sensitive to object occlusion or the presence of clutter in image observations. In contrast to this, *local features* only represent small object parts and thus can usually be determined more robustly. However, they are less distinctive, too. The second common categorization refers to the space in which features occur. It distinguishes *image features*, which arise from image observations of objects, from *model features*. The latter stem from object knowledge such as CAD data and are matched to image features in the course of computer vision procedures.

2.1.1 Prominent Features

There have been so many different features employed in computer vision systems that it would exceed the limits of this thesis by far to survey them all, even if such a survey was restricted to object localization. We therefore present only a few that have been very prominent in performing tasks like object localization or recognition.

A frequently used feature is color (for a detailed discussion see Luong [Luo93]): In Wixson & Ballard [WB89] as well as in Swain & Ballard [SB90], color histograms are used as global features for object recognition. Arnarson and Ásmundsson employ color blobs as local image features to detect bloodspots on fish. Socher [Soc97] presents the vision component of an integrated speech and image understanding system. In terms of low-level activities it performs region segmentation with a polynomial classifier that

transforms pixel color values to a set of 11 color labels. Regions of equal color labels then serve as local image features for more sophisticated object identification. In the context of these and many more applications color has been a valuable cue. However, it is not very robust to illumination changes.

Prototypes of holistic entities, termed *templates*, have successfully been used as features, too. Kölzow [Köl02] extracts templates that represent edge junctions and corners from CAD data and uses them as local features for object recognition, localization, and tracking. In [KMTB94], multiple ray-traced images of gearboxes are generated from CAD models. Each image shows a gearbox configured within a range of acceptable variations. The images are used to extract templates that encode the mean pixel intensities and intensity variations of small image regions containing assembly subparts. They are matched to real images by a multi-resolution template matching scheme. Generally, templates are reliable features as long as the represented entities do not undergo rotations within observation measurements.

A local feature that has been employed for decades in a variety of ways are edges [Shi78, Bro83, DPR92, Ros03]. As the system proposed in this thesis is mainly based on edge features, too, they are discussed in the following in some more detail. Towards the reasons for the sustained usage, Yang et al. [YMK94] note that edges are rather easy to measure from images in comparison to other model features such as slots, holes or pockets. For this task a number of edge detectors such as [Can86] and [SB97] have been developed. They exploit the fact that surface or reflective discontinuities often yield abrupt changes in measured image pixel intensities. The respective detectors have been used successfully in many computer vision systems. For instance, in the 3DPO system of Horaud & Bolles [HB86], rigid objects that are jumbled together in a pile are recognized and localized from range images by matching range image edges to mixtures of 3D circular and straight model edges. The ACRONYM system from Brooks et al. [Bro83] creates scene descriptions by using parametric models built from generalized cone features. They are matched to ribbons which are specific groups of image edge segments. Another well known example is the SCERPO system presented by Lowe [Low87] that localizes rigid objects, e.g. disposable razors, by matching 3D model lines to 2D straight image edge segments.

Unlike many prior systems, SCERPO predicts self-occlusion of model edges: Given an hypothetical object pose relative to the camera, only those model edges that would be visible from the respective viewpoint are matched to image edge segments. Each model edge is associated with a unit vector set representing hemispheres from which the edge can be seen. This approach is quite approximate but even simpler ones exist. A common heuristic, e.g. reported by Chen & Li [CL02], is based on the dot-product of a feature's surface normal and the proposed viewing direction. The feature is assumed visible for negative dot-products.

Recently, an accurate and efficient approach that models the visibility of geometrical features such as edges has been presented by Ellenrieder et al. [EKSH05]. Given a feature

reference point on the surface of an object, a unit sphere is centered on the reference point. All object surfaces are projected onto the sphere. Afterwards, the sphere is rastered at discrete azimuth and elevation angles. Note that each raster position represents a unique view direction upon the object's reference point. If any surface has been projected to a specific raster position, a *true*-value is entered into a Boolean matrix, which denotes that the reference point is occluded under the associated view direction. If a surface has been projected to the reverse view direction, the reference point is not on the object's contour which also yields a *true*-entry into the Boolean matrix. Consequently, entries of *false*-value are recorded whenever no surfaces have been projected to the considered raster positions on the unit sphere. The resulting Boolean matrix is called *visibility map* because it accurately encodes the visibility of the reference feature point. Visibility maps usually contain rather large connected regions of the same visibility status and thus can be compressed quite well by run-length encoding which yields a memory efficient representation.

Edges are versatile features but have the drawback that surface or reflective discontinuities do not always appear as intensity gradients in images, depending on the illumination type and position within a scene. Olsen & Huttenlocher [OH97] alleviate this problem by restricting themselves to *contour edges*, i.e. edges which form an object's silhouette against arbitrary backgrounds. They are more robust to illumination changes but they might still be affected by shadows. Interestingly, in any 2D view of a 3D polyhedral model the number of contour edges is usually much smaller than the total number of edges. For polyhedral models with n edges, Kettner & Welzl [KW97] provide empirical evidence that 2D views typically contain contour edges in the order of $O(\sqrt{n})$. In summary, contour edges are versatile features because they are comparatively robust to changes in illumination and yield efficient object representations. It is because of these two advantages that the system proposed by this thesis uses contour edges as primary local model features. Optionally, colored regions can be used as additional local model features.

2.1.2 Automatic Model Feature Acquisition

Once an appropriate feature set is chosen an inspection system must, by means of a feature extraction stage, acquire features that can later be grouped to object models. In highly automated manufacturing environments this task should preferably be automated as well. However, many computer vision systems rely on a manually guided model feature acquisition. This is true in particular for all the systems we have presented so far, except for the one introduced by Khawaja et al. [KMTB94]. It must be noted that feature extraction generally is a quite demanding task. For example, object models based on image features must be trained from test images (e.g. in [WB89, KMTB94, OH97]). The generation of test images is time consuming because it typically involves observing the same physical object from many different views and under varying illumination

conditions. Additionally, test images often undergo preprocessing operations like region segmentation. The results must then be carefully monitored in order to guarantee high quality training data. All this effort easily amounts to long training sessions which are unfavorable in manufacturing environments. Many feature extraction procedures therefore generate model features from CAD data. The latter are usually a byproduct of general product design workflows. In such a case, feature extraction can proceed immediately, given that the input data has the correct format. From our own painful experience we have to note, though, that CAD data conversion can corrupt model data, e.g. by introducing cracks or reversing surface normals. CAD data conversion is less time consuming than test image generation but unfortunately not as mature as the tool providers like to advertise.

Ji & Marefat [JM97] survey approaches for the machine interpretation of CAD data in manufacturing applications. They classify CAD model based feature extraction algorithms into five different categories: syntactic pattern recognition, graph-based, rule-based, volumetric methods, and evidence-based reasoning. *Syntactic pattern recognition* approaches as in [Jak82, Li96] have been applied together with extended context-free or regular right part grammars. The grammar rules generate part descriptions from geometric primitives such as line or curve segments. The rules usually couple these primitives with sweeping or revolution operators. This allows to describe 3D parts from 2D cross-sections. Given a sequence of primitives that describes a part and its generating grammar, parsers can extract features like holes, depressions or protrusions. *Graph-based* approaches as in [JC88, HCG90] work on graphs that reflect part topologies. Typically, the nodes and links of a graph correspond to edges and faces of a part's boundary representation which models objects by hierarchically storing their boundaries in terms of faces, edges, and vertices. The graph is then searched for isomorphic subgraphs that represent features like cavities or protrusions. Because searching for subgraph isomorphism is *NP* complete, heuristics are often used to initially divide the graph into small components that could contain features. In the *rule based approaches* of Henderson [Hen84] or Dong & Wozny [DW88], inference rules encode knowledge about geometrical and topological feature characteristics. An inference mechanism applies the rules to model data, employing forward chaining, backward chaining or opportunistic rule firing. *Volumetric* strategies, e.g. presented by Woo [Woo82], extract features from solid models. They systematically decompose the volume of a part into smaller volumes in order to characterize the material that must be taken away from a raw stock in order to produce a part. *Evidence-based reasoning* feature extraction proceeds in two stages. At first, feature hypotheses are generated through pattern recognition techniques. The second stage verifies features based on additional constraints. For example, Hanheide [Han01] uses scoring functions that assign weights to the edges of a boundary representation of a rigid 3D part. The scoring functions rate local edge properties like the convexity of adjoining surface patches or the angle between their surface normals. The subsequent feature veri-

fication removes edges with too little weight. It can restrict features further, e.g. to those edges that meet with others in specific corners.

The system reported in this thesis is based on Hanheide's work because his scoring functions are favorably simple compared to the heuristics or rules needed by alternative approaches. The approach is extended in various ways, e.g. with the visibility map concept of Ellenrieder et al. [EKSH05] which was sketched above. This unique combination amounts to an automated determination of contour edges for single parts. The whole model preparation stage is detailed in Chap. 3.2.

2.2 Rigid Part and Assembly Representation

Now that we have learned how objects can be characterized by features, the next question is how features can be organized to model assemblies and their parts. This question is answered in the following by presenting relevant work in the context of object recognition and localization tasks.

A good starting point for systematic considerations is the envisioned model purpose. According to the widely recognized book on object recognition by Grimson et al. [GLPH90, p.8], part and assembly models must facilitate a process that matches model to image features in an attempt to obtain feasible observation data interpretations. Pope [Pop94, p.4] presents some criteria that such models should meet: First, assembly and part models must provide an appropriate *scope* and *sensitivity*, i.e. they must describe all relevant shape characteristics and preserve object distinctions. Second, the representation should be *unique* such that only identical physical objects will have an identical model representation. Third, the models should be *stable* which means that small shape changes yield small changes in description. Finally, the chosen representation must provide data structures that support an *efficient* feature access.

Many different model representations have been proposed in the past but none satisfies the above requirements exclusively better than the others. By looking at the choice of coordinate system for the localization of model features most representations can be categorized as either object-centered or viewer-centered. Both types are illustrated in Fig. 2.2. *Object-centered* approaches attach a single local coordinate system to each rigid object or part of a represented assembly [Low89, BM98]. The coordinate system affixed to a part is used to localize all model features belonging to that specific part. Physical relationships between parts such as parts that move with respect to each other are then efficiently encoded by specifying the possible transformations between the respective coordinate systems. On the other hand, *viewer-centered* approaches represent objects with a number of different views [AKSA05, MN95]. Each view encodes object appearance from a certain perspective and slight variations to it. Therefore such approaches are also referred to as *appearance-based* representations.

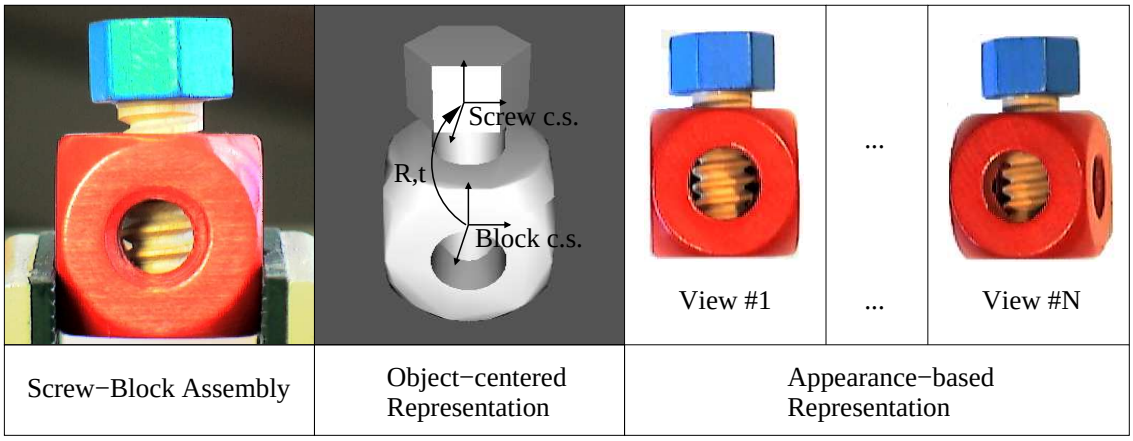


Figure 2.2: An exemplary assembly in object-centered representation (with individual screw and block models and coordinate systems) and viewer-centered representation (with N different views)

Especially for the task of object recognition, viewer-centered approaches have received much attention recently (the well-known object recognition via SIFT features proposed by Lowe [Low04] is a popular example). Peters [Pet03, p.12] even cites various articles of biological and psychological vision research in order to support the claim that "there are uncountable behavioral studies with primates that support the model of a view-based description of three-dimensional objects by our visual system". And they undoubtedly offer a number of advantages. For instance, each view automatically accounts for object regions that are hidden from the viewer. Furthermore, views can be compared to images in 2D which greatly supports fast computations. However, in order to capture appearance information accurately even simple objects like rigid parts often require many views. For multi-part assemblies the number of views grows excessively with an increasing number of internal DOF. Because object-centered approaches, on the contrary, are much more compact in terms of required storage and can be designed to account for occlusion, too, they were chosen for the system presented in this thesis. Finally, it must be noted that hybrid approaches exist that combine object-centered models with view-like appearance information [HS96, BA98]. However, they have not been investigated in the context of this thesis because the training effort involved in maintaining appearance information was considered very high and the expected performance increase neglectable.

A thorough survey of assembly representations has been carried out by Bauckhage [Bau02] in the context of Collaborative Research Center 360 (SFB 360) activities at Bielefeld University. His considerations emerged from a scenario that studied advanced human-computer interaction in the field of cooperative assembly of toy airplanes [BFF+06] (quite similar to the toy airplanes appearing as example assemblies within this thesis). The survey proposes the level of abstraction as another important dimension

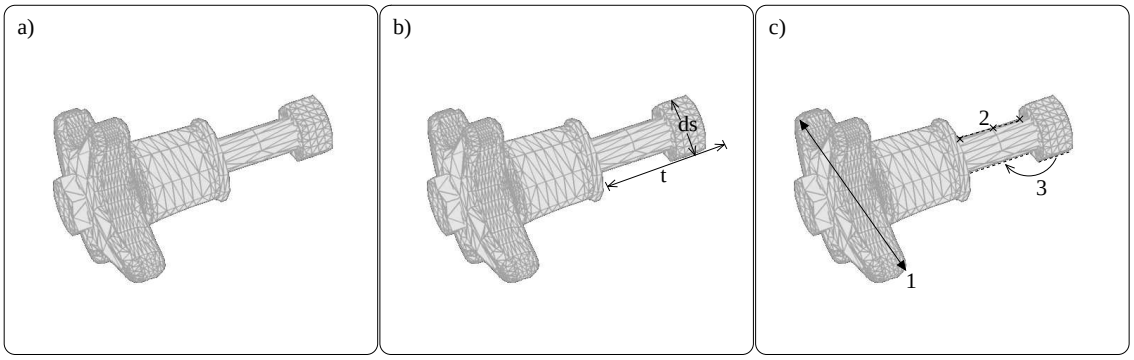


Figure 2.3: Three subtypes of object-centered geometric models. a) Pure part model. b) Parameterized model with translation and size parameter. c) Constrained model with 1) constant distance constraint, 2) co-linearity constraint and 3) parallel constraint

along which assembly representations can be categorized. *Geometric* models have a very low level of abstraction [RP96]. They are typically constructed with the help of CAD software and aim at accurately modeling the spatial position, orientation and shape of assembly parts. *Structural* models are placed on a higher level of abstraction [dMS90]. Instead of representing geometric details they denote semantic, topological or functional dependencies such as contact relations or forces between assembly parts. Bauckhage refers to *syntactic* models as the most compact form of structural assembly knowledge which is represented as grammars. He further shows that such grammars facilitate generic assembly detection. However, as the detection step is not part of the considerations of this thesis, syntactic models will not be considered further. In this thesis, the chosen representation is essentially geometric in order to convey a high amount of spatial shape information which is essential for a fine-grained pose estimation. Nevertheless, it is also attributed with structural information that encodes hierarchical dependencies and the possible motion of parts relative to each other.

Summing it up, we have learned so far that the system proposed by this thesis employs an object-centered geometric assembly representation. Let us now take a closer look at the literature reported on just this kind of representation. Three different types are apparent, namely pure part, parameterized, and constrained models. They are illustrated in Fig. 2.3.

Pure part models represent assemblies as a plain set of parts. No further information than individual part description is registered. Accordingly, part localization proceeds for each part individually, i.e. without accounting for any previously found parts. Models of this type have been used in early computer vision systems such as [Shi75, Per78, Goa86] where they were successfully employed for the pose estimation of rigid parts. However, with respect to the initially introduced representation requirements they fail to provide an appropriate scope for assembly models because they don't supply information on spatial dependencies between parts. In contrast to this, *parameterized models* con-

sist of features, the description of geometric relations between them, and free variables that parameterize different shape aspects. Such models were for example employed in [Bro81, Low89, KDN93]. The free variables model the internal DOF of the represented objects. A common problem of parameterized models is that they are difficult to generate automatically for it is unclear without further knowledge how the free parameters are selected that capture the dependencies between parts. A more general type of representation is offered by *constrained models* which were introduced by Hel-Or & Werman in [HOW96]. The constrained model of an assembly consists of a set of features per part, matrices representing the transformations from local part coordinate systems to the camera coordinate system and a collection of constraints that is given as a set of equality and inequality equations. The constraints reflect e.g. the co-linearity of part or feature locations and rotational or translational relationships between parts.

The assembly representation in this thesis is closely related to constrained models. However, Hel-Or & Werman propose the use of "hard" constraints that model how assemblies *must* be configured. But in an inspection scenario many spatial relationships might only hold for correctly assembled artifacts. Accordingly, "soft" constraints are desireable that are able to cope with misplaced parts. The latter define how parts *can* be configured in terms of physically possible or likely variations. Thus, the representation proposed in this thesis models physically feasible ranges of part locations and orientations. Instead of equality and inequality equations, the range information is encoded in a tree-like structure supporting efficient sampling from feasible part locations which is important for the kernel particle filter employed for pose estimation.

2.3 Inspection Planning

The placement of camera sensors and light sources is crucial to reliably estimate assembly poses and determine fault configurations. Badly placed cameras might miss important assembly parts and might capture some observed regions out of focus. Badly placed light sources almost always incur shadows or reflections that might distract vision algorithms. But even if it is known where cameras and light sources should be placed, the manufacturing environment itself sometimes prohibits a certain setup. *Inspection planning* activities aim to counteract problems like these. Because they have been studied extensively in the past, this thesis will not investigate them further. However, in order to give a thorough overview of visual inspection in the literature some important references are given in the following.

Among the first systems that not only considered the placement of cameras but also aimed at modeling the lighting conditions was the one reported by Cowan & Bergman [CB89]. Given polygonal CAD models of the inspected objects with explicitly marked flat target surfaces, their system first determines boundaries of 3D regions in which the placement of a camera satisfies a number of constraints. The constraints express requirements

regarding minimum spatial resolution, field of view limitations, depth of field ranges, and target surface visibility. A second stage automatically chooses a suitable aperture. Concerning the placement of lights, the authors only model one point light source. Furthermore, object surfaces only show pure Lambertian reflection plus a specular lobe. The light source is placed such that no specular reflection is turned to the camera sensor. The light placement technique was later enhanced by Cowan in [Cow91]. The proposed technique tries to position a light source such that the contrast between target surfaces is maximized. The envisioned purpose of this approach is to support edge detection operations.

Yang et al. [YMK94] created an inspection planning system that puts remarkable effort in automatically determining target features for inspection planning. They propose a unique representation that encodes objects in a boundary representation enriched with geometric and part knowledge. The latter describes semantic features (e.g. slots and holes) and possible feature interactions (e.g. intersection). The proposed knowledge representation further provides inspection planning information like the camera model. The authors use a geometric reasoning component to infer topological entities such as edges that should be extracted from images in order to measure their dimensional attributes. A sensor planning module then employs linear programming to search for camera sensor arrangements that are optimal in terms of target feature visibility and minimum path-length between sensor positions. Field of depth, spatial resolution or aperture requirements are not considered, neither do the authors model lighting.

The above mentioned systems have provided important advances in the field of inspection planning. However, their shortcomings have rendered them unsuitable for real manufacturing environments. The first system designed to explicitly meet the requirements of such environments has recently been put forward by Ellenrieder [Ell05]. Besides offering a detailed survey on state-of-the-art inspection planning, the author describes a new system that proceeds in four steps. Each step aims at optimally solving a specific subproblem of the high-dimensional planning problem. First, an *assignment phase* is used to assign target feature areas to observing perspective pinhole-cameras. The author describes a method that numerically minimizes the number of cameras while accounting for maximum feature area visibility. The method is compared to the performance of a brute force solution. Second, a *definition phase* enforces constraints regarding spatial resolution, focus, field of view, viewing angle, visibility, and many other inspection task requirements that are all modeled as convex scalar functions. Third, a *viewpoint optimization stage* employs a simplex approach to find the six-dimensional external parameters of each camera that was assigned within the assignment phase. Finally, an *illumination planning phase* optimizes internal camera parameters and illumination device positions according to criteria that are expressed as (quasi)-convex cost functions. Internal camera parameters include shutter-time, aperture, and focus setting. Surface reflection is modeled for Lambertian and Non-Lambertian materials. The whole system is successfully evaluated on synthetic and real inspection tasks.

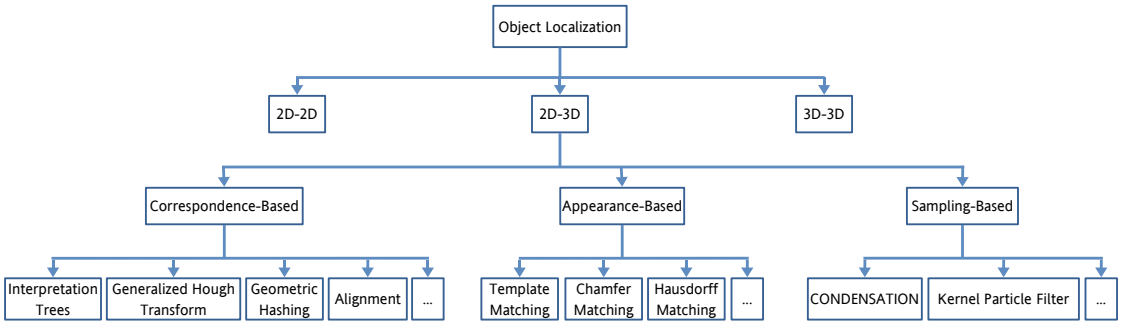


Figure 2.4: A coarse overview of the topics and concepts that are discussed in Chap. 2.4

Regarding the inspection of assemblies, Ellenrieder's system does not explicitly account for the internal degrees of freedom of articulated objects. However, extending the feature area concept used by the system to make use of knowledge on articulated objects might be a straight forward task. Interestingly, loose flexible objects like tubes or cables are already treated by the system. The author reports that the system is being introduced in real factory setup.

2.4 Object Localization

The last three sections have considered tasks that are, at least from the perspective of this thesis, essentially offline tasks: Extracting model features, composing them to part and assembly models and devising inspection plans are activities that this thesis proposes to be precomputed. This section deals with recent work on the most important online task of the suggested system, namely object localization. It proceeds by first outlining the origins of pose estimation approaches in the literature. After characterizing different categories and judging their relevance for this thesis, past work on the most important category of pose estimation techniques is surveyed in more detail. An outlook of the key topics of this section is provided in Fig. 2.4.

The determination of an object's pose from image measurements is a well studied problem, e.g. in the research fields of photogrammetry, robotics, and computer vision. In this thesis, the term *pose* denotes a set of parameters specifying a rotation and translation with respect to a reference coordinate system that bring a given object model into best accordance with observation measurements. In the photogrammetry literature the pose estimation problem is also termed *exterior orientation problem*. Work in this field dates back to the second half of the 19th century (nearly 80 classical manual solutions have been surveyed in the work of Szczepanski [Szc58]). Contributions from computer vision have obviously been published much later. First proposals like the one of Roberts [Rob65]

were put forward in the 1960s. However, according to Haralick & Joo [HJ88], the first really robust approach to computer vision based pose estimation was the RANSAC method of Fischler & Bolles [FB81] from 1981.

The literature on computer vision pose estimation techniques can be categorized according to the dimensionality of the measurement and model data, as it is done by Chen [Che91]. The resulting categories are 2D-2D, 2D-3D, and 3D-3D approaches. *2D-2D* methods use two-dimensional image measurements to localize two-dimensional models. They are considered irrelevant here because manufacturing models and their model space are three-dimensional. *3D-3D* techniques rely on three-dimensional image data to localize three-dimensional models. Except for one well known example (the 3DPO system that was briefly introduced in Chap. 2.1) these approaches are not discussed here because 2D imaging is considered a more desirable foundation for pose localization than 3D imaging. One reason for this is that hardware like an industry-standard CCD camera is much cheaper than accurate 3D imaging devices. Furthermore, a single CCD camera is comparatively easy to set up and calibrate which implies low setup cost. Another appealing fact is that the physical space requirements of a single CCD camera are comparatively small. Consequently, *2D-3D* approaches which estimate the pose of three-dimensional object models from two-dimensional image data are of major interest in the following. However, because the number of proposed systems is so large, surveying them all would by far exceed appropriate size limitations of this thesis. We will therefore only report systems that have either proposed ideas also used by our system or that illustrate frequently used techniques. For a detailed survey on object localization from computer vision until the mid 1990s refer to Goddard [God97]. A fine-grained survey of more recent methods is provided by Rosenhahn [Ros03].

2.4.1 Interpretation Trees

The separation of pose estimation activities into offline precomputations and an online part is a frequently used strategy to increase a given system's online performance and is also used by the system proposed in this thesis. It was first introduced by Goad in [Goa83]. Goad's system relies on matching straight image edges to model lines and proceeds by using search trees, also called *interpretation trees*, in a predict-observe-back-project loop: Given a candidate image edge, the system predicts possible camera positions and orientations from which a virtual camera might be looking at a specific matching model edge. Given this prediction, other model edges are back-projected to the image and compared to the measured image edges. Each matching edge reduces the considered range of camera viewpoints and lets the search step down one level within the search tree. The search terminates once a certain depth in the tree is reached, i.e. a minimal number of edges have been matched. Mismatches aid in pruning the search tree. The method is successfully tested on real images but has three major limitations.

First, the distance between the camera and the object must be accurately known, reducing the dimensionality of the pose estimation problem to 5 DOF. Second, internal visibility assumptions of the algorithm require a camera with rather small field of view. Third, it only facilitates the localization of single rigid models.

Pose localization based on interpretation trees has been used quite frequently. The PRE-MIO system of Camps et al. [CSH91] uses a Branch-and-Bound algorithm to improve the tree search when localizing single rigid objects. The 3DPO system [BHH83] combines a model-directed tree search similar to that of Goad with a low-level data-driven analysis that locates edges and groups them to circular arcs and straight lines. By evaluating *focus features* first, i.e. model features that are expected to be of strong visual salience, the average search time is reduced considerably. The method works with rigid objects, only, which can be jumbled together in a pile. Hauck et al. [HLZ97] use search trees to localize articulated objects from video images. Articulated objects are also modeled in tree-like structures which are known as *kinematic trees*. Their nodes represent rigid parts while information about part-connecting joints or about the pose of two parts relative to each other is attached to respective edges. Kinematic trees offer a compact representation of motion dependencies between parts connected by joints that is used by many computer graphics modeling tools and also by the system proposed in this thesis. However, the approach of Hauck et al. has a severe limitation which renders it inappropriate for the inspection scenario considered in this thesis: The motion of joints between any two parts of the object is restricted to one DOF.

The restriction of joint articulation that was mentioned above illustrates the limitation of interpretation tree based pose estimation. A formal analysis of this method has been published by Grimson et al. [GLPH90]. They show that the expected number of search steps is linear in the product of model and image edges, if all image edges arise from a single rigid object in the processed scene. When further objects are present, the expected number of search steps grows exponentially in the number of matches that must be established for a full scene interpretation. Especially the problem of localizing articulated objects suffers from this combinatorial explosion in the search space.

2.4.2 Generalized Hough Transform and Geometric Hashing

Two main methods for the recognition and localization of objects are the generalized hough transform and geometric hashing. The *generalized hough transform* searches in the space of pose transformations rather than in feature correspondence space. Examples are given in [Bal81] and [BB82, pp. 128-131]. Transformation parameters are represented as dimensions of an accumulator array in which votes for specific pose parameters are collected by hypothesizing matches between model and image feature subsets. As the accumulator consumes space exponential in the number of array dimensions the method does not scale to recover full poses of articulated objects at once. To dampen the memory

consumption of the generalized hough transform, Byne & Anderson [BA98] augment the geometric models of articulated objects with appearance information from real training images. For new images this information leads to a rejection of most candidate transformations before they are entered to the accumulator array. The latter is encoded with a sparse array representation to further dampen memory consumption. For each rigid part of an articulated object the system generates a number of part pose hypotheses, including false positives. The most likely full pose is then searched by an evaluation of part pose hypotheses combinations. To reduce the average time complexity of this exhaustive search which is exponential in the number of models, the appearance information is used to reject part pose hypotheses that do not match the image data well. Unfortunately, this rejection step relies on strongly colored or textured materials whereas in manufacturing environments parts might be monochrome and textureless. Together with the bad time complexity of the pose parameter search, the high cost for appearance information training and missing results on the pose estimation accuracy, this fact has led us to consider this approach inappropriate for the industrial inspection of articulated objects.

Geometric hashing was proposed by Lamdan & Wolfson [LW88]. It proceeds by first preparing a model library: During a preprocessing step, k-tupels of model features lying in planar sections of a 3D model are selected as a coordinate system basis. The remaining model feature positions are transformed to this coordinate system. The new coordinates are hashed to a table that stores all $(\textit{model}, k-\textit{tuple})$ pairs for all coordinates. For 3D models, preprocessing is carried out with $k = 4$. Online recognition or localization proceeds by selecting 4-tupels of image features and transforming the remaining image features to the respective coordinate system. The results are used to obtain votes for a certain model from the hash table. If the votes score strongly for a specific model, it is assumed present in the image and a rough object pose estimate can be retrieved. Otherwise, further 4-tupels are selected and matched against the hash table. As this method works on fixed coordinate systems defined by model and image feature subsets, it is especially well-suited for the recognition and localization of single rigid objects. It has been applied to the recognition of articulated objects in [BW91] but only for 2D models. A major limitation of this method is that it does not explicitly model occlusion between parts. In [SVD03], a different hashing approach called *Parameter-Sensitive Hashing* is used to localize models of human bodies exhibiting 13 DOF from color images. Instead of feature coordinates, the approach inserts compact representations of whole feature sets into hash tables. The drawback of this method is that it needs excessive amounts of segmented training images (150.000 for the human localization example).

Alignment or *hypothesize-and-test methods* [HU86] can be seen as an extension of methods like geometric hashing and the generalized hough transform. They start with a data-driven analyzation of a certain number of model and image feature correspondences at a time, of dimensionality sufficient to compute a complete preliminary pose. Preliminary poses are called *pose hypotheses* in order to indicate that they still need verification or rejection. The latter is provided in a model-driven fashion by matching the respective ob-

ject models to an image after transforming the model feature coordinates according to the pose hypotheses. Unlike geometric hashing or hough transform techniques, hypothesize-and-test methods have no fixed algorithmic approach to obtain pose hypotheses and quite often heuristics are used that incorporate external knowledge. For instance, Kölzow [Köl02] uses edge histogram matching for initial hypotheses generation. He further specifies rules that define when to fuse similar hypotheses or to delete unpromising ones. The rules incorporate knowledge of a motion tracking module such that hypotheses conflicting with motion estimates are deleted after some time. The approach of [BA98] mentioned above is another example for an alignment-based system (employing a generalized hough transform). In general, hypothesize-and-test methods are computationally heavy due to large numbers of hypotheses that must be verified. So far, they have mainly been used to localize rigid objects. To our knowledge, there exists no alignment-based system that would yet facilitate the visual inspection scenario targeted by this thesis.

2.4.3 Feature Correspondence vs. Object Appearance

The pose estimation problem is often separated into two subproblems [RKRS01]: The *correspondence problem* that aims at establishing a mapping between model and image features and the *spatial matching problem* that tries to find a pose parameterization minimizing some mismatch function. Accordingly, many computer vision systems use search trees, generalized hough transform, geometric hashing or other strategies only to obtain an initial solution of the correspondence problem after which a spatial fit is performed [HEG+91, DD95]. A classical example is the already mentioned SCERPO system from Lowe [Low87]. Once initial matches have been established, the 6 DOF transformation relating the model to the world coordinate system is determined by a least-squares fit. For this, Lowe linearizes the equations describing the model to image projection, assuming an affine camera model. The resulting linear equation system is solved iteratively by using Newton's method. Interestingly, this pose estimation procedure could principally determine fully articulated object poses, too, but it remains unclear how to obtain the necessary model to image feature correspondences.

Basri states in [Bas93, p. 879] that "finding the correspondence between the model and the image is the difficult problem in recognition." The vision system details presented so far might illustrate that it is also the difficult problem in localization. Especially in the case of articulated objects composed from multiple parts, a brute-force evaluation of all possible feature mappings is computationally intractable. This could explain why many systems don't solve the problem at all but let the user establish initial correspondence information manually [DC00, GBCS00]. A recent example is the work of Taylor [Tay00] where the body pose of humans is inferred from single uncalibrated images. Given a weak camera model, the absolute lengths of body segments and a manual selection of joint positions in an image, the system estimates the relative positions of joints in 3D

space with respect to a reference point. In comparison to the ground truth measured with a motion capturing system, the average reported angle deviation is about 5 degrees.

If a solution of the correspondence problem is provided, the spatial matching can be successfully determined in a variety of ways. Many published computer vision approaches are solving what Fischler & Bolles [FB81] termed the *Perspective-n-Point problem*, i.e. the spatial matching procedure relies on minimizing the distance of n corresponding model and feature points. For example, Haralick & Joo [HJ88] have compared Lowe's approach with different methods for robust least-squares fitting. Araújo et al. [ACB96] have improved Lowe's approach by exchanging the affine with a fully projective camera model. Lu et al. [LHM00] developed an algorithm that decouples the computation of rotation and translation parameters by minimizing a unique error function based on collinearity in the 3D object space.

The spatial matching problem has of course not only been treated as an Perspective-n-Point problem. Quite often, mappings between image measurements and higher order geometrical entities have been exploited as well. Published approaches were for example based on lines [PHYP93], line-plane correspondences [Hom91], polynomials of second order [GJW94], ellipses [SRTBS91], and image conics [KBG97]. Free-form objects have been considered, too, e.g. in [KVP92, ZN96]. Rosenhahn [Ros03] has recently reformulated the 2D-3D pose estimation problem as an interaction of Euclidean, projective, and conformal geometry and expressed the interaction in a conformal geometric algebra. His approach proceeds by projectively reconstructing image features and transforming the results together with model features to entities in conformal space. The latter are compared by using scaled constraint equations which is interpreted as obtaining a distance measure in the Euclidean space. Rosenhahn's approach allows to express the pose estimation based on points, lines, planes, circles, spheres, cycloidal curves, and kinematic chains in one unifying mathematical framework and to use these entities simultaneously. In summary, the spatial matching problem can be considered to be solved to a very satisfying degree while on the other hand, similar progress is still lacking for the correspondence problem, especially with regard to articulated objects.

All pose estimation methods discussed above demand a mapping between individual image and model features, i.e. they are *correspondence-based*. In contrast to this, *appearance-based* methods directly compare viewer-centered object representations or view-specific groups of model features with the content of 2D images, e.g. by template matching [KMTB94], chamfer matching [Gav00] or even neural networks [WWH97]. Ekvall et al. [EKH05] first obtain a rough pose estimate with appearance-based color histogram matching that is refined with correspondence-based techniques. However, they only deal with single rigid objects. Appearance-based object localization is particularly effective if object rotations can be restricted to camera-plane rotations or if the localized objects are completely rigid. In contrast to this, articulated objects with many parts require a very high amount of training images which is why this approach was not pursued

in the context of this thesis. Nevertheless, the system in this thesis uses a generalization of the chamfer matching technique mentioned above, namely matching by minimizing the *Hausdorff distance*, as important part of a density estimation process. Note that Hausdorff matching has been covered in detail in the book of Rucklidge [Ruc96]. The need for training images is eliminated by designing the system such that it can transform its object-centered model of an articulated object online to a viewer-centered representation.

2.4.4 Sampling-Based Pose Estimation

So far, this section presented major methods and computer vision systems for 2D-3D object localization. It also described their limitations. An important conclusion from the presented information is that the reported methods either only localize single rigid objects or else have some properties that render them inappropriate for our inspection scenario. This might explain why, regarding the vast amount of literature on pose estimation, comparatively few systems have been proposed yet to deal with visual inspection in a manufacturing environment. Section 2.6 presents some systems in more detail that form a basis for later comparison. But apart from correspondence and appearance-based methods there is one more major class of methods that have been used in the context of object localization. *Sampling-based* methods, such as the particle filters employed by Isard & Blake [IB98a, IB98b] or kernel particle filters [CA03, SKF06], generate a discrete sample set representation of a continuous posterior probability density, in short "posterior". The posterior, or rather its sample set approximation, captures how much evidence for hypothetical object poses arises from given image measurements. But unlike correspondence-based methods traversing the space of feature mappings, sampling-based object localization operates in the pose space. Each sample represents a hypothetical pose of the object under consideration, similar to hypotheses employed by the already mentioned hypothesize-and-test methods. Additionally, sampling-based methods associate a weight with each sample that rates how strongly the respective pose agrees with available image measurements.

In contrast to hypothesize-and-test methods, sampling-based methods proceed by iteratively resampling whole sample sets. In this way, all samples contribute to the solution of the localization problem and not just the verified ones. Another benefit of sampling-based techniques is that they avoid solving the correspondence problem. On the other hand, the pose space of articulated objects is high-dimensional, i.e. many samples might be needed to obtain a discrete approximation to the posterior. In order to keep their number in a computationally tracktable range it is necessary to represent "important" regions, only. This is feasible because the posterior density is usually very low for vast parts of the pose space. A suitable strategy is thus to concentrate on representing the modes of the posterior. A detailed illustration of a sampling-based object localization scheme and the extensions that have been developed in the context of this thesis in order to maintain

a compact representation of the posterior is provided in Chap. 4.2. To our knowledge, no one has so far tried to solve the assembly localization problem by using an equivalent approach.

2.5 Classification

Given that an inspection system has somehow acquired the position and orientation of an object within an image, the inspection scenario that was presented in the introduction chapter gives rise to a number of further questions that must be answered by the system. These new questions concern quality aspects of the localized object, e.g. in the sense of ”does the observed object have exactly the expected parts?” or ”are the object parts all positioned and oriented according to the design plans?”. The former question asks for *part completeness* while the latter is concerned with *pose integrity*. Both quality aspects can be analyzed by means of performing classification techniques.

In the following, *classification* is understood in a pattern recognition sense, i.e. as the task of assigning a label to a given input feature or to a set of input features. The label is a symbolic description of the class to which the input is mapped. In the case of part completeness classification, input features might be a mixture of the image features presented in Chap. 2.1. The corresponding labels qualify the input features as ”part missing”, ”part present”, or as ”rejected”, if no reliable decision can be taken. In the case of pose integrity classification, the input features are recovered assembly poses and the output labels either ”valid pose configuration”, ”fault configuration”, or ”rejected”.

The pattern recognition literature offers a vast amount of different techniques to solve classification tasks. Rather than attempting an exhaustive survey, only three wide-spread techniques are sketched in the following. Their presentation aims at introducing some fundamental concepts that are relevant for the discussion in Chap. 4.3. A thorough formal introduction to classification in the context of pattern recognition is provided by [Nie83].

The *nearest neighbor* (NN) classifier [CH67, DK80] is a frequently used classification technique. It proceeds by first storing all features of a labeled training set as prototype features. Input features are then classified according to their distance to the stored prototypes within the feature space. Note that this simple procedure can be extended to allow feature rejection, if the m nearest neighbors of an unclassified feature are taken into account. The great advantage of the NN classifier is its simplicity. A major drawback is the induced computational load, because all prototype features must be searched during each classification. Furthermore, the performance of a NN classifier depends strongly on the choice of the distance function that is used to determine the nearest neighbors. Finding a suitable distance can be very difficult.

Another group of frequently employed classifiers is known in the literature under the term *decision trees*. A decision tree is a tree whose nodes contain tests that are performed on

the input features. The leaf nodes of the tree contain the class labels. Classification proceeds by filtering input features through the tests of the tree nodes, starting at the root node. A number of algorithms have been proposed to automatically learn decision trees from labeled test sets [Qui93, BFOS84]. The learning procedures automatically determine suitable tests and an appropriate test order, usually by minimizing some uncertainty criterion. The major advantage of decision trees is that they are very fast. Compared to the many distance calculations that are usually made in the course of NN classification, only a very small number of tests are performed with decision trees. The tree training is computationally heavy but can proceed offline.

A third well-known classifier is the *Bayes classifier* [Ris01]. It operates on a set of input features, the prior probabilities of each output class (class priors) and the probabilities of the input features conditioned on the class (conditional feature models). Given appropriate class priors and conditional feature models, the Bayes classifier determines the output class that is associated with the highest posterior probability. The class priors and conditional feature models can be learned from labeled training sets. An appealing property of the Bayes classifier is that its decisions are optimal w.r.t. minimizing the probability of misclassification. Furthermore, its online performance is quite fast. Above all, the Bayes classifier provides means of combining different types of input features within one classification. Because each feature is associated with an individual conditional feature model, the Bayes classifier can be interpreted as transforming each feature value to a global probabilistic space in which the classification is carried out. This principle has e.g. been exploited in [Köl02]. Compared to the other two classifiers mentioned above, it is important to note that the Bayes classifier performs no rejection.

2.6 State-of-the-Art Inspection Systems

The previous sections have provided an overview of related work on all major aspects of computer vision systems for visual inspection of assemblies. What remains to be established is an evaluation of today's inspection systems performance. Ideally this would yield a set of performance indicators against which the system proposed in this thesis could be compared. Such indicators should e.g. document the accuracy and precision of the object localization process. Unfortunately, this kind of performance information is hard to obtain, though a few systems have been designed that could be used as reference for comparison. For instance, Khawaja and colleagues [KMTB94] report an inspection system for nearly rigid assemblies but fail to document any numbers on the localization accuracy and precision at all.

A thorough survey of the computer vision literature yielded five systems which perform pose estimation under comparable conditions and whose performance has been documented to some degree. Together they form the baseline for the performance evaluation

of the system proposed by this thesis. All five systems are concerned with localizing object poses, and two of them perform part completeness or pose integrity classification. All five system are research prototypes as, to our knowledge, none of the commercially available image processing tools provides an out-of-the-box solution to the pose estimation of arbitrarily articulated objects. They are presented in more detail in the following. Afterwards, the systems' measurement characteristics for pose estimation are summarized in Tab. 2.1.

Kölzow [Köl02] has proposed a model-based system for the localization and classification of rigid objects from sequences of monocular images that has already been mentioned before. In his work, classification refers to the decision whether a rigid object is present in an image sequence. The strength of the system lies in its robustness. The author presents a unique strategy that fuses information on the quality of local feature matches across different feature types to a global match value. Such values can be used to reliably refine multiple pose hypotheses in a correspondence-based hypothesize-and-test framework. Notably his system generates object models according to the approach in [Han01]. The latter has been used in our system, too, but only for the generation of part models. With regard to the classification process, the author has not documented performance indicators but of all systems presented here, he conducted the most thorough evaluation of the measurement performance.

The research system TINA, as reported by Lacey and colleagues [LTCP02], was originally developed within a pick-and-place scenario in robotics where a robot arm is guided by a vision system in order to manipulate objects. The system locates objects from stereo image data by performing 3D model matching: After edge detection, edge point correspondences are built for the stereo images and 3D edge points inferred. These are linked to straight lines and circular arcs and matched to a wireframe model by using a local feature-focus technique. The strength of the system lies in the fact that today it is a rich framework for vision research as many modules have been contributed to it over time. Its main drawback is that it only operates on rigid models.

Hel-Or & Werman [HOW96] have presented an approach for the localization of articulated objects stereo images. They employ a unique Extended Kalman Filter that estimates the pose of objects to conform with measurements while satisfying constraints modeled as a set of constraint equations. Unfortunately, the constraint fusion mechanism inherently depends on establishing correct model to image feature correspondences so that the algorithm still suffers from the problems of correspondence-based approaches, though in somewhat alleviated form. Consequently, the authors state that the success of the method relies on a manually defined approximate pose initialization. Furthermore, they localize a three-part desklamp whose joints exhibit only 5 DOF.

Bank et al. [vBGW03] presented a system for appearance-based industrial quality inspection from single monocular images. In an offline stage, the system extracts a large set of 2D edge templates from 3D CAD models. The template set captures the appearance of rigid objects from a dense discretization of a view hemisphere. It is organized

in a template hierarchy in order to improve the efficiency of the subsequent online pose estimation process. The latter employs an hierarchical edge matching scheme based on the chamfer distance transform. The pose is then classified into either correct or fault configurations. The strength of the system lies in its robust performance and its processing speed which is close to real-time. Furthermore, no approximate pose initialization is needed. On the other hand side, only rigid objects exhibiting 5 DOF are inspected. The system is able to classify the placement of ignition plugs with a rate of 98.4% correctly recognized faults at a false positive rate of 0%.

Socher [Soc97] described a vision component for the integrated speech and image understanding system QUASI-ACE. The system was developed in the context of the joint research project "Situated Artificial Communicators" within the Collaborative Research Center 360 at Bielefeld University. Socher's component recovers a 3D scene reconstruction based on stereo images and geometrical models of the contained rigid objects. Exemplary objects are wooden building blocks from the toy construction kit `baufix`® which are used within the experimental investigations of this thesis, too. The reconstruction process proceeds by hypothesizing objects from color blobs and edge features and subsequently refining the object poses in a hypothesize-and-test manner. The pose refinement is carried out as an iterative minimization of a cost function measuring the difference between projected model features and measured image features. The system is designed to deal with rigid objects, only. The measurement performance of her system is illustrated in Tab. 2.1, together with the characteristics of the other four systems presented here.

Table 2.1: Measurement characteristics of 5 computer vision systems for pose estimation. A '-' denotes that the respective value is unknown

System	N_{data}	N_{obj}	N_{parts}	$\frac{DOF}{object}$	$\frac{\text{mm}}{\text{pixel}}$	$\|\mu_{trl}\|$	σ_{trl}	$\|\mu_{rot}\|$	σ_{rot}
[Köl02]	420	1	1	6	≈ 0.7	$\leq 1\text{mm}$	$\leq 1\text{mm}$	$\leq 1^\circ$	$\leq 2^\circ$
[LTCP02]	-	1	1	6	-	$\leq 5\text{mm}$	-	-	-
[HOW96]	-	1	3	5	≈ 1.5	-	-	$\leq 1^\circ$	$\leq 1^\circ$
[vBGW03]	-	1-2	1	5	0.2	$\leq 1\text{mm}$	-	$\leq 1^\circ$	-
[Soc97]	27	2	1	6	≈ 0.4	$\leq 4\text{mm}$	-	$\leq 5^\circ$	-

The first six columns of Tab. 2.1 document important aspects of the pose localization processes. More precisely, N_{data} denotes the total number of measurements performed. Unfortunately, only Kölzow and Socher reported their test set size[3]. Each measurement

[3]Kölzow [Köl02] localized a variety of objects with measurements in the order of 10^4 images. The numbers stated here reflect a subset showing an oil cap attached to a car engine. The other reported computer vision systems localized objects of similar size. In contrast to this, the remaining part of Kölzow's test set contained objects like vehicles or airplanes whose scale is so large that the measurement process would not have been comparable at all. We therefore concentrated on analyzing the oil cap subset.

localized N_{obj} separate objects consisting of N_{parts} parts. It can be seen that the systems had to determine 6 DOF at maximum when localizing an object, even though in the case of Hel-Or & Werman objects consisted of three parts. The 6th column of Tab. 2.1 lists the scale of the processed image data, or estimates thereof that were obtained from published images together with known object dimensions.

The final four columns of Tab. 2.1 indicate the performance of translation and rotation parameter estimation. Regarding translation parameters, any values on how well the systems estimated object distances to the camera were ignored in order not to favor those estimating only 5 DOF, as the systems that determined only 5 DOF knew the camera distance in advance. The measurement accuracy is stated in terms of the mean absolute error μ_{trl} and μ_{rot} with which the pose estimates deviated from the ground truth. The measurement precision is stated in terms of standard deviations σ_{trl} and σ_{rot}. Unfortunately, four out of the five systems failed to document precision values. Furthermore it must be noted that the way in which the ground truth values were recorded differs significantly from system to system. Hel-Or & Werman and Socher used manually measured recordings, Kölzow relied on a manually established fit via PovRay, Bank et al. employed a calibrated robot arm and Lacey et al. don't discuss their methodology. As a consequence, all listed accuracy and precision values are rounded to the next integer precision in order to avoid an over-interpretation of the available data.

The information provided in Tab. 2.1 does not suffice to make out "the best" system. However, it does give a clear picture of today's computer vision systems that localize objects: When determining object poses with up to 6 DOF within images of approximately $1\frac{\mathrm{mm}}{\mathrm{pixel}}$ pixel resolution, a measurement accuracy and precision in the order of 1mm and 1° can be considered state-of-the-art. The visual inspection system that is presented in the following chapters will be compared against this finding.

3 Model Preparation

The previous chapter presented a detailed overview of the literature on all major problems and tasks related to visual assembly inspection. We have learned that the respective tasks can be categorized into offline and online tasks, according to the question whether they can be precomputed or depend on the latest image measurements. We have learned, too, that model feature extraction and assembly modeling are typical offline tasks. Together, they were subsumed under the term *model preparation*. This chapter reports how model preparation is accomplished in the system proposed by this thesis. It initially presents a system overview in order to illustrate the driving forces that result in requirements on the system's model representation. The subsequent sections elaborate how part models are extracted from CAD data, optimized with regard to memory consumption and access speed and finally aggregated to assembly models.

3.1 System Overview

Manufacturing processes depend on thorough planning, accurate design, and continuous optimization. Neither can be achieved without the strict application of engineering techniques which have considerably evolved with the advent of *CAx* techniques like computer-aided design (CAD), computer-aided manufacturing (CAM), computer-aided engineering (CAE), and computer-aided process planning (CAPP). Consequently, Hoffmann et al. pointed out that computer vision systems should evolve likewise: "Ideally the entire manufacturing process should be information driven, so that all functions necessary to support machine vision are performed solely from computer-aided design (CAD) models." [HKT89, p.1477]. The authors also address the problem that other information sources for vision systems like training examples are quite often too expensive to be useful in a manufacturing context. As a consequence, the system proposed here employs CAD data as its main source of information from the manufacturing environment. In this, it follows the same paradigm of CAD-driven vision systems that was used by all the systems presented in Sect. 2.6, too. Our system additionally relies on some external scene knowledge, e.g. a camera model and a reference to the assembly model that represents the object under inspection. Given a new image measurement, our system localizes the assembly model with respect to the camera coordinate system.

3.1.1 System Modularization

A functional system decomposition that refines the architecture overview of Fig. 1.2 is illustrated in Fig. 3.1. We can see that it has a pipe-and-filter architecture, with CAD data at its beginning and pose data at its end. The filter chain consists of two main parts, namely offline model preparation and online inspection. The initialization isn't counted as a system part but illustrated separate from the online inspection because it doesn't need to be repeated with each new image measurement.

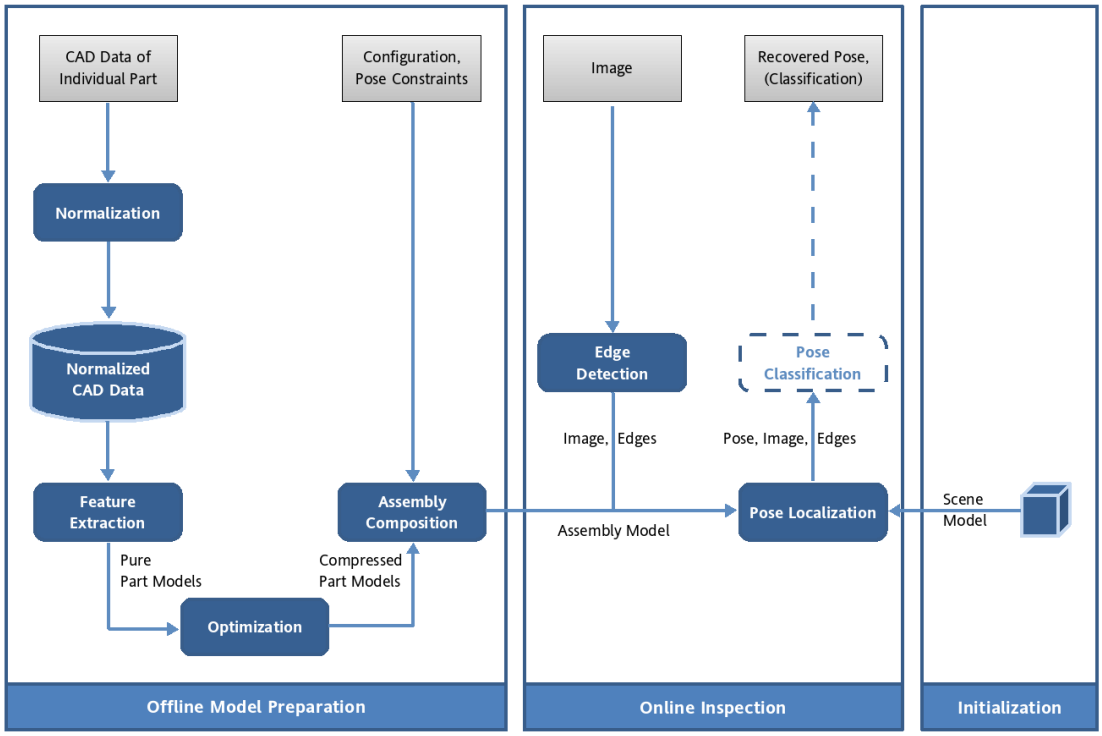


Figure 3.1: Refined architectural view of the proposed inspection system

Figure 3.1 shows that *offline model preparation* proceeds by first normalizing the CAD data for individual assembly parts. Normalization has two steps. First, the data is converted from one of the many existing formats to one appropriate for the subsequent algorithms. Second, the CAD information is transformed in order to fix topological irregularities resulting from the CAD processing tool chain like the ones already mentioned in Sect. 2.1. These steps can be performed by commercially available converter tools and so-called mesh healers and thus won't be described any further. The resulting normalized CAD data are sets of vertices, edges, and facets that together form the closed-surface representation of an assembly part. These part representations are processed by the feature extraction stage in order to automatically extract sets of characteristic model features,

yielding pure part models. The latter are optimized with respect to storage consumption. Afterwards, the compressed pure part models are composed to a constrained model which represents a validly configured assembly and physically possible (but not always valid) configuration deviations.

Online visual inspection, as illustrated in Fig. 3.1, must be initialized with a scene model. It specifies the overall inspection task, e.g. by providing the camera model parameters and a reference to the assembly model that represents the currently inspected object. Afterwards, new image measurements are preprocessed by an edge detection stage. Note that the system employs a simple pinhole camera model for 3D-2D central projection. A detailed discussion of this well known camera model and important variations is given in the book of Hartley & Zisserman [HZ03]. In order to compensate lens distortion, for which the pinhole camera model does not account, image measurements are assumed to be rectified[1]. The edge information and the original image are then both passed to the pose localization module. This employs a novel kernel particle filter (KPF) in order to determine the full assembly pose parameterization. Finally, a classification stage might receive the pose data, image, and edge information in order to classify individual part completeness and pose integrity. As the prototypical implementation of the proposed system currently doesn't have a classification module, the latter is visualized in a dashed frame in Fig. 3.1.

3.1.2 Assembly Model Requirements

From the system's overall task, its functional decomposition and interaction with a manufacturing environment, a set of requirements on the assembly model representation can be derived. The detailed presentation of the system's model feature extraction, optimization, and assembly model composition, which is carried out in the following sections, will address these requirements and discuss how far they can be met:

1. The visual inspection task places a high priority on measurement accuracy. The model features that are extracted by the offline model preparation stage must therefore facilitate accurate and robust object localization.

2. Optimizing or changing manufacturing processes can lead to changes in the produced parts and assemblies. Such changes must be propagated quickly to the inspection system. The offline model preparation stage must therefore be simple to use, at best fully automated.

[1] Image rectification and the corresponding camera calibration issues are not discussed within this thesis, hence the rectification process has been omitted in the system illustration.

3. The pose localization follows a sampling-based 2D-3D approach. As already indicated in Sect. 2.4, this approach samples a large number of pose hypotheses and compares them to the current image. This means that for each generated pose hypothesis, the object-centered 3D assembly model must be projected to the 2D image plane by employing the pinhole camera model, which produces a very high computational load. Above all, the constrained assembly model must thus be speed optimized for this kind of online transformation to a viewer-centered representation.

4. The central projection transformation must precisely account for view-dependend model feature occlusion. Inaccuracies will either yield false negative model edges, which should have been projected to the image plane but were considered hidden, or false positive model edges, which were considered visible but are really occluded. Both cases might degrade the accuracy of subsequent image measurement comparisons.

3.2 Automatic Model Feature Extraction

Hanheide [Han01] reported a powerful model feature extraction stage for model-based object localization systems which was already successfully used by Kölzow in [Köl02]. Its appeal lies in the fact that it automatically extracts local edge features from CAD models of single parts. Hanheide's and Kölzow's experimental investigations also showed that the extracted model features are well suited for accurate part localization. By employing his approach, we are therefore immediately able to meet the first two requirements stated above, at least for rigid parts. For the system proposed in this thesis, Hanheide's approach is combined with the visibility map concept reported in [EKSH05]. We will see later in this section that this combination is important for meeting requirements 3 and 4 with respect to single rigid parts. The resulting automatic model feature extraction system is presented in the following.

The model feature extraction stage processes normalized CAD models M_{cad}, each of which represents an individual rigid part as a 3-tupel

$$M_{\mathrm{cad}} = (V_{\mathrm{cad}}, E_{\mathrm{cad}}, S_{\mathrm{cad}})\,. \tag{3.1}$$

Here, V_{cad} denotes a set of 3D vertices and E_{cad} a set of connecting edges. Furthermore, S_{cad} specifies a set of polygonal surfaces (or facets) that are defined w.r.t. the edge set. The first step of the model feature extraction stage preprocesses M_{cad} by decomposing the arbitrary polygonal surfaces in S_{cad} to triangles, adding new edges to E_{cad}. All edges are then subdivided into minimal edges such that they always belong to exactly two triangles.

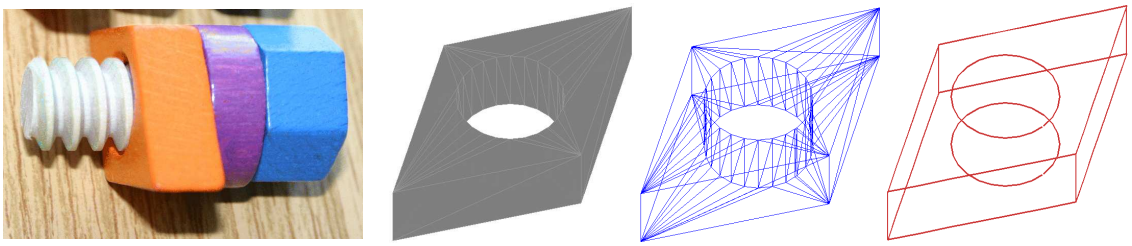


Figure 3.2: From left to right: A baufix® screw-ring-nut assembly, the nut's normalized CAD model, preprocessed edges E_{pre}, and the final contour edges E_{contour}

Subdividing an edge might insert new vertices to V_{cad}. The resulting preprocessed model is

$$M_{\text{pre}} = (V_{\text{pre}}, E_{\text{pre}}, S_{\text{pre}}) \,. \tag{3.2}$$

Afterwards, the model feature extraction stage assigns constant scores to each edge in E_{pre} longer than 1% of the largest Euclidean distance between any two vertices in V_{pre}. Additional constant scores are given to those edges whose adjacent facets' normals form an inner angle larger than 1° and whose adjacent facets' normals are oriented away from each other which together determines convex edges. Finally, all edges with a score higher than some fixed threshold form a new edge set $E_{\text{contour}} \subseteq E_{\text{pre}}$. Figure 3.2 illustrates an example of a physical object, its normalized CAD model, its preprocessed edge set, and its contour edge set.

The accumulation of scores, which can also be interpreted as gathering evidence for characteristic model features, and the subsequent edge selection ensures that the edges in E_{contour} share two key properties. First, they are of significant length relative to the whole model extent. This is important because long edges on the corresponding physical part might result in strong intensity discontinuities within image measurements. Second, the model edges are convex. Under 3D-2D transformations such as image measurements, only convex edges can be part of an object's contour or silhouette, i.e. the outline which separates the object from the image background. As the silhouette usually appears in real world images as a very strong intensity gradient and is also robust against illumination changes, the *contour edges* E_{contour} form a set of highly characteristic local features, meeting the first of the requirements specified in the previous section. If the facet information of the CAD model is attributed with color descriptors, the color information can be extracted as an additional local feature.

With respect to the criteria of [Pop94, p.4] that we presented in Sect. 2.2, Hanheide's work showed that the contour edges E_{contour} are sufficiently sensitive, unique, stable, and of appropriate scope. However, concerning the access efficiency, there is still much room for improvement because the edge set hasn't been optimized for 3D-2D central projections, yet. Basically, the problem is that for any central projection of the model

onto an image plane, most edges in E_{contour} are either occluded from view by other parts of the model or just don't belong to the actual contour[2]. As long as there is only one rigid part to be observed in a scene, both cases depend entirely on the view direction under which a model is perceived. Hence, the visibility of E_{contour} elements can be precomputed for the representation of a single rigid part.

In Hanheide's work, edge visibility is modeled either by a simple dot-product rule similar to [CL02] or by a union of hemispheres that describe from which viewing directions an edge can be seen. The latter rule is comparable to the work of [Low87]. However, both approaches model occlusion very inaccurately. What is more, convex edges that are facing the viewer but aren't part of the projected object's silhouette aren't modeled at all. Both issues are addressed much more satisfactorily by the *visibility map* concept of Ellenrieder et al. [EKSH05] which we used instead. A visibility map f_{vis} represents the visibility of a point on an arbitrary facet model. An example is shown in Fig. 3.3. It is a Boolean matrix that encodes whether the point can be seen under given azimuth and elevation angles. For our system, an azimuth and elevation step width of 1° was considered small enough to meet requirement 4. The overall generation scheme has already been detailed in Sect. 2.1. Given surfaces S_{pre} consisting of n triangles, a visibility map can be constructed with a computational effort of $\mathcal{O}(n)$. The encoded information can later be looked up in constant time, which exquisitely meets requirement 3 when performing the 3D-2D central projection of a single rigid part.

As already indicated, visibility maps model the status of a specific point. Fortunately, Ellenrieder et al. show that this concept is also suited for model features with a spatial extent in 2D or 3D. Our contour edges E_{contour} can for example be associated with visibility maps by first choosing one reference point r_i per contour edge $e_i \in E_{\text{contour}}$. Afterwards, for each of the reference points r_i a visibility map $f_{i,\text{vis}}$ is determined. Given that the camera distance to the reference points is many times larger than the largest feature diameter $\max_i \{\|e_i\|\}$, the error introduced by this operation is smaller than the inaccuracy arising from the discretization of the unit sphere.

For the tasks in the subsequent chapters and sections, the facets S_{pre} are of no more importance and are thus deleted from a rigid part's model. The vertex information in V_{pre} is kept and will be used in the following section. Furthermore, each of the N contour edges from E_{contour} is grouped together with its reference point and visibility map, resulting in a model feature set F_{contour}. Thus, the representation of a rigid part, resulting from the automatic model feature extraction stage presented so far, is a model of the form

$$M_{\text{part}} = (V_{\text{pre}}, F_{\text{contour}}), \text{where} \tag{3.3}$$

$$F_{\text{contour}} = \{(e_i, r_i, f_{i,\text{vis}})\}_{i=1}^{N}. \tag{3.4}$$

[2] In [SHS+04], experimental investigations with `baufix`® part models under random central projections showed that on average only 12% of the contour edges were forming the model's contour.

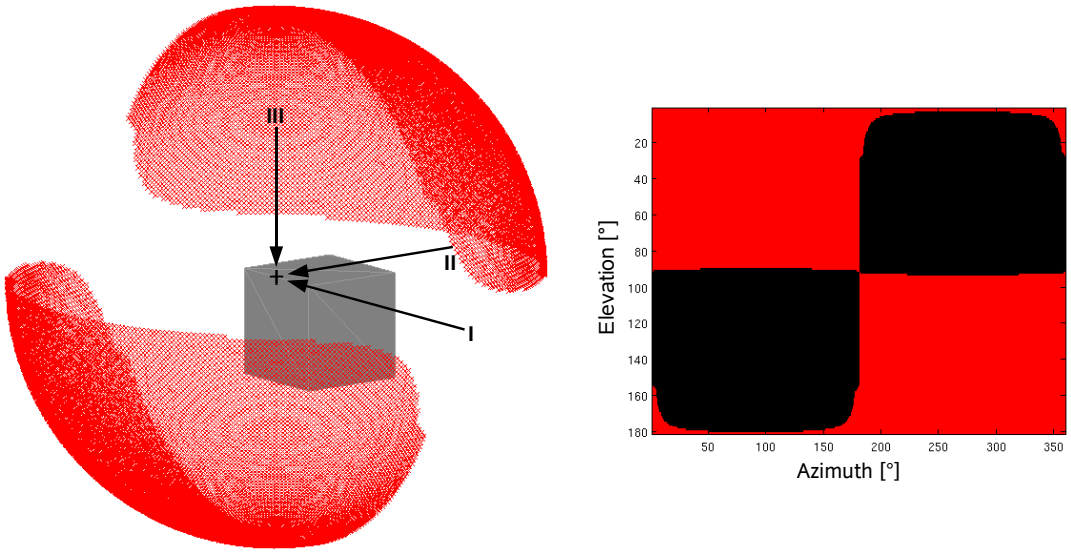


Figure 3.3: Cube with model feature point (black), around which a sphere is highlighted (red) for all positions, from which the feature point is part of the cube's contour. The positions are marked in red, too, in the corresponding visibility map on the right. The arrows indicate (azimuth, elevation) pairs of $(0°, 0°)$ at I, $(90°, 0°)$ at II, and $(\cdot, 90°)$ at III

In summary, this section has presented us a very versatile model feature extraction scheme. It can automatically extract sets of contour edges from CAD descriptions of rigid assembly parts. The contour edge visibility is determined accurately and automatically, too. Furthermore, the visibility information is stored in a compact representation that supports fast look-up operations, e.g. in the context of 3D-2D central projections. However, the approach presented so far only models single rigid parts. Therefore, this thesis contributes proposals how to extend the visibility concept to assemblies, how to optimize part models, and how to aggregate part models to assembly models. Taken together, these contributions yield a new, unique approach to create powerful constrained assembly models.

3.3 Model Feature Set Optimization

Along with the model feature extraction stage employed by the system proposed in this thesis, the previous section presented us some considerations on feature occlusion regarding a single rigid part. This section will first extend the considerations to assemblies composed from arbitrary many parts. Afterwards, we will see that the chosen occlusion determination approach can be storage and speed optimized by refining the sets of contour edge features from the previous section. This model feature set optimization is

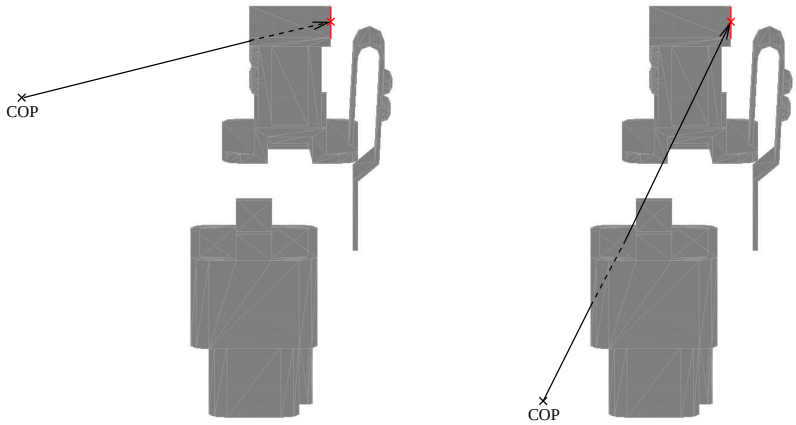


Figure 3.4: Model of an ignition plug and connector (courtesy of DaimlerChrysler AG). A contour edge's reference point (red cross) is hidden due to intra-part occlusion (left) and inter-part occlusion (right). COP denotes a camera's center of projection

carried out by minimizing a cost functions that either evaluate visibility map information or geometrical properties of contour edges.

3.3.1 Extending the Visibility Map Concept

Visibility maps model how reference points on the surface of a rigid part can become hidden behind regions of the same part, i.e. they represent *self-occlusion* which can also be termed *intra-part* occlusion, as it describes the effects occurring within a rigid part. If more than one part is present in a scene, however, reference points on one part can also become hidden from view because they are occluded by regions of another part in the scene. The latter phenomenon can be termed *inter-part* occlusion, as it refers to the effects between different objects. Performing 3D-2D central projection on the contour edges of multi-part assembly models requires the representation of both phenomena, intra-part as well as inter-part occlusion, if feature visibility is to be modeled accurately. Figure 3.4 illustrates the two occlusion types.

One might ask if the visibility map concept can be modified to account for inter-part occlusion. Unfortunately, a straight-forward extension is intractable in terms of memory consumption. The reason for this is that an extended visibility map would have to record reference point occlusion for each discrete view direction *and* each discrete assembly pose parameterization. However, the number of discretizations of the assembly pose space grows exponentially in the number of assembly parts. Thus, for assemblies with more than two or three parts, the discretization would have to be extremely coarse-grained, if a tractable number of assembly poses is to be analyzed. This in turn would soon yield an unacceptably large approximation error. In summary, we can't precalculate

the *entire* visibility information. The key to success lies in finding an information subset for which this is possible.

Obviously, inter-part occlusion can only occur for model features that would otherwise have been visible in a certain scene, i.e. inter-part occlusion can only affect model features that aren't already self-occluded. Therefore, the visibility information of model features on multi-part assemblies can be determined with a two-stage process. First, given the pose of an assembly relative to a camera or an observer, one can determine all potentially visible model features that aren't affected by intra-part occlusion. Afterwards, the potentially visible model features can be narrowed down to fully visible ones by evaluating inter-part occlusion. As we already described a representation that yields fast determination of self-occlusion, the question remains how we intend to perform the second step.

This thesis models inter-part occlusion with the help of *oriented bounding boxes* (OBBs). According to Arvo & Kirk [AK89], OBBs have been used in the past to speed up ray-tracing scenarios where viewrays must be tested for intersection with objects in a given scene. A bounding box is a cuboid which is tightly fitted around a set of 3D points, in our case the vertices in V_{pre} which provide a sufficiently dense representation of the original part shape. An oriented bounding box always exhibits an orientation in 3D space which yields a tight fit to the point cloud V_{pre} while minimizing the enclosed volume. On the contrary, *axis aligned bounding boxes* (AABBs) are always oriented parallel to the axes of the world coordinate system. Discussing the details of OBB generation lies outside the scope of this thesis, though, as this is a too common technique. It must suffice to note that OBBs can be generated automatically from the vertex data of CAD models, e.g. by using the approach of Gottschalk et al. [GLM96]. An outline of this approach is given in appendix A.

Given that our system can automatically create a set of OBBs for each part of an assembly model, the two-stage determination of contour edge visibility proceeds as illustrated in Fig. 3.5. We assume that a scene is given containing a camera and an assembly whose part locations are known in camera coordinates. The first step then retrieves all potentially visible contour edges by querying all visibility maps. The potentially visible contour edges are called *active*. The respective *active model feature set* of an assembly part with index j is denoted F^j_{active} in the following. Obviously $F^j_{\mathrm{active}} \subset F^j_{\mathrm{contour}}$, where F^j_{contour} specifies the model feature set of assembly part j as defined in Eqn. (3.4). In the second step, for each part j, view rays between the camera's center of projection (COP) and each reference point in F^j_{active} are intersected with the OBBs of all assembly parts $k \neq j$. Only active edge segments whose view ray doesn't intersect any OBB are perceived as visible by the camera. They can now be projected to the camera's image plane by performing central projection. Some real contour edge determination results are shown in Fig. 3.6.

Requirement 3 from Sect. 3.1 has already specified that determining the visibility of model features is a mission critical part of our application. Now that we have defined a

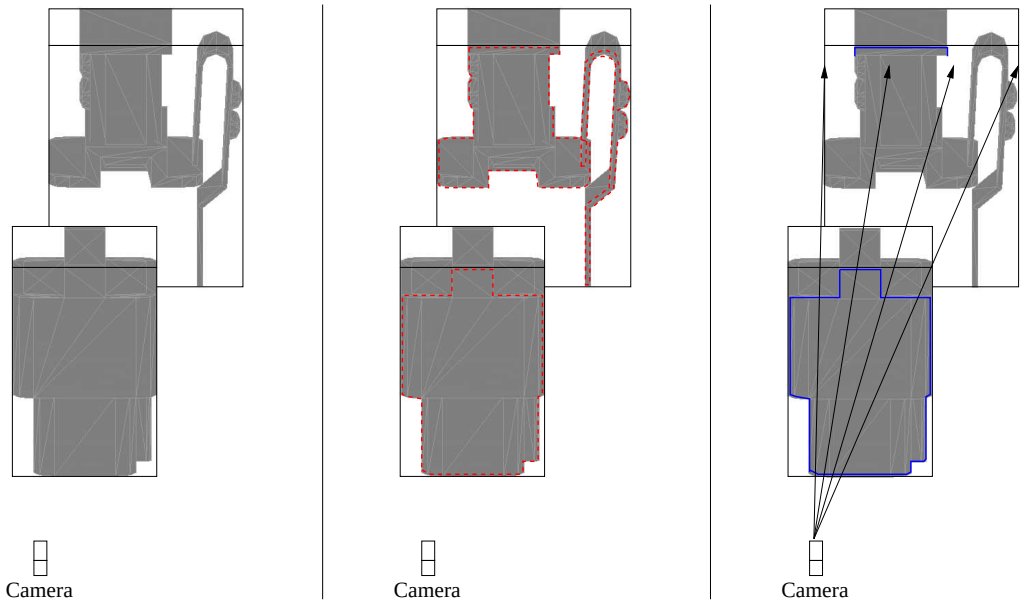


Figure 3.5: Two-stage occlusion determination. *Left:* Ignition plug assembly model (courtesy of DaimlerChrysler AG) with tightly fitted OBBs. *Middle:* Active contour edges yielded by the 1st stage (red dashed lines). *Right:* Visible edges resulting from the 2nd stage (blue lines). The arrows denote topmost OBB-intersecting view rays

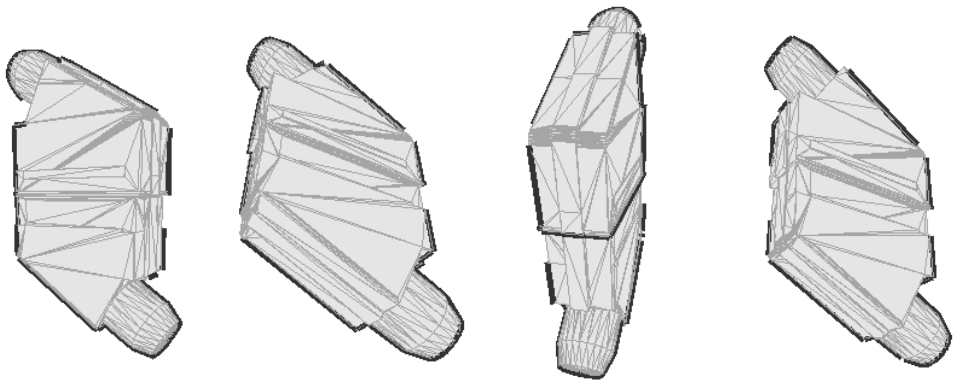


Figure 3.6: A scart plug and connector, perceived from varying directions. The black lines are contour edges that have been determined as visible. The gaps in the contour occur because of bounding box intersections, modeling inaccuracies, and due to the fact that a number of edges have been eliminated in the preprocessing stage because they were too short

corresponding process, the question remains what its computational complexity is. Let N_{features} denote the total number of contour edge features in all F^j_{contour}. Furthermore, let N_{active} be the total number of elements in all F^j_{active} and N_{OBBs} the total number of OBBs fitted around assembly parts. Given this, performing self-occlusion checks has a worst-case time complexity of $\mathcal{O}(N_{\text{features}})$, as each of the N_{features} contour edge features is tested for self-occlusion in constant time. Additionally, all active model features must be tested against the bounding boxes, which is performed in $\mathcal{O}(N_{\text{active}} \cdot N_{\text{OBBs}})$ in the worst case. Consequently, the overall worst-case time complexity of the two-stage visibility determination process is $\mathcal{O}(N_{\text{features}} + N_{\text{active}} \cdot N_{\text{OBBs}})$.

The computational complexity of the visibility determination process is linear in all of its three variables. This result is already very satisfying but it might still be possible to decrease it in the order of some constant factor by minimizing the three contributing variables. Concerning the number of OBBs N_{OBBs}, such a minimization is easy to achieve. One simply sets the number of OBBs to the lowest value that permits intersection tests, i.e. one OBB per part. All experimental investigations reported in the evaluation chapter of this thesis have been carried out with this setting which shows that it still yields a reasonable contour edge visibility approximation.

In contrast to the OBBs, minimizing the total number of contour edge features N_{features} and the number of active model features N_{active} is a more complex task. It proceeds by considering each assembly part separately and erasing elements from the associated contour edge feature set F^j_{contour}. However, the elements to erase must be selected carefully because the remaining set has to facilitate online inspection. More specifically, the remaining model features must still be suited for pose localization as the system's classification performance will depend crucially on the pose localization accuracy. When minimizing contour edge features, we therefore try to eliminate those elements that contribute little to solving the pose localization task.

Such a choice depends on the pose localization approach itself. Correspondence-based approaches require at least three part model points that are neither co-linear nor co-planar to be mapped to an image before the spatial matching problem can be solved for that part. Line segments like contour edges often serve as robust point features by mapping their mid-point to the mid-point of image edges. Consequently, at least three contour edge elements whose mid-points aren't co-linear and co-planar must be visible per assembly part to recover that part's pose. More than three visible contour edges in general yield a more accurate part localization. Furthermore, matching model to image features proceeds more robustly, if pairs of contour edges meet in corner-like structures. In contrast to this, appearance-based and sampling-based pose estimation approaches operate on the full appearance of an object which is in our case represented as the object's contour. This means that predicted object contours must match well with the really detected ones. Deleting more and more contour edge features from a part model might give rise to changes like cracks or gaps in the predicted contours which might decrease the respective match quality. Consequently, edge feature minimization for such approaches must

reduce the total number of model features N_{features}, while leaving the number of potentially visible model features N_{active} almost unchanged. In the following, a minimization procedure is specified that can be tailored for all pose localization approaches mentioned above. The model feature set optimization contributed here might thus be suited for many more model-based vision systems than the inspection system detailed in this thesis.

3.3.2 Optimizing Sets of Part Model Features

Let M^j_{part} be the assembly part model whose contour edge feature set F^j_{contour} is currently processed in order to obtain an optimized model feature set F^{j*}_{contour}. Furthermore, let all the individual contour edge features $f^j_{i,contour} \in F^j_{\text{contour}}$ of the part model, analogous to Eq. (3.4), be of the form $f^j_{i,contour} = (e^j_i, r^j_i, f^j_{i,\text{vis}})$. Assume that for each of the contour edge features a score u^j_i can be determined that rates the model feature utility for the pose localization task. The problem of optimizing the set of contour features can then be specified as a one-dimensional knapsack problem with Boolean variables (see [MT90] for an in-depth discussion of this problem class). It has the form

$$F^{j*}_{\text{contour}} = \max \{ \sum_{i=1}^{N^j} u^j_i x^j_i | \sum_{i=1}^{N^j} x^j_i \leq b, b \in \mathbb{R}, b \geq 1, x^j_i \in \{0,1\}, i = 1, ..., N^j \}, \quad (3.5)$$

where the x^j_i are Boolean variables. They define which of the N^j contour edge features $f^j_{i,contour} \in F^j_{\text{contour}}$ belong to the optimized set. The optimization in Eqn. (3.5) maximizes the model feature utility score while keeping the total number of elements in the optimized set below a threshold b. We assume in the following that all elements in F^{j*}_{contour} are sorted such that the associated utility scores u^j_i comply with the *regularity condition*

$$u^j_1 \geq u^j_2 \geq \cdots \geq u^j_{N^j}. \quad (3.6)$$

The problem from Eqn. (3.5) is an important mathematical model for combinatorial optimization and generally known to be *NP*-hard [BDKS04]. Fortunately, there exist a number of algorithms for the fast computation of good approximate solutions which completely suffice for the task at hand. Among these, *greedy* algorithms were proposed as favorable in [BDKS04] because they have an execution time linear in the number of contour edge features N^j, if the regularity condition (3.6) holds. Accordingly, such an approach was chosen for the model feature set optimization proposed here.

The greedy algorithm employed in this thesis proceeds as follows. Initially, the infeasible solution $\mathbf{x^j} = (x^j_1, \cdots, x^j_{N^j}) = (1, \cdots, 1)$ is chosen that represents the completely unoptimized model feature set F^j_{contour}. Afterwards, starting with the right-most element $x^j_{N^j}$, the x^j_i are iteratively set to zero, until a feasible solution is found that satisfies the

optimization equation (3.5). Due to the regularity constraint, this procedure first erases the contour edge feature with the smallest utility score and then eliminates further ones in the order of increasing utility.

Optionally, F^{j*}_{contour} can be restricted to satisfy a number of additional constraints that can be obtained from visibility maps. Let $f^{j}_{i,\text{vis}}(a,e)$ denote the visibility of a contour edge feature that is perceived from a direction specified by the discrete azimuth a and elevation e. Then, each pair (a,e) for which a Booelan entry was recorded to $f^{j}_{i,\text{vis}}$ yields a constrain of the form

$$\sum_{i=1}^{N^j} f^{j}_{i,\text{vis}}(a,e) \geq N_{\min}, \tag{3.7}$$

i.e. for any discrete view sphere position, F^{j*}_{contour} must provide at least $N_{\min}$ active model features. Equation (3.7) is termed *visibility constraint* in the following. The greedy algorithm accounts for extra constraints by setting any element x^j_i only to zero, if this operation doesn't lead to a violation of the additional constraint set. If it would, the greedy algorithm skips this element and resumes with processing x^j_{i-1}.

Visibility constraints are important when optimizing model feature sets for correspondence-based pose localization. In this case they are necessary to provide sufficiently many contour edge features that can later contribute to solving the spatial matching problem. Given visibility constraints, threshold b from Eqn. (3.5) is best set to a value that is just high enough to permit a non-empty optimization solution. An appropriate choice of b can be found by initializing it to $N^j - 1$ and continually decreasing it as long as the greedy optimization procedure yields a non-empty set.

In the context of appearance-based or sampling-based pose localization, a good choice for b is much harder to obtain. We have already learned that these methods operate on the object appearance as a whole and that the size of a contour edge set can only be reduced as long as the represented contour information is preserved, at least to a large degree. Clearly, removing a contour edge feature $f^j_{i,contour}$ from F^j_{contour} can only induce a loss of contour information for those view directions, for which the feature would have been active. Consequently, if the condition $f^j_{i,contour} \notin F^j_{\text{active}}$ holds regardless of the chosen view direction, i.e. the model feature is never active at all, it can be removed safely. On the contrary, model features that are active from many view sphere positions might induce too much loss of contour information. Thus, before a suitable choice of b can be specified, it is necessary to obtain a measure for the amount of contour information conveyed by a model feature set F^j_{contour} and to decide how much loss is acceptable.

Given a model feature set F^j_{contour}, a statistical measure for its amount of contour information is given by the expected number of active contour edge features per view. It is denoted q^j in the following, or q^{j*} when referring to the respective optimized set. Given the visibility maps $f^j_{i,vis}$, the value of q^j is calculated by evaluating the visibility maps for each discrete view direction recorded in the maps and normalizing over the number of

view directions. Let $\mathbb{D}$ denominate the set of azimuth and elevation pairs (a, e) for which visibility map entries have been recorded, with $d = card(\mathbb{D})$. Then, as outlined above, q^j is given by

$$q^j = \frac{\sum_{i=1}^{N^j} \sum_{(a,e)\in\mathbb{D}} f^j_{i,\mathrm{vis}}(a,e)}{d}. \tag{3.8}$$

We can now specify, how many percent of q^j might be traded in for the sake of a model feature set reduction and write this acceptable loss as Δq^j. Thus, for a model feature set optimization in the context of appearance-based or sampling-based pose estimation, the original problem in Eqn. (3.5) can additionally be constrained by restricting feasible solutions to those sets $F^{j*}_{\mathrm{contour}}$ that comply with

$$q^{j*} \geq q^j(1 - \Delta q^j). \tag{3.9}$$

Given this extra constraint, a suitable value for b can again be found by initializing it to $N^j - 1$ and continually decreasing it as long as the greedy optimization procedure outlined above yields a non-empty set.

3.3.3 Feature Utility Scores

It has so far been assumed that a score u^j_i can be assigned to each contour edge feature of the set F^j_{contour}. The score's purpose is to represent the model feature utility with respect to a given pose localization task and might therefore vary for different tasks. It is specified in the following, which scores are employed in this thesis.

In the context of appearance-based and sampling-based pose estimation, the previous considerations have stressed the paramount importance of potential model feature visibility. The larger the view sphere area is, from which a contour edge feature is potentially visible, the more likely it is that the feature is classified as visible at the end of the 2-stage process that was outlined at the beginning of this section. In order to select those model features that are likely to be visible, u^j_i can be calculated from the visibility map $f^j_{i,vis}$ of the corresponding model feature $f^j_{i,contour}$ by determining

$$u^j_i = V(f^j_{i,contour}) = \frac{\text{visible area of } f^j_{i,vis}}{\text{total area of } f^j_{i,vis}} \approx \frac{\sum_{(a,e)\in\mathbb{D}} f^j_{i,\mathrm{vis}}(a,e)}{d}, \tag{3.10}$$

where $V(f^j_{i,contour})$ is the *visibility ratio* from [EKSH05]. Given a small acceptable loss Δq^j, it can now be established empirically how much contour edge information of an assembly part model can be optimized away. Figure 3.7 shows the results of optimizing 21 different assembly part models, with Δq^j varying from 0 to 6 percent. It can be seen that the size of contour edge feature sets is reduced by approx. 50 percent in the best

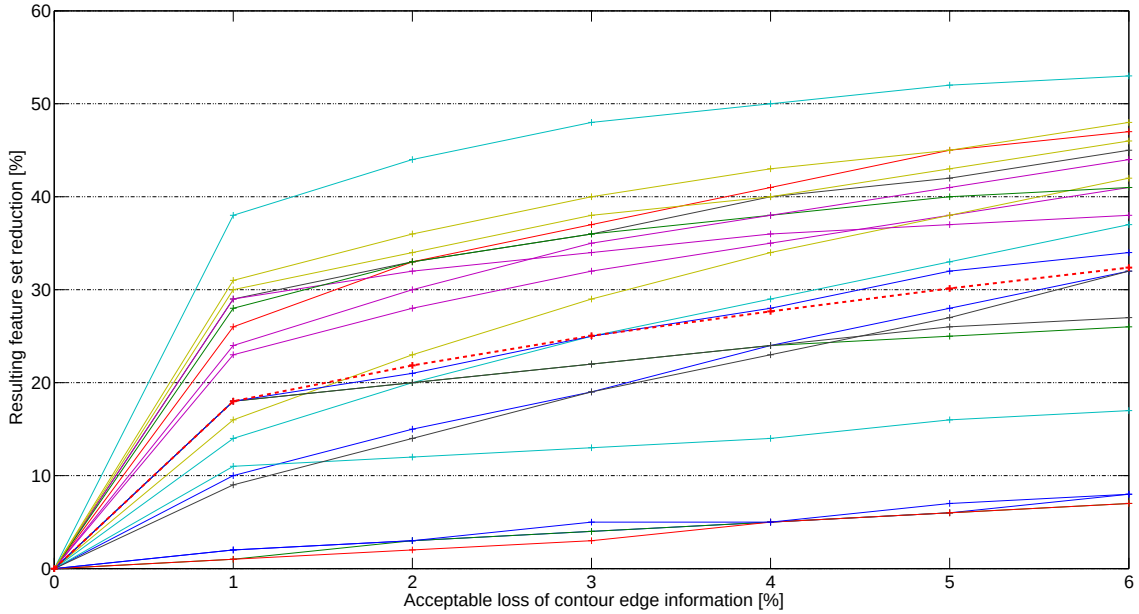


Figure 3.7: Optimization of 21 assembly part models with varying Δq^j. The dashed red line illustrates the mean of all measurements

cases. The fat dashed line, which illustrates the measurement mean, shows that more than 25% size reduction is achieved on average by performing the optimization scheme with $\Delta q^j \geq 3$.

In the context of correspondence-based approaches, model feature visibility is already sufficiently enforced by means of the visibility constraints. Therefore, the utility score doesn't need to account for this kind of information any more. Instead, groups of contour edge features should be selected that form locally salient structures like corners when projected to the image plane. Corners are of high utility to correspondence-based pose estimation because they can be matched more robustly to image features than straight edges. Accordingly, in the context of such pose estimation approaches, utility scores u_i^j should reflect the "cornerness" of pairs of contour edge features under projection to an image plane. In the following, the absolute sine of two meeting edges' inner angle is used as score. Therefore, perpendicular edges receive the highest possible weight, while co-linear edges are discounted.

The utility scores for an assembly part model M_{part}^j are calculated by traversing each view direction $(a, e) \in \mathbb{D}$ for which visibility map entries were recorded. For each view direction, the set of active contour edge features F_{active}^j is looked up and all of its elements are projected to the 2D image plane. Afterwards, all 2D edge segment pairs are determined that have endpoints meeting in an $\epsilon - \text{ball}$. For each such pair (e_l^j, e_k^j), the associated utility scores u_l^j and u_k^j are increased by $\text{abs}(\sin(\angle(e_l^j, e_k^j)))$.

Table 3.1 illustrates the model feature set reduction rates that were achieved when optimizing the 21 models from the previous experimental investigation. The setting was $N_{\min} = 15$. It can be seen that this setting yields a compression rate of up to 91% and achieves an average reduction of 54%. Thus, whenever a vision task needs part models that always provide a small number of characteristic visible model features, e.g. in order to establish correspondences, the automatically extracted contour edge feature sets can be dramatically reduced in size by employing the optimization stage contributed here.

Table 3.1: Optimization of 21 assembly part models with visibility constraints and $N_{\min} = 15$

Model	Feature set reduction
3-hole bar	42%
5-hole bar	57%
7-hole bar	67%
chinch connector, type a	73%
chinch connector, type b	68%
chinch plug, type a	73%
chinch plug, type b	58%
cube	74%
flat washer	48%
handle	91%
hexagonal nut	29%
ignition connector	32%
ignition plug	71%
industry norm screw	20%
nut	27%
oil cap	48%
wheel bearing	53%
ring	61%
scart connector	44%
scart plug	52%
toy screw	39%
average	54%

This section presented a two-stage process for the fast, reliable, and accurate determination of model feature visibility that operates on multi-part assembly models. When analyzing the time complexity of this process, it became clear that the system's online performance would benefit from reducing the number of assembly part features. In order to achieve such a reduction, this section contributed a new model feature optimization stage which reduces the number of contour edge features down to a minimal number. Two preliminary experimental investigations documented typical model feature set size reductions that can be achieved with this approach. It remains to be established, whether

the resulting contour edge feature sets are really suited for visual assembly inspection. Performance aspects such as this one are evaluated in Chap. 5.

3.4 Aggregating Rigid Parts to Assembly Models

The previous two sections have shown us how contour edge features can be automatically extracted from CAD models and how the model feature visibility determination can be extended from part to assembly models. It has further been detailed how part model feature sets can be optimized with respect to memory consumption. Taken together, this information gives a clear picture of the structure of part models. However, the structure of assembly models, i.e. part model aggregations, still remains to be covered to a similar degree. In order to do so, it is important to get an idea of the the application context, i.e. the vision system operations that must be supported by the assembly model representation. It can then be defined how assembly models are structured and explained how application context and assembly representation fit together. The following section addresses these issues, starting with a brief recapitulation of the information on assembly models presented so far and a description of the application context.

3.4.1 Application Context of Assembly Models

Chapter 2.2 already explained that the system proposed in this thesis employs an object-centered geometric assembly model representation and also motivated the choice of representation type. According to this categorization, each assembly part model is affixed with its own object coordinate system and conveys a high amount of geometric shape information. The latter is represented as sets of contour edge features[3]. We have also learned that assembly models are attributed with structural knowledge like descriptions of hierarchical part dependencies and possible motion between parts. Nevertheless, it is still unclear what these hierarchical part dependencies really are and how motion between parts is defined and constrained. And it is unclear in which respect these aspects are relevant for performing visual inspection tasks. Within this application context, it is mainly the assembly pose localization stage that imposes requirements on the assembly model representation. It has already been declared before that this thesis employs a new sampling-based pose estimation approach, namely a unique KPF. Details of this approach are presented in the following chapter. At this point, it is sufficient to declare that the KPF interfaces with assembly models mainly via three operations:

[3]It has been shortly mentioned before that color regions can be used as an additional kind of model feature. The current implementation of the assembly inspection system doesn't extract them automatically, though the proposed model feature extraction stage could be extended accordingly. Therefore, the considerations within this thesis are centered on contour edge features, as this is the only model feature type that was automatically extracted within the experimental investigations

1. `Transform`: Given an assembly pose, retrieve the position and orientation of each assembly part model with respect to the camera coordinate system.

2. `Query`: Determine the visibility of assembly part features with respect to a given assembly pose parameterization and camera.

3. `Sample`: Randomly draw an assembly pose parameterization from some distribution over the space of all physically possible assembly poses.

3.4.2 Assembly Pose Representation

All three interface operations somehow depend on an *assembly pose* or on an *assembly pose parameterization* which is used as a synonym. Hence, it should be explained how assembly poses are modeled within this thesis. Let $\mathcal{A}$ denote an assembly model that is configured according to an assembly pose $\mathbf{x}_{\mathcal{A}}$. Furthermore, the j individual part models of the assembly are again referred to as M^j_{part}. Because we decided to use an object-centered representation, each part model is affixed with its own local coordinate system. Consequently, assembly pose $\mathbf{x}_{\mathcal{A}}$ must specify the orientation and position of each part's coordinate system. This leads to a pose vector of the form

$$\mathbf{x}_{\mathcal{A}} = (\mathbf{x}^1_{\text{trl}}, \mathbf{x}^1_{\text{rot}}, \cdots, \mathbf{x}^j_{\text{trl}}, \mathbf{x}^j_{\text{rot}})^{\text{T}}, \tag{3.11}$$

where $\mathbf{x}^k_{\text{trl}}$ and $\mathbf{x}^k_{\text{rot}}$ with $k = 1, \ldots, j$ specify the translation and rotation of the j part models relative to some reference coordinate system. Note that the current system implementation represents rotations with roll, pitch, and yaw angles as defined in [Cra89, pp.45-48].

If the $\mathbf{x}^k_{\text{trl}}$ and $\mathbf{x}^k_{\text{rot}}$ would specify part translations and rotations relative to the same reference coordinate system, $\mathcal{A}$ would be a pure part assembly model. However, it has already been pointed out in Chap. 2.2 that models of this type fail to represent spatial dependencies between individual parts. Because spatial dependencies help in stabilizing the pose localization task, they are an important part of assembly models.

The description of spatial part dependencies is commonly divided into *static* and *dynamic* information. The static description relates pairs of assembly part models to each other. For each pair of models, one is declared as *attached part* and the other as *base part*. By definition, the former might move with respect to the latter. Hence, the coordinate system of an attached part is defined relative to the one of its base part. The dynamic description of spatial part dependencies then specifies the relative pose parameter values. In our case, the dynamic description is represented by the pose vector $\mathbf{x}_{\mathcal{A}}$. Accordingly, part translation $\mathbf{x}^k_{\text{trl}}$ and orientation $\mathbf{x}^k_{\text{rot}}$ with $k = 1, \ldots, j$ are interpreted with respect to the base part of the kth part model.

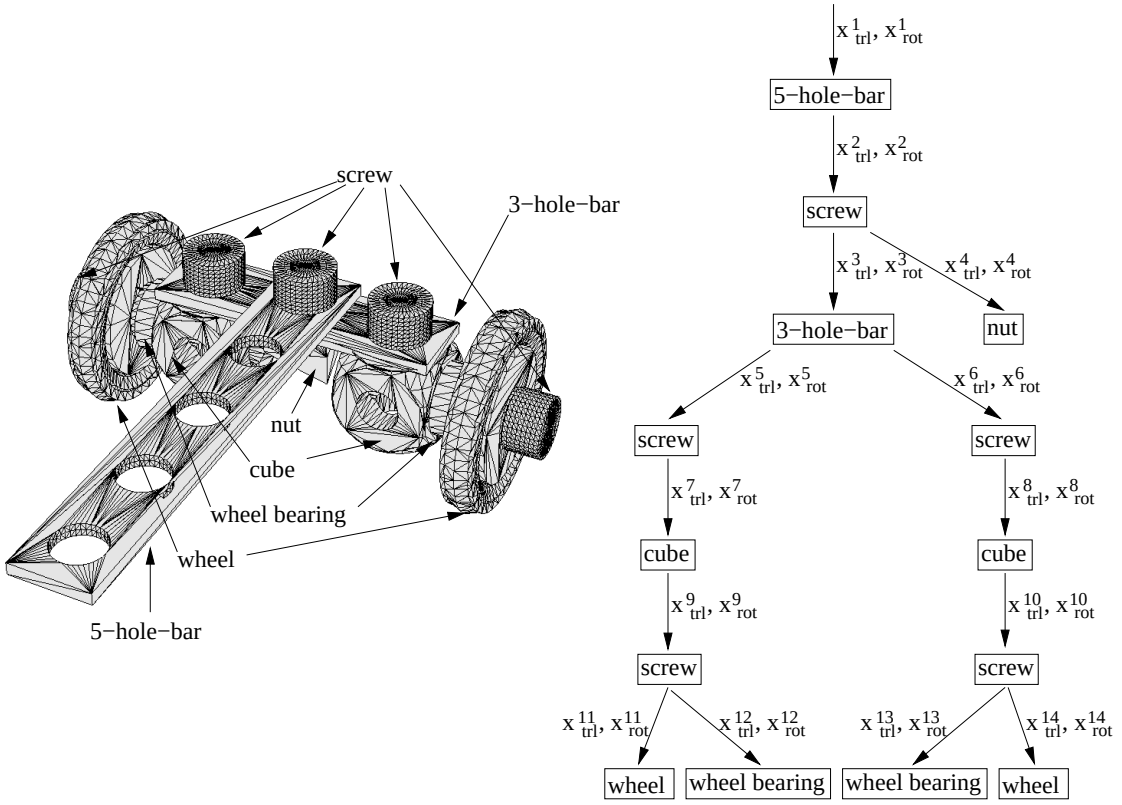


Figure 3.8: A `baufix`® assembly (left) and its kinematic tree (right)

Many CAD modeling packages represent the static part of spatial dependencies as a directed graph structure in which nodes denote part models and directed edges represent part relationships. Within such a dependency graph representation, any part model can be attached to multiple other parts and also be the base part relative to multiple attached elements. For our purposes, it is sufficient to restrict the graph representation to trees. This decision simplifies modeling spatial relationships as it restricts attached parts to have exactly one base part. Within an assembly model, the edges of a dependency tree are attributed with dynamic part information, resulting in a *kinematic tree* as introduced in Chap. 2.4. Thus, the pose $\mathbf{x}_{\mathcal{A}}$ of an assembly $\mathcal{A}$ is changed by modifying the edge attribute values of the assembly's kinematic tree, whereas the tree structure remains fixed.

Figure 3.8 illustrates an assembly composed from `baufix`® part models, together with a corresponding kinematic tree. The example shows that the tree root is also a part, alas without any base part. It is termed the *root node part* in the following. Its position and orientation is always interpreted with respect to the world coordinate system.

Note that the kinematic tree from Fig. 3.8 isn't unique. For example, there are many different ways of modeling "being attached to" relationships, so a screw might be modeled attached to a cube or vice versa and such decisions yield different dependency trees. One

could also choose any other part for the tree's root node. Fortunately, the pose localization stage doesn't make any assumptions concerning kinematic trees. It just depends on *some* spatial information being specified at all. So how is a kinematic tree like the presented example obtained? Principally, this could be done by exploiting the internal data of CAD modelers or by automatic conversion of *liaison graphs*. The latter were originally introduced in [Bou84]. They encode physical contact relationships and can be created automatically by analyzing surface contacts of CAD models or bounding box intersections. Nevertheless, the current system implementation relies on manual kinematic tree specifications, encoded as XML documents.

Returning to the application context, the above considerations also help to explain how the `Transform` operation interacts with assembly models. First, the coordinate system of the root node part is transformed to the camera coordinate system. This is easy because the transformation between the world and the camera is known from applying standard camera calibration techniques. Thus, one must simply concatenate the pose of the root node part, which is defined relative to the world coordinate system, to the calibration result. Afterwards, the `Transform` operation recursively traverses the kinematic tree. With each step along a tree edge, the relative pose data that is attributed to the edge is concatenated to the transformed pose of the respective base part. Once all model part coordinate systems have been transformed to the camera coordinate system, the `Query` operation can determine the assembly feature visibility by proceeding as presented in Chap. 3.3.1.

3.4.3 Constrained Assembly Models

`Transform` and `Query` operate on given assembly poses. In contrast to this, `Sample` is supposed to generate them. It has already been declared above that the task of this operation is to draw a random sample out of some distribution over the space of physically possible poses. It remains to be established, what the meaning of "physically possible" is, what the purpose of this concept is, and how such a space can be represented.

Concerning the definition of "physically possible", Chap. 2.2 referred to those assembly configurations that are specified in accordance with the laws of physics. For example, both assembly poses in Fig. 3.9 are physically possible, though only the right hand side instance is configured regularly. Impossible assembly poses would for example arise from parameter values in $\mathbf{x}_{\mathcal{A}}$ that would require some assembly parts to float freely in the air or vacate the same region in 3D space. In the following, $\mathcal{C}_{\mathcal{A}}$ denotes the space of physically possible poses for assembly $\mathcal{A}$ or an approximation thereof.

The concept of "physically possible" assembly poses is important in order to describe the search space of assembly pose estimation activities in the context of visual inspection. Most importantly, the space $\mathcal{C}_{\mathcal{A}}$ must subsume all regularly configured assemblies as well

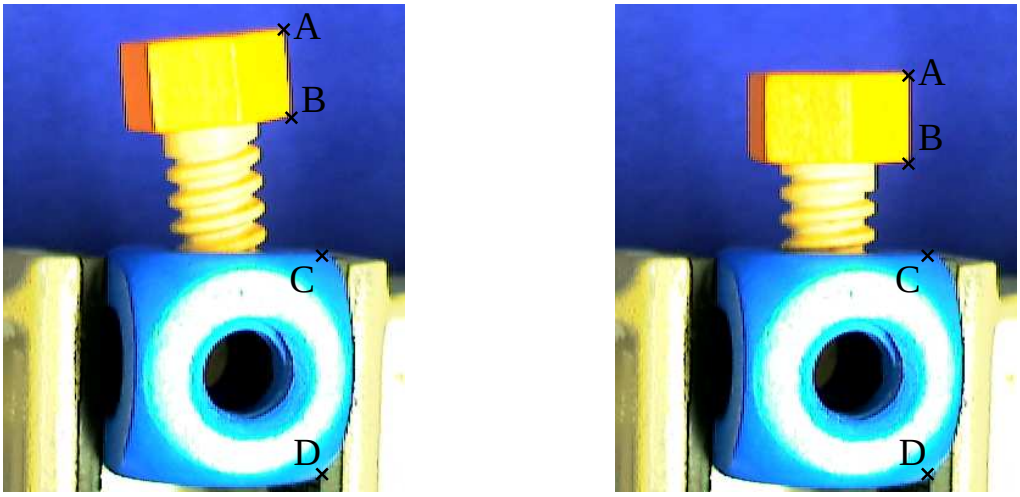


Figure 3.9: Screw-block assembly in a fault configuration (left) and with regularly mounted screw (right). A, B, C, and D indicate specific part point positions

as fault configurations. Some systems only consider assemblies of the former subspace. For example, the already presented system of Hel-Or & Werman [HOW96] estimates the pose of a desklamp based on a set of constraint equations that provide valuable information to the pose estimation task. We have learned in Chap. 2.2 that these constraints define spatial invariants like co-linearity, co-planarity, parallelity or fixed distance relationships of feature point groups. However, none of these constraints might hold if an assembly isn't built properly which is exactly the type of situation our system might encounter. This problem is illustrated in Fig. 3.9. The points A,B and C,D on the screw-block assembly could be subject to a parallel-constraint which would hold regardless of how far the screw is twisted into the block. However, in case of a fault configuration such as the illustrated one, the constraint would be violated and corresponding pose solutions rejected.

This thesis modifies the constrained model concept of Hel-Or & Werman such that it becomes applicable for the visual inspection scenario. The assembly models proposed here represent spatial constraints as additional kinematic tree attributes instead of constraint equations. This approach exploits the fact that kinematic trees already model hierarchies of parts that are being attached to each other. Therefore, what remains to be specified is the variety of positions and orientations that any attached part might exhibit relative to its base part. For this, a very simple model is chosen in the following.

Let the static spatial dependencies between the parts of an assembly be expressed as a kinematic tree. Furthermore, let $\mathcal{C}_{\mathcal{A}}^{k}$ denote the space of poses that a part model M_{part}^{k} with $k = 1, \ldots, j$ can exhibit relative to its respective base part. Define the pose space of the root node part model $\mathcal{C}_{\mathcal{A}}^{1}$ relative to the world coordinate system. Under the assumption

that all part pose spaces $\mathcal{C}_{\mathcal{A}}^{k}$ are static and mutually independent, the assembly pose space $\mathcal{C}_{\mathcal{A}}$ can be composed by means of the cartesian product

$$\mathcal{C}_{\mathcal{A}} = \mathcal{C}_{\mathcal{A}}^{1} \times \cdots \times \mathcal{C}_{\mathcal{A}}^{j}. \tag{3.12}$$

We define that the kth part is expected to be placed at a fixed position and orientation relative to its base part, called reference translation and orientation. Around this reference, $\mathcal{C}_{\mathcal{A}}^{k}$ might deviate in all 6 DOF within some enclosing hypercuboid. Formally, this model of $\mathcal{C}_{\mathcal{A}}^{k}$ can be expressed as

$$\mathcal{C}_{\mathcal{A}}^{k} = \begin{pmatrix} \Delta x \cdot \phi_1 \\ \Delta y \cdot \phi_2 \\ \Delta z \cdot \phi_3 \\ \Delta \alpha \cdot \phi_4 \\ \Delta \beta \cdot \phi_5 \\ \Delta \gamma \cdot \phi_6 \end{pmatrix} + \begin{pmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{pmatrix}, \ \phi_1, \ldots, \phi_6 \in [-1, 1], \tag{3.13}$$

where x, y, z denote the reference translation and the reference rotation is specified as X–Y–Z fixed angles with γ roll, β pitch, and α yaw as defined in [Cra89, pp.45–48]. Note that this rotation angle convention is used consistently within this thesis. Additionally, $\Delta x, \Delta y, \Delta z \in \mathbb{R}$ define the maximal deviation from the reference translation and $\Delta\alpha, \Delta\beta, \Delta\gamma \in [0, \pi]$ the maximal deviation from the reference rotation. Figure 3.10 illustrates how the pose space $\mathcal{C}_{\mathcal{A}}^{\text{screw}}$ of a screw is modeled with this approach. In this example, Δy models the possible screw translation along its thread axis as it is twisted into the block. An arbitrary screw rotation about its thread is modeled with $\Delta\beta = \pi$. Furthermore, possible fault configuration are accounted for by specifying small values $0 < \Delta\alpha \ll \pi$ and $0 < \Delta\gamma \ll \pi$.

Principally, a pose space $\mathcal{C}_{\mathcal{A}}^{k}$ isn't independent from the other pose spaces $\mathcal{C}_{\mathcal{A}}^{l}$ with $l = 1, \ldots, j$ and $l \neq k$. This is so because a specific part pose $\mathbf{x}_{\mathcal{A}}^{k}$ can become physically impossible, e.g. if another part is already vacating the associated position in space. Because this interdependency between part pose spaces is neglected in the model of $\mathcal{C}_{\mathcal{A}}$, Eqn. (3.12) and (3.13) strictly speaking capture not only physically possible poses. Instead, they yield a rough approximation to the real space of physically possible poses. The model nevertheless sufficiently constrains the assembly pose estimation process. And it is general enough to represent regular and fault configurations of all major conventional joint types like revolute, prismatic, cylindrical, planar, screw, and spherical joints. What is even more important, one can easily sample from $\mathcal{C}_{\mathcal{A}}^{k}$ by drawing scalars $\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6$ from a uniform distribution in the interval $[-1, 1]$ and applying Eqn. (3.13). Concatenating the sampled part poses analog to Eqn. (3.12) then yields a sampled assembly pose.

Sampling assembly part poses as proposed above yields a pose distribution which is uniform over the translation subspace but not over the rotation subspace. The latter phenomenon arises from the employed rotation angle representation. For uniformly distributed values of roll, pitch, and yaw angles, the encoded orientations are distributed

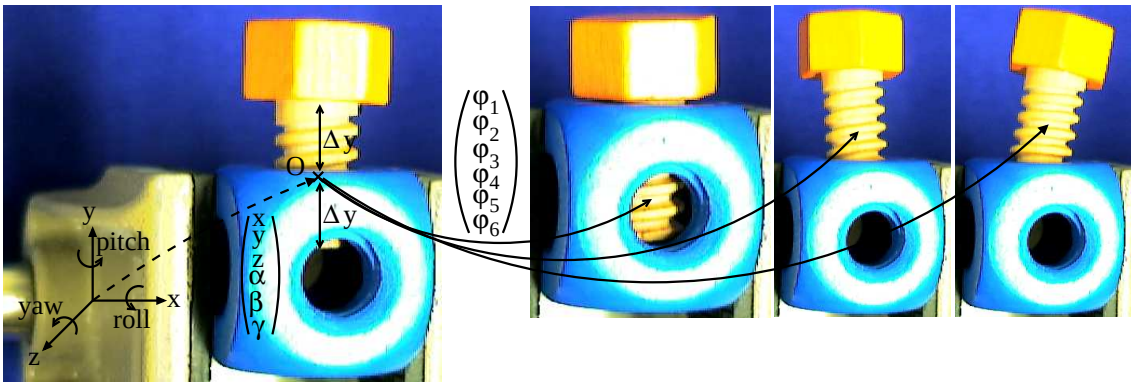


Figure 3.10: Representation of $\mathcal{C}_{\mathcal{A}}^{\text{screw}}$ for a screw-block assembly. Left: The screw origin O is at reference $(x, y, z, \gamma, \beta, \alpha)^{\mathrm{T}}$ w.r.t. the fixed coordinate system of its base part (drawn at the lower left for illustrational purposes). Arrows illustrate the maximal deviation Δy. The other screw configurations correspond to different parameterizations of $(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6)^{\mathrm{T}}$

non-uniformly. However, the resulting distribution is smooth enough for the KPF to determine assembly poses accurately. Furthermore, rotation angles are easier to interprete and specify for humans as alternative representations like unit quaterions. This is important here because the current system implementation relies on a manual definition of reference pose values and their maximal deviations. The automatic determination of such parameters has already been attempted. For example, Sinha et al. [SGPK98] retrieve articulation information of assemblies by analyzing their CAD surface models. Unfortunately, such approaches so far only cope with planar, cylindrical, and spherical joints between assembly parts and thus aren't mature enough for our purposes.

3.5 Summary

This chapter has covered all important aspects of the preparation of models for automated visual assembly inspection. It initially provided "the big picture" of the system modularization and the interaction of the functional units. It also motivated, why CAD models have been and still are a fundamental source of information for computer vision tasks in the context of automated manufacturing. The system overview concluded with the enumeration of four requirements to assembly representations. In this way, the driving forces were documented that dominated all subsequent considerations and design decisions.

The first section of this chapter then presented an approach for the automatic extraction of model features from CAD models. Here, we learned that the chosen approach is a

versatile combination of work from [Han01] and [EKSH05]. It can automatically retrieve sets of contour edge features from CAD descriptions of single rigid assembly parts. The model feature visibility is automatically precomputed for the case of scenes with single parts and efficiently organized in look-up tables.

The second section initially discussed, how model feature visibility can be modeled in the context of assemblies. As a result of the considerations, the visibility map concept for single parts was extended to the multi-part case by specifying a fast two-stage visibility determination process. The automatic model feature extraction stage and the two-stage visibility determination process were reported in [SHS+04] which was published and presented at the DAGM 2004 Conference in Tübingen.

When analyzing the time complexity of the visibility determination process, it became clear that performance gains can be achieved by reducing the size of contour edge feature sets. Therefore, this thesis contributed a new approach for the optimization of part feature sets. The approach uses two different optimization strategies in order to account for the different requirements of correspondence-based or sampling-based (and appearance-based) pose estimation techniques. Preliminary experimental investigations indicated that the automatically extracted sets of contour edge features can be reduced significantly in their size while retaining the information needed for solving the assembly pose estimation task. It is important to note that both optimization strategies rely on the specification of only one parameter which might be determined in advance for a given application domain. This means that large parts of the model preparation effort can be automated, namely the extraction and optimization of assembly part models.

The last section presented how part models are combined to form assembly models. It first gave a more detailed specification of the application context within which assembly models are embedded. Most importantly, a set of operations was defined that defines the interface between assembly models and the system module for assembly pose localization. Afterwards, the structure of assembly pose specifications was discussed. Here, we learned that assembly models within this thesis represent static ”being attached to” relations between connected parts together with dynamic position and orientation information via kinematic trees. Furthermore, this thesis contributed a versatile extension of the concept of constrained models: By additionally attributing kinematic trees with reference pose and maximal pose deviation information, assembly models obtain an approximate description of the space of possible part poses. Finally, this description was shown to facilitate pose sampling in a simple and straight-forward manner. This property is very important for the new KPF that is presented in the next chapter.

4 Assembly Inspection

Model-based computer vision demands high quality models. If the employed models don't live up to this strong expectation, none of the algorithms and approaches presented here are able to turn water into wine. Accordingly, the previous chapter dedicated much effort to the model preparation topic. This part illustrates how assembly models are put into action. More specifically, it describes the online activities of the proposed system for automated visual assembly inspection.

The considerations start with discussing assembly inspection task specifications, i.e. the set of information that is needed to unambiguously define inspection scenarios such that online visual inspection can proceed in an automated fashion. The following section then provides a fine-grained presentation of the new kernel particle filter for assembly pose estimation that is contributed by this thesis. Based on this description, it is finally explained how part completeness and pose integrity classification can be performed, once an assembly pose has been recovered.

4.1 Inspection Task Specification

Before our visual assembly inspection system can enter online operation, it needs to be handed task specific information from the manufacturing environment. It is important to discuss this information set because it concisely represents important assumptions on which the proposed inspection system is built. For example, we have learned in the introduction of Chap. 2 that the system expects to be told which assembly is inspected next. This expectation is justified because production lines must track the identity of any object that proceeds along them. Accordingly, object detection and recognition topics were excluded from the scope of this thesis. In the following, the collection of external task specific information is termed *assembly task specification*. The term *scene model* is used as a synonym.

In the previous chapter, Fig. 3.1 has already illustrated that an assembly task specification or scene model is used to initialize the proposed inspection system. As long as the type of inspected assembly or any other item of the scene model information doesn't change, this initialization needs only to be performed once. The initialization data is currently represented as a set of XML documents. In short, assembly task specifications collect the following data from the manufacturing environment:

- Assembly ID: An identifier of the assembly that is currently being inspected. The identifier enables the inspection system to select the corresponding assembly model from a database that maintains all model preparation outcomes.

- Interior camera parameters: Description of the camera's effective focal length, decentering, and lens distortion (the current system implementation uses the "Brown-Conrady" distortion model that was proposed by Brown in [Bro66]). Based on this parameter set, input images can be rectified prior to any further processing. After rectification, the imaging transformation is represented with a simple pinhole camera model. This parameter set is determined together with the exterior camera parameters by applying standard camera calibration techniques.

- Exterior camera parameters: Parameterization for the 6 DOF that relate the coordinate system of the inspection camera to the world coordinate system. Together with the camera's effective focal length, this parameter set is needed for the `Transform` operation from Chap. 3.4.

- Root node part pose initialization: Just as any other pose estimation approaches known from the literature, the new KPF filter that is presented in this chapter performs much more robustly, if approximate pose information is available as an initial pose hypothesis. Nevertheless, the KPF doesn't depend on a full assembly pose initialization. Instead, it only requires that the root node pose is known in advance. This information is present in any manufacturing environment where the root node part is manipulated in a very controlled manner, e.g. by mounting it on a fixture or gripping it with a robotic manipulator. The root node pose might also be located with any of the pose estimation approaches for single rigid objects that have been presented in the related work chapter.

- Localization targets: In many situations, only some parts of an assembly must be localized. Accordingly, this information identifies all assembly parts for which pose information must be recovered. The KPF uses this list to restrict the pose estimation efforts to the smallest connected subtree of the kinematic tree that represents the assembly part dependencies.

- Classification targets. Analog to the localization targets above, this data lists the identifiers of those parts for which classification results must be obtained. Note that this set must be a true subset of the localization targets, i.e. classification is only possible for parts that have also been localized.

- Valid pose space: Defines the space of valid assembly poses. This space is a strict subspace of $\mathcal{C}_{\mathcal{A}}$ from (3.12) and (3.13). The description of the valid pose space is needed in order to distinguish correct, i.e. valid, from fault assembly configurations. It might be represented in a way similar to the representation of $\mathcal{C}_{\mathcal{A}}$.

4.2 Assembly Localization

With regard to the system overview that was presented in Fig. 3.1, the parts denoted as ”Offline Model Preparation” and ”Initialization” have now been covered in detail. Concerning the remaining issue of ”Online Inspection”, assembly pose localization is addressed next.

In Chapter 2.4, sampling-based approaches were identified to be the most promising methods for assembly localization. A key advantage is that they don't depend on solving the correspondence problem. Due to the self-occlusion that often arises from monocular observations of multi-part assemblies, this property is very valuable. Furthermore, sampling-based approaches work well with memory-efficient assembly model representations. This sets them apart from appearance-based techniques. Consequently, it was decided to pursue a sampling-based assembly pose localization approach in this thesis. More specifically, the chosen method is a unique combination, modification, and extension of recent particle filtering techniques.

Figure 4.1 presents the functional modularization of the assembly localization process that is contributed by this thesis. The process starts whenever a new input image becomes available for inspection. The corresponding data element is positioned in the top left figure corner. It's underlined to indicate its status as major input- or output resource. The upper half of the figure illustrates the preprocessing steps that are applied to each new image, namely rectification, SUSAN edge detection [SB97], and chamfering [Bor86]. Parallel to the image preprocessing, the localization process is reinitialized whenever a new assembly is presented to the observing camera. This process is visualized in the box at the lower left corner of the figure. It employs the assembly pose space specification $\mathcal{C}_{\mathcal{A}}$ from Chap. 3.4.3 to construct an initial particle set that represents the prior knowledge on assembly poses. The concept of particle filters, particles and their initialization is explained in full detail in Chap. 4.2.2, as indicated by the box subtitle.

The initial particle set is passed to the extended kernel particle filter which is detailed in Chap. 4.2.3-4.2.7. This phase is depicted as the big box in the lower right half of the figure. Its first major subpart is a conventional particle filter. The latter transforms the input particle set to a new particle set that represents a coarse interpretation of the latest image measurement. Within this transformation, the assembly model is projected to the image plane under many different hypothetical assembly poses. Each projection is then matched against the current image. As indicated in the figure, this step involves evaluating the image cues that are detailed in appendix C. Furthermore, it makes use of the reference pose information from Eqn. (3.13). The resulting particle set is subsequently refined by several iterations of mean shift and weight update processes. The weight updates involve projecting the assembly model and evaluating the image cues in a similar manner than the SIR particle filter process. The final particle set is used to recover the assembly pose. Furthermore, if new images of the inspected assembly become available, the particle set can be fed back to the SIR particle filter without reinitialization.

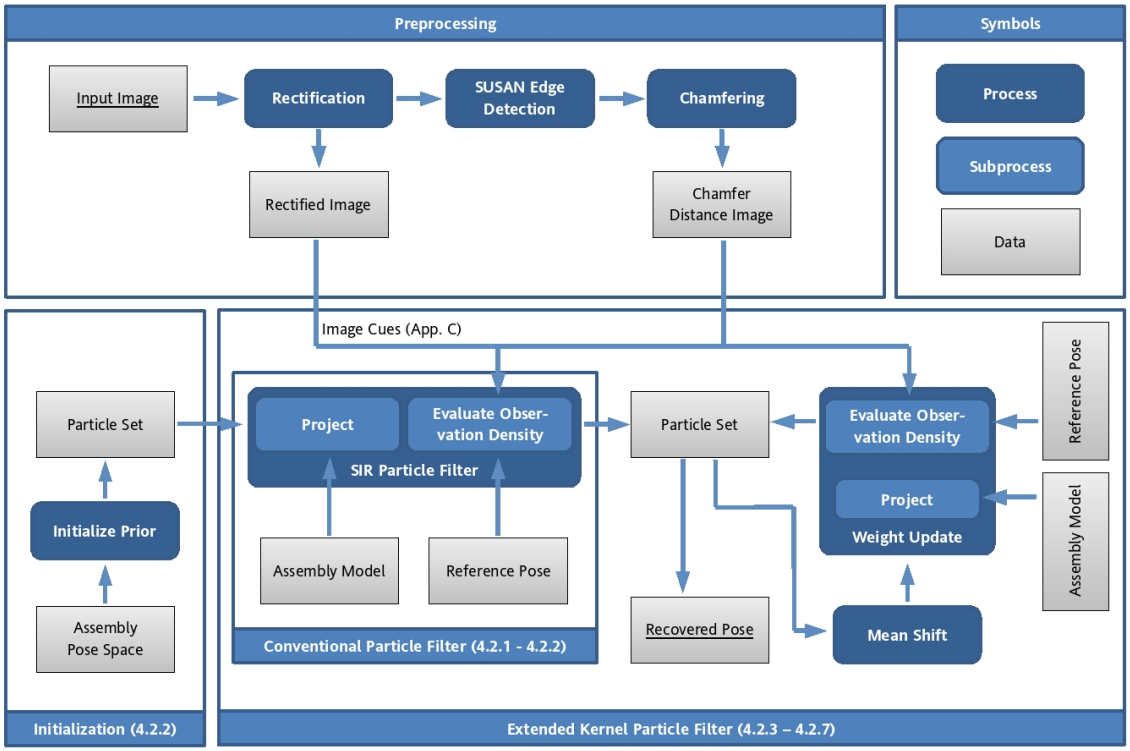


Figure 4.1: Architectural overview of the assembly localization process

Now that the modularization of the localization process has been introduced, its fundamental image processing steps can be explained in more detail. This is done in Fig. 4.2. With respect to the architectural overview in Fig. 4.1, it zooms into the interaction details of the extended kernel particle filter module. Most importantly, this figure introduces symbolic representations of the interacting elements that will be used throughout the theoretical considerations in the remainder of this chapter.

The illustrated process starts with the particle set $\{\mathrm{s}_{t-1}^n, w_{t-1}^n\}$ in the top left corner of the figure. It can be interpreted as hypothetical assembly poses that are associated with weights. By means of the weighted pose hypotheses, the localization module represents all knowledge that is has of an assembly, after evaluating a sequence of images that capture the assembly up to a scalar time step $t-1$. At the subsequent time step t, this knowledge is augmented with the information obtained from a new image observation of the assembly. This is done by performing the depicted optimization loop that was coarsely introduced in the previous figure. Note that the loop is performed for each pose hypothesis and for several times, generating a set of weighted optimized hypotheses $\{\mathrm{s}_t^n, w_t^n\}$ of the current time step t. The latter is finally used to obtain an estimate of the assembly pose $\hat{\mathrm{x}}_t$ in the observing image at time step t, as shown at the lower right side of the figure.

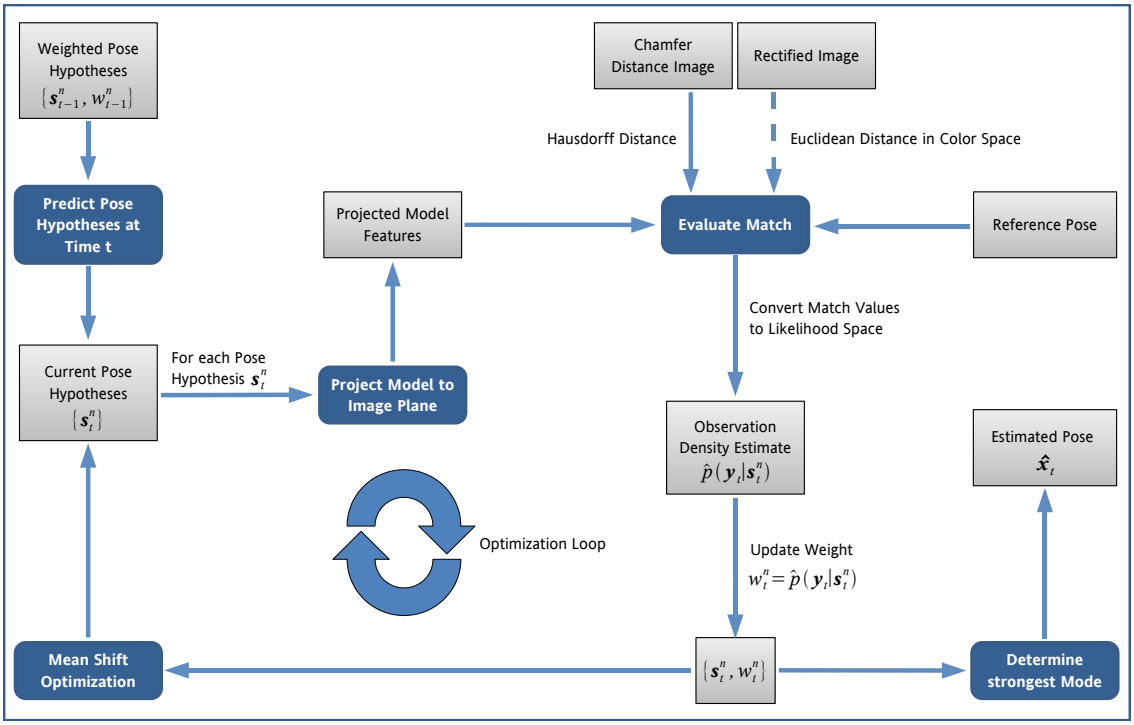


Figure 4.2: The core image processing activities of the assembly localization module

Conceptually, the optimization loop carries out two principal tasks. Given a set of hypotheses that predict the assembly pose at time step t, the first task is to update the hypotheses weights. As stated before, the resulting weighted set represents the knowledge of the assembly pose in the observing image sequence at time step t. The second task is to optimize the predicted $\{\mathrm{s}_t^n\}$. Afterwards, the loop can start all over again.

The first principal task is initiated by projecting the assembly model to the image plane. As already explained in the context of the previous figure, each projection corresponds to a specific pose hypothesis. The resulting projections are then individually matched against the current image by means of evaluating image cues. Figure 4.2 illustrates that the cues are based on the Hausdorff distance and the Euclidean distance in some color space. The latter is depicted with a dashed arrow, in order to indicate that the dependency of the system on color information is purely optional. Afterwards, the system converts each match score into a likelihood value. Note that this step assumes normally distributed cue responses, which is explained in detail in Chap. 4.2.2. The likelihood value is used as an estimate of the observation density $\hat{p}(\mathrm{y}_t|\mathrm{x}_t)$ which is assigned as new hypothesis weight w_t^n. Once the task of updating the hypotheses weights has been accomplished in this way, the second principal task is then carried out by means of mean shift optimization on the weighted pose hypotheses $\{\mathrm{s}_t^n, w_t^n\}$.

Chapter 1 has announced that this thesis proposes a system for the localization of assemblies from *single* monocular images. Considering this statement, it may come as a surprise that the localization process introduced above is apparently designed to deal with *sequences* of images. However, the findings aren't contradictory. For the proposed localization approach, a single image is just a special type of image sequence, namely one that provides measurements only for time step $t = 1$. In this case, the weighted hypotheses set of the previous time step $t = 0$ is obtained from a uniform random sampling from the space $\mathcal{C}_\mathcal{A}$ as stated earlier in this paragraph. The weights of this initial hypotheses set are distributed uniformly, too.

The architectural overview from Fig. 4.1 shows that the image processing operations introduced above implement different particle filtering techniques. It remains to be specified, though, what *particle filters* are. Most generally, they are categorized as Sequential Monte Carlo methods. Their key idea is to approximate unknown probability distribution functions (pdfs) with sets of weighted random samples such as the $\{\mathbf{s}_t^n, w_t^n\}$ introduced above. In the context of computer vision tasks, the target pdfs usually model the state of dynamic systems like moving objects. Particle filters observe how such systems evolve over a series of scalar-valued time steps. At each time step t, a new image measurement is made and the sample set approximation is updated to incorporate the latest observation.

The sequential update process of particle filtering is very versatile in the context of computer vision tasks like tracking because it can process image streams online, whereas batch procedures need the full set of image measurements in advance. Accordingly, many different particle filters have been proposed for tracking. As tracking is closely related to the pose localization task, it deemed plausible that particle filters should also be capable of performing 3D assembly pose estimation when this dissertation was started. This principle turned out to be sound. Therefore, some important particle filters for visual tracking are outlined in the following. Then, it is explained how the task of assembly pose localization can be addressed with conventional particle filtering. Afterwards, it is shown how kernel particle filtering can be used in order to obtain a more compact representation of the modes of a pdf. The remaining subsections step-by-step introduce important modifications that together form the new extended kernel particle filter for the localization of multi-part assemblies.

4.2.1 Particle Filtering for Visual Tracking

A number of different approaches to particle filtering have influenced this thesis. In order to account for each of them appropriately, they are outlined in the following. Note that they have entirely been proposed for performing object tracking within image sequences. It is discussed at the end of this paragraph in which important respects this task differs from general pose estimation for assembly inspection.

A widely recognized paper was published by Blake & Isard [IB98a] who were among the first authors that used particle filtering to accomplish computer vision tasks. In their work, they propose the CONDENSATION algorithm which they successfully apply to the problem of visual tracking. For example, they track the contour model of a person walking in front of other persons, the contour model of a dancing girl's head, and the model of a human hand. The first two examples require the determination of 6 DOF whereas the hand model exhibits 12 DOF. Note that the hand was moved over a very cluttered desk. However, in all cases the object motion was restricted to affine transformations which simplifies the tracking task considerably. Furthermore, a strong motion model was available. It was learned prior to CONDENSATION tracking in a bootstrap procedure that employed conventional Kalman filtering on video footage without (or only little) clutter.

The work of Blake and Isard received much attention in the computer vision community because it became apparent that, in the context of object tracking, particle filtering offers advantages over conventional techniques like Kalman Filtering. A major reason for this is the finding that object tracking frequently involves the approximation of non-Gaussian and multi-modal pdfs, based on observation pdfs that are also non-Gaussian and multi-modal. Deutscher et al. [DBNB99] illustrate this problem in the context of tracking human motion. Such pdfs violate the fundamental assumption of Kalman filters and extended Kalman filters that the respective pdfs are Gaussian. On the contrary, particle filtering doesn't impose any restrictions on the approximated pdfs.

A large body of literature has sprung from the original proposal of Blake and Isard. For instance, Fritsch [Fri03] extends it with the incorporation of symbolic context knowledge in order to recognize manipulative gestures in an office and an assembly construction domain. Nevertheless, it is important to note that CONDENSATION particle filtering becomes computationally intractable for state spaces of a dimension higher than 10 to 15. The basic problem is that a suitable approximation of pdfs requires particle numbers to increase exponentially in the dimension of the state space. Consequently, standard particle filtering is computationally intractable for the pose localization of multi-part assemblies, as their state space easily exceeds a critical number of dimensions.

Chang & Ansari [CA03, CA05] and Schmidt et al. [SKF06] recently proposed *kernel particle filtering* to alleviate the above mentioned problem. By interpreting particles as state space positions around which kernels can be shaped, they combine particle filtering with kernel density estimation. This approach offers the advantage that positions between samples can be interpolated via kernel density estimation. The kernel representation thus allows to approximate a pdf with rather sparsely distributed particles. Furthermore, the authors note that quite frequently one isn't interested in approximating a whole target pdf but rather needs to find its modes. They consequently apply a local mode finding approach, namely the mean shift algorithm. An instructive tutorial that demonstrates the application of this standard technique in the domain of image segmentation can be found

in [CM02]. The application of mean shift iterations on sets of particles yields a compact representation of the modes of a high-dimensional pdf. With this technique, Schmidt et al. manage to track the articulated 3D model of a human torso and arm with 10 DOF in real-time performance on a standard PC. In this thesis, we also follow a kernel particle filtering approach. Several extensions and modifications are contributed that improve the measurement accuracy and precision of the respective assembly pose localization.

The density estimation that is inert to kernel particle filtering demands the specification of bandwidth parameters. Their number grows linear in the dimension of the sample space. However, Chang & Ansari [CA05] suggest that one bandwidth parameter is sufficient, if the sample space undergoes a variance normalization. This thesis takes up the idea of Chang & Ansari and extends it with an automatic bandwidth selection scheme. The latter is similar to an approach that was proposed by Comaniciu et al. [CRM01] in the context of mean shift image segmentation. The resulting KPF still depends on one bandwidth parameter but behaves more stable.

Deutscher at al. [DBR00] published an alternative idea in order to dampen the amount of particles needed for pdf approximation. The authors generate particle sets in a layered fashion. Each layer contains a small number of particles that are sparsely distributed in the state space. By manipulating the function that associates weights to particles, Deutscher and colleagues manage to iteratively migrate the particles to the modes of the target pdf. Because their approach uses ideas from simulated annealing procedures, they name it *annealed particle filtering*. This thesis incorporates a new KPF extension that is related to the idea of weighting function manipulation in the course of mode detection.

Gavrila & Davies [GD96] proposed a multi-view approach for the 3D model-based tracking of humans. Here, they use a *search space decomposition* strategy in order to reduce the complexity of the tracking task. It proceeds by first determining the position of head and torso. Afterwards, the model parts representing arms and legs are fitted to the image independent from each other. The advantage of this approach is that it divides the state space into three subspaces within which the subsequent search is computationally tractable. However, the proposed partitioning of the search space is quite ad hoc as the authors don't state a decomposition strategy. In this thesis, the proposed KPF employs a heuristic that dynamically partitions the state space into subspaces of constant dimensionality. This strategy is fundamental for obtaining a KPF that can perform pose estimation for assemblies that are composed from multiple parts.

In Summary, all mentioned approaches have contributed advances in the field of visual tracking. Their capability to localize even articulated objects is appealing. However, it must be noted that the presented tracking approaches rely on two key assumptions that can't be made in the more general case of pose estimation for assembly inspection. First, all mentioned approaches depend on a full pose initialization to be given in advance. Second, based on this initialization, the approaches determine the object pose in subsequent images by exploiting a tracking assumption. The latter assumes that an object can't

move far in the small time that elapses in between the recording of two successive image frames. The tracking assumption helps to exclude large image and pose space regions from further consideration and thus provides a powerful search constraint. Furthermore, in some of the proposed tracking scenarios, motion models are learned in bootstrap processes which also guide the tracking.

In a case like assembly pose localization, where all the above mentioned assumptions and constraints are missing, any single of the proposed tracking approaches wouldn't take us very far. Therefore, the most important contribution of this thesis is to combine and extend the ideas presented above in order to obtain a new kernel particle filter that performs the task of assembly localization.

4.2.2 Particle Filtering for Assembly Pose Estimation

In order to gain a thorough understanding of the proposed new kernel particle filter, the theoretical foundation of particle filtering in general is shortly recapitulated first. We then encounter the algorithmic implementation of an important particle filter subtype whose working principle is closely related to kernel particle filtering. The presented algorithm is very simple but illustrates all the better, how our assembly models are employed in the process of particle filtering. Taken together, the considerations of this section will introduce us firmly to assembly pose estimation from sequences of image measurements. We will also see that the method is applicable to the special case of single images. Once this has been established, it is discussed in the subsequent sections how kernel density estimation techniques can be mixed with particle filtering and how this combination can be refined in order to increase the localization accuracy and precision for a fixed number of samples.

Theoretical Foundation

Consider the example assembly that is illustrated in Fig. 4.3(a). In order to obtain its model, the CAD models from subfigure 4.3(b) were processed with the approach from Chap. 3.2 to automatically extract the part model features that are illustrated in subfigure 4.3(c). The latter were optimized, following the procedure described in Chap. 3.3. Finally, the optimized part models were manually composed to an assembly model which is sketched in 4.3(d) together with its kinematic tree. Our overall aim is now to determine the assembly model pose from image measurements like the one shown in subfigure 4.3(e). This task is first considered theoretically in the following.

Let $\mathbf{x}_{\mathcal{A},t} \in \mathcal{C}_{\mathcal{A}}$ denote an assembly pose vector of the form given in Eqn. (3.11). From the perspective of particle filtering, such a pose vector describes the state of an arbitrary

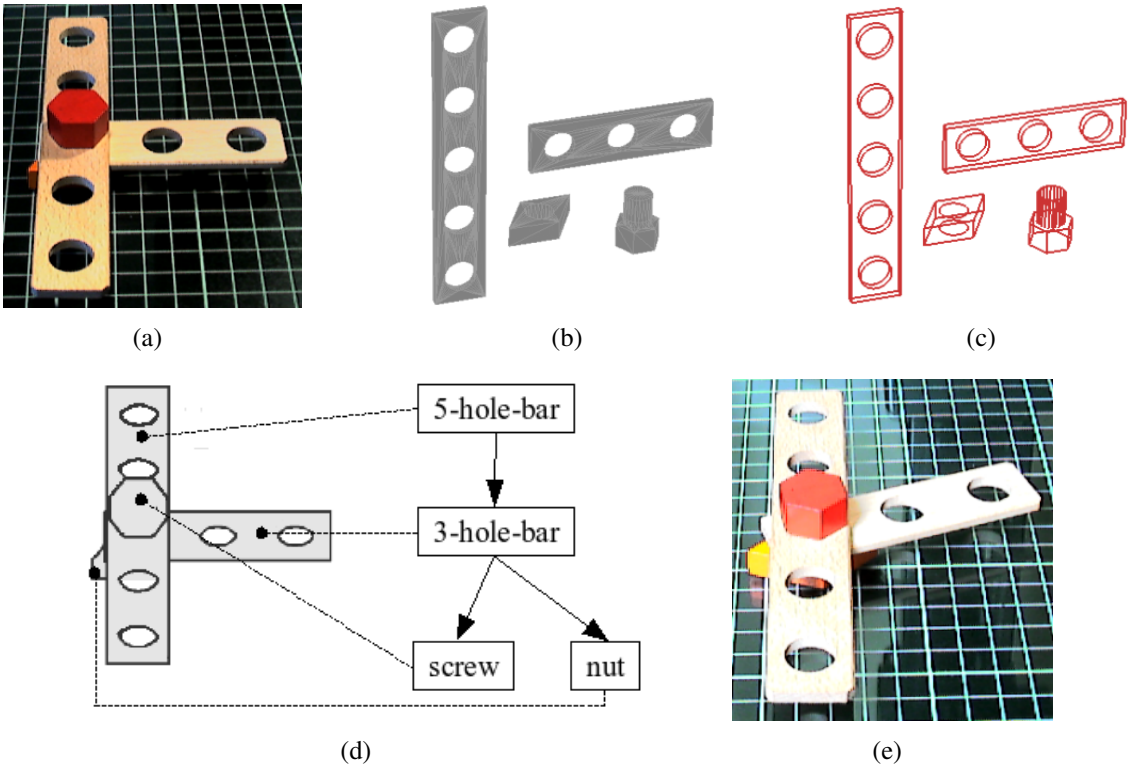


Figure 4.3: The assembly localization scenario. a) An assembly of two bars, a screw, and a nut under reference configuration. b) Input CAD models for the model preparation stage. c) Automatically generated part models. d) The final assembly model and a sketch of its kinematic tree. Here, the assembly is configured according to the reference pose values from $\mathcal{C}_{\mathcal{A}}$. Black lines denote visible contour edge features. e) A new image that contains the assembly with unknown pose parameters

system and is therefore also termed *state vector* or *system state*. Generally, system states are expected to change over time. For ease of notation, the system state or assembly pose at time step $t \in \mathbb{N}$ is expressed as $\mathbf{x}_t$ in the following. An estimate of the true system state will be denoted as $\hat{\mathbf{x}}_t$, consistent with the notation from Fig. 4.2. Furthermore, let $\mathbf{y}_t$ be an image measurement which observes the assembly at time step t, and let the history of individual image measurements be denoted by $\mathcal{Y}_t = \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$. The overall task of particle filtering in such a setting is to determine $p(\mathbf{x}_t|\mathcal{Y}_t)$.

The pdf $p(\mathbf{x}_t|\mathcal{Y}_t)$ is a probabilistic characterization of the knowledge about assembly pose $\mathbf{x}_t$ that is gathered from the history of image measurements $\mathcal{Y}_t$. In order to construct this characterization, particle filtering relies on a *system model* and an *observation model* that are written as

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) \text{ for } t \geq 1 \tag{4.1}$$
$$p(\mathbf{y}_t|\mathbf{x}_t) \text{ for } t \geq 1 \tag{4.2}$$

The system model (4.1) captures the expected system dynamics, independent from any observation measurements. Particle filtering assumes that this model is a first-order Markov process that depends solely on the knowledge of the previous state. The observation model (4.2) specifies a pdf that is also termed *observation density* in the following. It reflects how well the latest image measurement complies with a specific assembly pose.

Conceptually, particle filtering constructs the pdf of the current state $p(\mathbf{x}_t|\mathcal{Y}_t)$ by implementing a *recursive Bayesian filter*. Such filters operate in two steps that are repeated for each new measurement. The first step recursively processes the result of the previous iteration, $p(\mathbf{x}_{t-1}|\mathcal{Y}_{t-1})$, by updating it with the expected system dynamics. Formally, this update or prediction step is described as [IB98a]

$$p(\mathbf{x}_t|\mathcal{Y}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathcal{Y}_{t-1})d\mathbf{x}_{t-1}. \tag{4.3}$$

At time step $t = 1$, there clearly is no previous state pdf to be updated and the history of measurements $\mathcal{Y}_{t-1}$ is the empty set. Therefore, it is assumed that an overall prior $p(\mathbf{x}_0)$ is given such that one can define $p(\mathbf{x}_0|\mathcal{Y}_0) \equiv p(\mathbf{x}_0)$. The second step then accounts for a new incoming image measurement by augmenting the intermediate prediction step result with the observation model (4.2), yielding the *posterior* pdf $p(\mathbf{x}_t|\mathcal{Y}_t)$ at time step t. Assuming that the image measurements depend conditionally only on the current time step, this is done by applying Baye's rule as a propagation step

$$p(\mathbf{x}_t|\mathcal{Y}_t) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathcal{Y}_{t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathcal{Y}_{t-1})d\mathbf{x}_t}. \tag{4.4}$$

Arulampalam et al. [AMGC02] provide a detailed introduction to recursive Bayesian filtering and various algorithmic implementations thereof. They make clear that Kalman filtering provides the exact solution in the highly restricted case where all involved pdfs are Gaussian and the functions that are inherent to the observation and system model are linear. For non-Gaussian pdfs, particle filtering is a simple and effective way to obtain an approximate solution.

Particle filtering generates N_s discrete *samples* $\{\mathbf{s}_t^n\}_{n=1}^{N_\mathrm{s}}$ within the state space considered at time step t that can be interpreted as hypothetical instantiations of the system state $\mathbf{x}_t$. In our case, the state space is the space of physically feasible assembly poses as

described in Chap. 3.4.3, i.e. $\mathbf{s}_t^n \in \mathcal{C}_\mathcal{A}$ for $1 \leq n \leq N_\mathrm{s}$. Each sample is associated with an individual weight w_t^n. A weighted sample is called *particle*. The outcome of a particle filtering iteration, i.e. a *particle set* $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$, is a discrete weighted approximation of the posterior pdf. The approximation is formally expressed as

$$\hat{p}(\mathbf{x}_t|\mathcal{Y}_t) = \sum_{n=1}^{N_\mathrm{s}} w_t^n \delta(\mathbf{x}_t - \mathbf{s}_t^n), \tag{4.5}$$

where the Dirac δ-function provides the transition from the continuous to the discrete space. The more particles are contained in the set, the closer this approximation is to a functional representation of the posterior.

The core algorithmic problem of particle filtering is *how* to generate a particle set that approximates the posterior pdf as in Eqn. (4.5). Ideally, one would like to sample it directly from the posterior pdf. However, this would demand a functional representation that doesn't exist in situations where particle filtering is applied. Instead, many different particle filtering algorithms have been proposed, each manipulating the sample set in its own way in order to arrive at the posterior pdf approximation. According to [AMGC02], the main differences of particle filters lie in the way they generate weights and how they compensate particle set *degeneration*. The latter describes the problem that, after some iterations, a particle set might contain a large number of samples whose weights are almost zero and thus effectively don't contribute to the solution any more. This problem has been addressed with various resampling techniques. In this thesis, a mean shift based approach is used which is presented in Chap. 4.2.3 to 4.2.5.

The generation of weights has been approached in ways that mainly vary in the additional assumptions being made. However, the underlying theoretical foundation usually is an importance sampling approach. The principle of *importance sampling* is summarized in appendix B. It leads to weights w_t^n that are chosen according to

$$w_t^n \propto w_{t-1}^n \frac{p(\mathbf{y}_t|\mathbf{s}_t^n)p(\mathbf{s}_t^n|\mathbf{s}_{t-1}^n)}{q(\mathbf{s}_t^n|\mathbf{s}_{t-1}^n, \mathbf{y}_t)}, \tag{4.6}$$

where $p(\mathbf{y}_t|\mathbf{s}_t^n)$ and $p(\mathbf{s}_t^n|\mathbf{s}_{t-1}^n)$ are point-wise evaluations of the measurement and system model, and $q(\mathbf{s}_t^n|\mathbf{s}_{t-1}^n, \mathbf{y}_t)$ evaluates a *proposal distribution* $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ at specific sample positions. The proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ is used to generate new samples $\{\mathbf{s}_t^n\}_{n=1}^{N_\mathrm{s}}$ from the samples of the previous time step. If the weights of the new samples are updated according to (4.6), the resulting particle set is a valid representation of the posterior pdf.

Note that we're allowed to choose the proposal distribution $q(\cdot)$ freely. This enables us to fully control the state space regions from which new samples are drawn. Clearly, the

Algorithm 1 CONDENSATION

Input: $\mathbf{S}_{t-1} \leftarrow \{\mathbf{s}_{t-1}^n, w_{t-1}^n\}_{n=1}^{N_\mathrm{s}}$, new image measurement $\mathbf{y}_t$

// For single images, let $t = 1$ and initialize $\mathbf{S}_{t-1}$ from prior $p(\mathbf{x}_0)$

1: **For all** $n = 1 : N_\mathrm{s}$ **do**
2: Choose $\mathbf{s}_{t-1}^k$ randomly out of $\mathbf{S}_{t-1}$ with probability w_{t-1}^k and $1 \leq k \leq N_\mathrm{s}$.
3: Sample $\mathbf{s}_t^n \sim p(\mathbf{x}_t|\mathbf{s}_{t-1}^k)$.
4: Evaluate $w_t^n = \hat{p}(\mathbf{y}_t|\mathbf{s}_t^n)$.
5: **End For**
6: Normalize weights w_t such that $\sum_n w_t^n = 1$.

Output: $\mathbf{S}_t \leftarrow \{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$ // For single images: stop here

particular choice of $q(\cdot)$ is crucial for the performance of the associated particle filter. We will see later that the key to the good performance of kernel particle filtering lies in the fact that it employs a smart proposal distribution. At first, however, a simple choice of $q(\cdot)$ is discussed in the following that leads to a considerable simplification of the weight generation scheme from (4.6).

SIR Particle Filtering and CONDENSATION

The proposal distribution $q(\cdot)$ in Eqn. (4.6) reflects application specific knowledge, namely state space regions of paramount importance. If such regions are known, particles can be exclusively drawn from them instead of sampling from the whole state space. However, quite frequently a separate model for $q(\cdot)$ isn't available. In this case one can simply plug in the system model (4.1) by defining

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}). \tag{4.7}$$

Point-wise evaluation of $q(\cdot)$ at sample positions now reduces expression (4.6) to

$$w_t^n \propto w_{t-1}^n p(\mathbf{y}_t|\mathbf{s}_t^n). \tag{4.8}$$

This choice leads to *sampling importance resampling* (SIR) particle filters. Most interestingly, it will be shown later in this thesis that kernel particle filtering is closely related to SIR particle filtering. Therefore, its working principle is illustrated in the following. This is done by discussing CONDENSATION, which is a well known algorithmic implementation that was originally proposed by Isard & Blake [IB98a].

Algorithm 1 describes one iteration of CONDENSATION. It operates on the particle set $\mathbf{S}_{t-1}$ of the previous time step. As discussed in the previous paragraph, the initial

particles at time step $t = 1$ are created by sampling from the overall prior $p(\mathbf{x}_0)$. In our case, this simply means to draw samples from some distribution over the assembly state space $\mathcal{C}_\mathcal{A}$ by means of the `Sample` operation as described in Chap. 3.4.3. In the absence of any further knowledge, we choose a uniform distribution to create the samples $\mathbf{s}_0^n \sim p(\mathbf{x}_0)$ for $n = 1, \ldots, N_\mathrm{s}$. Each sample is assigned a uniform weight $w_0^n = \frac{1}{N_\mathrm{s}}$.

Once the input particle set $\mathbf{S}_{t-1}$ is available, the CONDENSATION algorithm creates the particle set of the current time step t. Each new particle at time step t emerges from a sequence of three operations. The first step performs *resampling*. This proceeds by randomly choosing a sample with replacement from $\mathbf{S}_{t-1}$. Afterwards, *stochastic diffusion* applies the system dynamics model as stated in line 3. As long as no further information is available, the latter simply adds some zero-mean Gaussian noise to the sample copied from $\mathbf{S}_{t-1}$. If global motion information is available, e.g. because the currently inspected assembly is deliberately moved with robotic manipulators or along an assembly line, it must be incorporated here as well. By means of stochastic diffusion, the algorithm implements the sampling from the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Furthermore, the resampling effects that each copied sample $\mathbf{s}_t^n$ receives a uniform weight $w_t^n = \frac{1}{N_\mathrm{s}}$. The next major algorithmic operation is to *update* the sample weight as stated in line 4. According to Eqn. (4.8), this update step incorporates the latest image measurement information by evaluating the observation model w.r.t. the current sample and image. The next paragraph will illustrate the chosen observation model in more detail. The evaluation step yields an approximation to the true density values up to a proportionality constant.

After performing resampling, stochastic diffusion and weight update, the particle weights are finally normalized as indicated in line 6, which enforces that they sum up to 1. The resulting particle set approximates the current time step's posterior pdf as given in Eqn. (4.5). The assembly pose at time step t can then be recovered in different ways. If the posterior is unimodal and unskewed, a MAP estimate $\hat{\mathbf{x}}_t$ of the expected assembly pose at time step t is given by the posterior's mean

$$\hat{\mathbf{x}}_t = \sum_{n=1}^{N_\mathrm{s}} w_t^n \mathbf{s}_t^n. \tag{4.9}$$

In this thesis, it is assumed that the posterior pdf is multi-modal. For such distributions, the highest local mode can be used to recover the assembly pose as a MAP estimate. In order to determine this mode from a given particle set, the particle (s_t^n, w_t^n) with the highest associated weight can be used as a coarse estimate. In Chap. 4.2.7, it is explained how the KPF proposed in this thesis determines a more robust estimate of the most prominent local mode.

Evaluating the Observation Density

So far, we have learned that particle filtering involves the representation of several pdfs. Before one of them is discussed in more detail in the following, it is important to put forward some words of caution. None of the "estimates" of pdfs that have been presented so far and that are presented in the remainder of Chap. 4.2 are valid statistical probability density estimates. The reason for this finding is that the number of particles that can be employed by a particle filtering implementation doesn't nearly suffice to obtain such estimates. Fortunately, they aren't needed because the overall aim of the proposed system is just to determine the positions in the pose space that correspond to the local modes of the posterior pdf. The true quality of the employed density estimates isn't important, as long as the sparsely distributed particles are sufficient to correctly locate peaks of the posterior. Principally, the particle filtering approach that is proposed in this thesis can be understood as the attempt to approximate a function that has its maxima at the same state space positions than the true posterior pdf.

As explained in the previous paragraph, a particle set representation of the posterior is obtained by choosing weights according to (4.8). Thus, the nth weight must be chosen proportional to the observation density $p(\mathbf{y}_t|\mathbf{x}_t)$, evaluated at the associated sample position $\mathbf{s}_t^n, 1 \leq n \leq N_\mathrm{s}$. This step is very important since it integrates the latest image measurement into the posterior estimate. And it is the first point at which the assembly models are put into action. In order to give a thorough account of the proposed assembly localization approach, it is therefore discussed in the following how the proposed system performs the weight update and estimates $p(\mathbf{y}_t|\mathbf{x}_t)$.

The overall concept of weight updates is illustrated in Fig. 4.4. The figure shows that each particle $\mathbf{s}_t^n$ can be interpreted as hypothetical assembly pose $\mathbf{x}_t = \mathbf{s}_t^n$. For each such pose, the assembly model is transformed to the camera coordinate space by invoking `Transform` as indicated in Chap. 3.4.2. The visible model features are then predicted by means of a `Query` operation (cf. Chap. 3.3.1) and projected to the image plane. Note that this step is the projection operation of the SIR particle filter that is illustrated in the overview figure 4.1. Finally, the observation density is evaluated by rating how well the latest image observation agrees with the current pose hypothesis and its model feature set. The resulting value is used as new particle weight.

It was already visualized in figures 4.1 and 4.2 that an estimate of the observation density $\hat{p}(\mathbf{y}_t|\mathbf{s}_t^n)$ is obtained from the evaluation of different cues. In the current system implementation, these cues are based on edge and color features. However, the approach allows to change or add cues at need, which is useful if other model and image features like texture are available. The cues are applied to the visible model features of each individual assembly part. Each resulting cue strength, also termed *filter response*, is then converted into a likelihood value by employing a Gaussian weighting function. The latter assumes that filter responses are normally distributed with zero mean and a cue specific variance

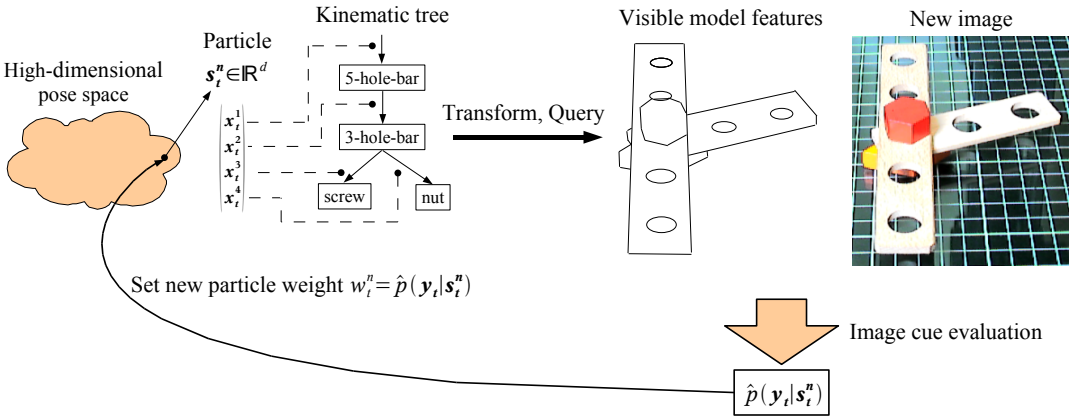
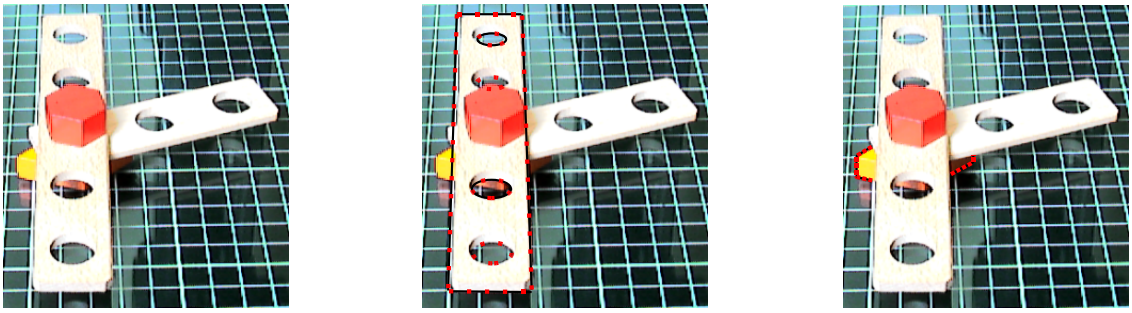


Figure 4.4: The weight update process of a specific particle $\mathbf{s}_t^n$

that can easily be estimated from training data. Though this assumption oversimplifies the real filter response distribution, it still facilitates a robust assembly pose estimation. The likelihood values are finally combined to an approximation of the observation density, up to an unknown but neglectable normalizing constant. Note that this approach simplifies previous work by Sidenbladh & Black [SB01] who additionally employ a background model. It is also related to the approach of Schmidt et al. [SKF06] but uses different cues and a more robust cue combination scheme. A full account of the observation density estimation procedure proposed by this thesis is given in the following.

As indicated earlier in this paragraph, the estimation of the observation density is prepared by obtaining the visible model features of an assembly under pose $\mathbf{x}_t = \mathbf{s}_t^n$. This is done by invoking a `Transform` and `Query` operation. For each assembly part, the visible model features are then individually projected to the image plane and a set of 2D points is created from sampling along the projected model features. Let $\mathbf{z}_t^k$ denote a set of 2D points that have been placed equidistantly along the projected visible model features of part k as illustrated in Fig. 4.5. Note that all points $\mathbf{z} \in \mathbf{z}_t^k$ and the image $\mathbf{y}_t$ share the same coordinate system in the remainder of this paragraph. Each cue can then be defined as a function $f_c(\mathbf{z}_t^k, \mathbf{y}_t)$ where c is a placeholder for the cue type. For the prototype of the proposed system, three cues were implemented, namely the forward distance cue (in short fw), the backward distance cue (bw), and the color cue (col). They are described in appendix C. By means of small pilot studies, it was found that the color cue provided little extra stability for our application domain. Therefore, all experimental investigations of the evaluation section were carried out with edge based cues.

In order to combine cues to an approximate observation density, the individual filter responses must first be transformed to a joint probabilistic space. Assuming that the filter



(a) (b) (c)

Figure 4.5: Sampling from projected model features. a) The 4-part assembly from the previous figures. b) Sample points (red) have been placed equidistantly along the visible contour edges (black) of part 1, resulting in $\mathbf{z}_t^1$. c) Sampling from the visible contour edges of part 4 yields $\mathbf{z}_t^4$

responses are normally distributed and centered at zero, Gaussian weighting functions are employed for this task. They are of the form

$$p(f_c(\mathbf{z}_t^k, \mathbf{y}_t)) \propto \exp\left(-\frac{(f_c(\mathbf{z}_t^k, \mathbf{y}_t))^2}{2\sigma_c^2}\right). \tag{4.10}$$

Here, c denotes either of the three cues presented above, i.e. $c \in \{\mathrm{fw}, \mathrm{bw}, \mathrm{col}\}$. Accordingly, σ_c is a cue specific variance that can be estimated from training data.

In the following, the $p(f_c(\mathbf{z}_t^k, \mathbf{y}_t))$ are interpreted as cue specific approximations of a likelihood function $p(\mathbf{y}_t|\mathbf{x}_t^k)$ that judges how likely different part pose hypotheses $\mathbf{x}_t^k$ are causing the current image measurement $\mathbf{y}_t$. They are also termed *cue likelihoods*. Unlike Schmidt et al., we combine these individual approximations to a more robust estimate of the likelihood function $p(\mathbf{y}_t|\mathbf{x}_t^k)$ by averaging over the weighted cues

$$\hat{p}(\mathbf{y}_t|\mathbf{x}_t^k) = N_{\mathrm{cues}}^{-1} \sum_{c\in\{\mathrm{fw,bw,col}\}} p(f_c(\mathbf{z}_t^k, \mathbf{y}_t)), \tag{4.11}$$

where N_{cues} is the number of cues over which we average[1]. An unknown normalization constant has been neglected here, which isn't problematic because the weights that arise from this estimate are normalized after updating.

[1]Based on the findings from [TvBDK00], forming the average can be considered more robust than using a product rule, if each of the combined densities is subject to strong estimation errors. This is certainly the case because the filter responses aren't really normally distributed.

Assuming that the likelihood functions for different assembly parts explain mutually independent parts of the image measurement, they can be combined to an estimate of the observation density $p(\mathbf{y}_t|\mathbf{s}_t^n)$ at the state space position $\mathbf{x}_t = \mathbf{s}_t^n$ by forming their product

$$\hat{p}(\mathbf{y}_t|\mathbf{s}_t^n) = N_{\text{cues}}^{-1} \prod_{k=1}^{j} \sum_{c\in\{\text{fw,bw,col}\}} p(f_c(\mathbf{z}_t^k, \mathbf{y}_t)). \tag{4.12}$$

Again, an unknown normalization constant has been neglected here. It can be safely ignored in the context of weight generation, because this estimate is still proportional to the true density and we know from (4.8) that this is sufficient. The weights are then normalized as stated in line 6 of Alg. 1.

Chapter 4.1 introduced localization targets as part of an assembly task specification. The set of localization targets consists of the indices of those assembly parts that are asserted relevant for the ongoing inspection task. This concept allows us to specify multiple inspection tasks that consider the same assembly but focus on different part subsets. The latter is important, if inspection planning yields that image measurements from a specific point don't suffice to capture all assembly parts but rather multiple camera perspectives and settings are needed. In that case, it is possible to restrict the product of cue likelihoods in Eqn. (4.12) to apply only to subsets of $\{1,\ldots,k\}$. The remaining parts are nevertheless important, namely for the prediction of visible model features by means of `Transform` and `Query` operations. Their pose must either be explicitly known or defaults to the reference translation and rotation from (3.13).

Sometimes a hypothesized assembly part whose index is among the localization targets yields only an empty set of visible model features. This happens if, with regard to the full considered assembly pose, the part is completely occluded by others. None of the above cues can then be evaluated. Assuming that, after proper inspection planning, localization targets should be at least partially visible within an image measurement, the observation density in such a case receives a value very close to zero. Most importantly, this effects the pose localization module proposed in this thesis is incapable of recovering assembly poses in which localization target parts are completely occluded because these pose hypotheses will always receive insignificant weights.

This paragraph provided a detailed introduction to particle filtering for assembly pose localization. It was shown how particle filters facilitate sequential processing of image sequences. Because no assumptions were made concerning the sequence length, all that has been said specifically holds for image sequences of length one. Based on a single monocular input image that observes the assembly under inspection, the presented SIR particle filter can therefore generate a sample set approximation of the posterior. The paragraph also explained the details of adapting SIR particle filtering to the task of assembly localization. The most important step was to define an application specific approximation of the observation density. The approximation is based on the combination

of different image cues. The resulting SIR particle filter is an important part of the kernel particle filter that is introduced next.

4.2.3 Foundations of Kernel Particle Filtering

A fair sample set representation of the posterior $p(\mathbf{x}_t|\mathcal{Y}_t)$ depends strongly on the dimensionality of the state space, which in our case is the assembly pose space. Clearly, the more DOF an assembly exhibits, the more particles are needed in order to cover the pose space and to identify subspaces of high density. In fact, after one iteration of the CONDENSATION algorithm from the previous paragraph, many particles will have nearly zero weight. We have already learned that this phenomenon is commonly known as the *degeneracy problem*. In order to determine, how many particles of a given set effectively contribute to the posterior approximation, the *effective sample size* was proposed (e.g. by Kong et al. [KLW94]). Its estimate, $\hat{N}_{\text{eff}}$, is obtained from

$$\hat{N}_{\text{eff}} = \left[\sum_{n=1}^{N_{\text{s}}} (w_t^n)^2 \right]^{-1}. \tag{4.13}$$

In order to identify critically degenerated particle sets, one can test whether $\hat{N}_{\text{eff}}$ falls below a task specific threshold T. Alternatively, one can try to estimate the *particle survival rate* α. The latter expresses the number of particles that are expected to survive a resampling step such as the one in line 2 of Alg. 1. For particle sets with large N_{s}, one can approximate the survival rate as [MI00]

$$\alpha \approx \frac{\hat{N}_{\text{eff}}}{N_{\text{s}}}, \text{ with } 0 \leq \alpha \leq 1. \tag{4.14}$$

Clearly, the smaller the survival rate α is, the more particles are needed in order to retain a minimal number of effective samples after a particle filtering iteration. More precisely, MacCormick & Isard [MI00] estimated that approaches like SIR particle filtering demand a particle set size N_{s} of at least

$$N_{\text{s}} \geq \frac{T}{\alpha^d} \tag{4.15}$$

particles, where T is the smallest acceptable efficient sample size and d is the dimension of the state space.

Many different approaches have been proposed in order to increase the particle survival rate α and thus alleviate the problem of excessive particle set sizes when facing high-dimensional state spaces. Within the application domain of visual inspection, we can

exploit the fact that we don't depend on representing the posterior pdf $p(\mathbf{x}_t|\mathcal{Y}_t)$ equally well over the full state space. Instead, we are much more interested in a compact representation of its local modes. The strongest such mode yields a MAP estimate of the assembly pose that gave rise to the observed image. In contrast to this, posterior regions of low density carry no information that is relevant for our task. The general idea of kernel particle filtering hence consists of moving the particles towards local modes of the posterior pdf.

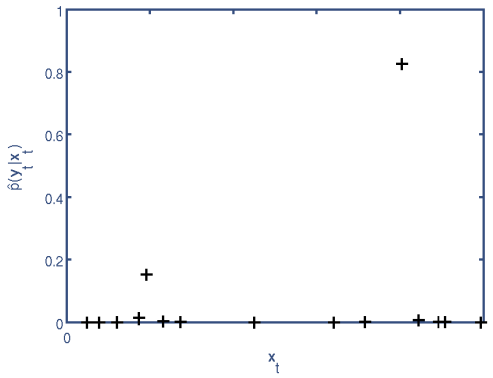
It is important to note that the central idea of kernel particle filtering was independently developed and published by Chang & Ansari [CA03, CA05] and Schmidt et al. [SKF06]. The following discussion adopts large parts of the approach in [CA05]. However, the presented contents differ in two important respects. First, a CONDENSATION step is used for the initialization. This stabilizes the approach by providing a coarse estimate of the posterior that can afterwards be improved with the mean shift iteration. At the same time, it reduces the computational load of the reweighting steps that are performed after each iteration of mean shift. Second, this thesis contributes a thorough grounding of the KPF algorithm to the concepts of kernel density estimation, mean shift and particle filtering. Above all, this allows to better understand why the KPF algorithm has been reported to improve the performance of SIR particle filters significantly.

The general concept of the kernel particle filtering procedure is illustrated in Fig. 4.6. The first subfigure shows the particle representation of a posterior pdf that was created by performing one iteration of CONDENSATION. It can be seen that the particles $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$ populate the state space sparsely. Nevertheless, they suffice to obtain a first estimate of the posterior by means of some procedure that will be discussed in the next paragraphs. Subfigure 4.6(b) shows a kernel density estimation outcome. The particles are then shifted towards the local peaks within the estimated density as depicted in Fig. 4.6(c).

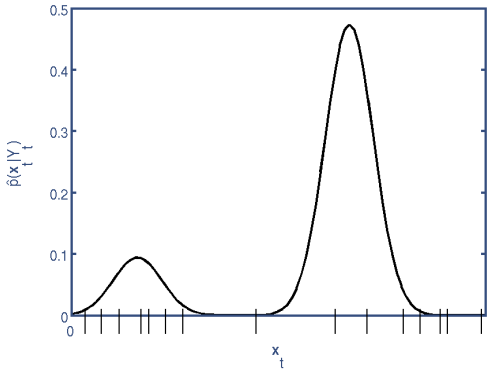
The shift operation leaves the particles scattered near local density modes. As a result, a more precise estimate of the density near the local modes can be obtained. One might consequently update the particle weights and repeat the density estimation and particle shifting steps of Fig. 4.6(b) and 4.6(c) for several times. In this way, the particles are gradually concentrated at the local modes of the estimated density. Afterwards, a new image measurement can be processed by performing a new CONDENSATION step that uses the shifted and reweighted particles as prior, and so forth. Both the density estimation and the particle shifting process will be introduced formally in the following.

Non-Parametric Density Estimation

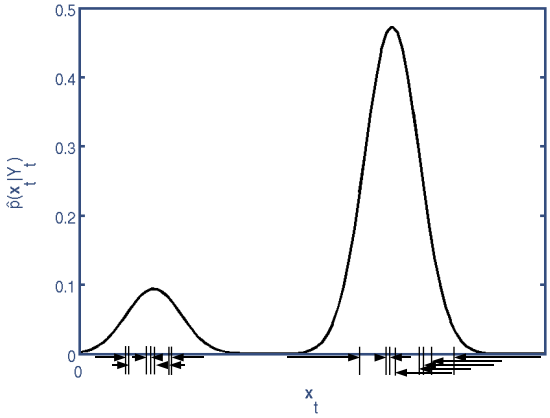
The previous paragraph showed us that kernel particle filtering involves the estimation of the density underlying the particle set $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$. This might seem a waste of time,



(a) Initial particles



(b) Kernel density estimate of the posterior



(c) Induced shifts

Figure 4.6: Kernel particle filtering. a) Initial particle set after CONDENSATION in a 1D example state space. The y-axis denotes the particle weight. b) Kernel density estimate of the associated posterior density. The particle positions $\{\mathbf{s}_t^n\}_{n=1}^{N_\mathrm{s}}$ are indicated as black vertical lines. c) Arrows denote the shift induced by one iteration of mean shift. Black vertical lines denote the new particle positions $\{\mathbf{s}_t^{n*}\}_{n=1}^{N_\mathrm{s}}$

given that this density is exactly the posterior that is already being approximated by the particle set according to Eqn. (4.5). However, the latter equation only holds for N_s that are large enough to permit a dense sampling of the posterior pdf. The reason for this is that the particle set approximation to the posterior pdf is fair at sample positions but possibly bad in between them. Given a rather sparsely sampled state space, the application of additional density estimation techniques therefore still has the potential to yield a more robust density estimate.

Principally, there are two classes of density estimation techniques. *Parametric* methods assume that a specific model function is known and aim at finding the set of parameters that best fit the model to the observations. Because we don't have such a model, we can dismiss the idea of employing parametric density estimation. The second broad class consists of *nonparametric* techniques which don't rely on knowing a specific model. For example, histograms are a widely used nonparametric density estimation approach. In the following, we focus on employing *kernel density estimation*. In the pattern recognition literature, this approach is also known as the *Parzen windowing* technique (e.g. [DH73]).

The *kernel density estimate* (KDE) $\hat{f}_K(\mathbf{x})$ of an unknown probability density function $f(\mathbf{x})$ is obtained from a sample $\{\mathbf{x}^n \in \mathbb{R}^d\}_{n=1}^{N_\mathrm{x}}$ of the unknown density. The KDE is computed by determining the distance between $\mathbf{x}$ and the N_x data points $\mathbf{x}^n$, rating the distances with a *kernel* K, and averaging the outcomes. Formally, this is expressed as

$$\hat{f}_K(\mathbf{x}) = \frac{1}{N_\mathrm{x} b^d} \sum_{n=1}^{N_\mathrm{x}} K\left(\frac{\mathbf{x} - \mathbf{x}^n}{b}\right), \tag{4.16}$$

where b is the *kernel bandwidth* that parameterizes the width of the kernel K. In the following, K is taken to be a radially symmetric, non-negative function that is centered at 0 and integrates to 1 [CM99]. Note that one could also employ a bandwidth matrix instead of scalar b in order to parameterize the estimator. However, we refrain from doing so because this would make the parameter value choice even more complicated than it already is for one bandwidth parameter. This topic is considered in more detail in Chap. 4.2.5.

The above estimator determines the density of the unweighted data points $\mathbf{x}^n$. Consequently, each data point contributes equally strong to the kernel evaluations. In case of weighted samples, the estimator can be extended in order to account for the weight information [Gis03]. In the latter form, kernel density estimation is directly applicable to a particle set $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$ in order to obtain a robust estimate of the underlying posterior pdf. If the particle weights were chosen according to (4.8) and (4.12) and afterwards normalized such that they sum up to 1, the posterior can be estimated as

$$\hat{p}(\mathbf{x}_t|\mathcal{Y}_t) = \frac{1}{b^d} \sum_{n=1}^{N_\mathrm{s}} K\left(\frac{\mathbf{x}_t - \mathbf{s}_t^n}{b}\right) w_t^n. \tag{4.17}$$

So far, it hasn't been specified what kernel K we intend to use. This choice is application specific. In the following, we employ the *Epanechnikov kernel* which has a parabolic shape. Besides being optimal with respect to minimizing the mean integrated square error (MISE), this kernel has the advantage of simplifying some of the later following equations. Its multi-variate and radially-symmetric version is defined as

$$K(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-\|\mathbf{x}\|^2) & \text{if } \|\mathbf{x}\| \leq 1 \\ 0 & \textit{otherwise}, \end{cases} \tag{4.18}$$

where c_d is a normalization constant that depends on the dimensionality d of the state space. Furthermore, $\|\cdot\|$ denotes the Euclidean norm.

In summary, we have learned how the KDE of a posterior pdf can be obtained. The next important topic to discuss is how the modes of such a density estimate can be efficiently found and the particles shifted towards it.

Mean Shift Based Mode Localization

An efficient technique for the iterative localization of modes within a KDE is offered by the mean shift algorithm. It was proposed by Fukunaga & Hostetler [FH75] and has since then been used in many different applications. In the following, the mean shift algorithm is presented briefly. A fine-grained discussion is provided by Comaniciu & Meer [CM99]. Their formalization has been adopted here to a large degree. One major difference, however, lies in the form of the employed kernel density estimator. While the literature usually refers to the form given in Eqn. (4.16), an estimator of form (4.17) is used here. Comaniciu & Meer [CM02] stress that the convergence properties of the mean shift algorithm remain unaltered in such a setting. Thus, mean shift is directly applicable to particle sets, too.

The mean shift algorithm is based on estimating the gradient of a density at point $\mathbf{x}_t \in \mathbb{R}^d$. Let $\hat{f}_K(\mathbf{x}_t)$ be the kernel density estimate with kernel K of a pdf that is represented by the particle set $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$. Furthermore, let K be a differentiable kernel. Then, the gradient $\nabla\hat{f}_K(\mathbf{x}_t)$ is given by

$$\nabla\hat{f}_K(\mathbf{x}_t) = \frac{1}{b^d}\sum_{n=1}^{N_\mathrm{s}} \nabla K\left(\frac{\mathbf{x}_t - \mathbf{s}_t^n}{b}\right) w_t^n. \tag{4.19}$$

In order to find zero-gradient points such as density modes, the mean shift algorithm proceeds as follows. First, the local density gradient is measured at an arbitrary point $\mathbf{x}_t$. The point is then shifted along the direction of the estimated density gradient. This procedure is repeated until the position of $\mathbf{x}_t$ converges to a point of zero gradient. In order to find all modes of a density function, each iteration evaluates and shifts a large number of such points that are distributed over the whole search space.

The mean shift algorithm obviously performs gradient ascent. As with all gradient ascent methods, the choice of the step size is crucial, if convergence to density modes must be

guaranteed. The mean shift algorithm automatically adapts its step size by pursuing the *normalized gradient*

$$m_b(\mathbf{x}_t) = c_b \frac{\nabla \hat{f}_K(\mathbf{x}_t)}{\hat{f}_G(\mathbf{x}_t)}, \tag{4.20}$$

where c_b is a normalization constant and $m_b(\mathbf{x}_t)$ is called the *sample mean shift*, both depending on bandwidth b. Furthermore, G is a kernel that is related to kernel K by $G(\mathbf{x}) = c_{G,K} K'(\mathbf{x})$, where $c_{G,K}$ is a kernel-dependent normalization constant. The normalized gradient effects that for regions of small density, the algorithm makes large steps along the density gradient, and vice versa.

Comaniciu & Meer [CM99] prove that this adaptation scheme guarantees convergence to points of zero density gradient. By using an Epanechnikov kernel and writing out the normalized gradient term, they furthermore show that

$$m_b(\mathbf{x}_t) = c_b \frac{\nabla \hat{f}_K(\mathbf{x}_t)}{\hat{f}_G(\mathbf{x}_t)} = \frac{1}{w_S} \sum_{\mathbf{s_t^n} \in S_b(\mathbf{x}_t)} \mathbf{s}_t^n w_t^n - \mathbf{x}_t. \tag{4.21}$$

Here, $S_b(\mathbf{x}_t)$ denotes a hyper-sphere of radius b that is centered on $\mathbf{x}_t$ and encloses particles with a total accumulated weight of $w_S \leq 1$. Equation (4.21) specifies that the sample mean shift at position $\mathbf{x}_t$ can be computed from the weighted average of all samples that fall into a sphere centered on $\mathbf{x}_t$. The appealing property of this expression is that it determines the sample mean shift without explicitly calculating the gradient estimate $\nabla \hat{f}_K(\mathbf{x}_t)$, which would be much more sensitive to noise.

As stated before, the mean shift algorithm repeatedly translates a number of evaluation points $\mathbf{x}_t$ by their sample mean shift $m_b(\mathbf{x}_t)$. Throughout this process, the sample positions $\mathbf{s}_t^n$ remain fixed. In contrast to this, the kernel particle filtering approach employs the mean shift operation directly on the particle positions $\{\mathbf{s}_t^n\}_{n=1}^{N_\mathrm{s}}$ as shown in Fig. 4.6. This is achieved by evaluating $m_b(\mathbf{x}_t = \mathbf{s}_t^n)$ for all $n = 1, \ldots, N_\mathrm{s}$ and then translating the particles by their respective sample mean shift. Doing so herds the particles towards local density modes of the posterior pdf estimate. In the following, the shifted particle positions are denoted as $\{\mathbf{s}_t^{n*}\}_{n=1}^{N_\mathrm{s}}$.

Now that we have learned how the mean shift algorithm can be applied to particles, a more detailed discussion of kernel particle filtering is possible. As illustrated in Fig. 4.6, kernel particle filtering is initialized with a CONDENSATION step that incorporates the current image measurement into the particle representation of the estimated posterior pdf. The resulting particles $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$ are then shifted to local modes of the estimated posterior by means of a mean shift step, yielding the new particle positions $\{\mathbf{s}_t^{n*}\}_{n=1}^{N_\mathrm{s}}$.

The shift moves the particles to interesting regions of the state space, namely regions of high estimated posterior density. But from a theoretical point of view, it also breaks the representation of the posterior because the weights of the $\{\mathbf{s}_t^{n*}\}_{n=1}^{N_\mathrm{s}}$ don't correspond to the new particle positions any more. The crucial question therefore is, how the weights must be updated in order to maintain a valid particle representation.

This question can be answered by defining an appropriate proposal distribution $q(\mathbf{x}_t^n|\mathbf{x}_{t-1}^n, \mathbf{y}_t)$. In the context of SIR particle filtering, we have learned that it is chosen according to (4.7), i.e. it is defined as the system dynamics model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Because the KPF is initialized with an SIR particle filter, its proposal distribution therefore subsumes $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ as well. After sampling from this distribution, however, the particles are shifted along their individual sample mean. The resulting distribution is expressed as $q_\mathrm{ms}(\mathbf{x}_t)$ in the following. Note that no functional representation of it is available. Instead, $q_\mathrm{ms}(\mathbf{x}_t)$ is *constructed* by performing mean shift steps on the particles of time step t. In order to express that $q_\mathrm{ms}(\mathbf{x}_t)$ is applied to particles sampled from $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, we define the KPF proposal distribution as

$$q(\mathbf{x}_t^n|\mathbf{x}_{t-1}^n, \mathbf{y}_t) = q_\mathrm{ms}(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1}). \tag{4.22}$$

By definition, sampling from $q(\mathbf{x}_t^n|\mathbf{x}_{t-1}^n, \mathbf{y}_t)$ means to shift the particles $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$ to the new positions $\{\mathbf{s}_t^{n*}\}_{n=1}^{N_\mathrm{s}}$. Plugging this into the weight update scheme (4.6) yields

$$w_t^{n*} \propto w_{t-1}^n \frac{p(\mathbf{y}_t|\mathbf{s}_t^{n*})p(\mathbf{s}_t^n|\mathbf{s}_{t-1}^n)}{q_\mathrm{ms}(\mathbf{s}_t^{n*})p(\mathbf{s}_t^n|\mathbf{s}_{t-1}^n)} = w_{t-1}^n \frac{p(\mathbf{y}_t|\mathbf{s}_t^{n*})}{q_\mathrm{ms}(\mathbf{s}_t^{n*})}. \tag{4.23}$$

Due to the resampling step that is inherent to CONDENSATION, we can again assume that w_{t-1}^n takes on uniform values, which simplifies the KPF weight update scheme to

$$w_t^{n*} = \frac{p(\mathbf{y}_t|\mathbf{s}_t^{n*})}{q_\mathrm{ms}(\mathbf{s}_t^{n*})}, \text{ for all } n = 1, \ldots, N_\mathrm{s}. \tag{4.24}$$

In order to evaluate $q_\mathrm{ms}(\cdot)$ at particle position $\mathbf{s}_t^{n*}$, we simply obtain the unweighted KDE over the set of shifted samples. This yields the estimate

$$\hat{q}_\mathrm{ms}(\mathbf{s}_t^{n*}) = \frac{1}{N_\mathrm{s} b^d} \sum_{l=1}^{N_\mathrm{s}} K\left(\frac{\mathbf{s}_t^{n*} - \mathbf{s}_t^{l*}}{b}\right). \tag{4.25}$$

It is important to note that the weight update scheme from Rashid & Ansari [CA03, CA05] is considerably more expensive in terms of induced computational load than (4.24). The reason for this is that, for each iteration of mean shift, Rashid & Ansari

Algorithm 2 Kernel Particle Filter (KPF)

Input: Particles $\{\mathbf{s}_{t-1}^n, w_{t-1}^n\}_{n=1}^{N_\mathrm{s}}$, image $\mathbf{y}_t$, bandwidth b_0, decrease factor $\lambda \in]0,1[$

// For single images, let $t = 1$ and initialize $\{\mathbf{s}_{t-1}^n, w_{t-1}^n\}_{n=1}^{N_\mathrm{s}}$ from prior $p(\mathbf{x}_0)$

1: Create particles $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}} \leftarrow \texttt{condensation}(\{\mathbf{s}_{t-1}^n, w_{t-1}^n\}_{n=1}^{N_\mathrm{s}}, \mathbf{y}_t)$.

2: **For all** $i = 1 : I$ **do**

3: Compute whitening matrix A from empirical sample covariance matrix $\hat{C}$.

4: Perform whitening $\{\mathbf{s}_{t,\mathrm{v}}^n\}_{n=1}^{N_\mathrm{s}} \leftarrow \{A(\mathbf{s}_t^n - \bar{\mathbf{s}}_t)\}_{n=1}^{N_\mathrm{s}}$ with $\bar{\mathbf{s}}_t = \sum_{n=1}^{N_\mathrm{s}} \mathbf{s}_t^n$.

5: Shift particles according to $\{\mathbf{s}_{t,\mathrm{v}}^{n*}\}_{n=1}^{N_\mathrm{s}} \leftarrow \texttt{mean shift}(\{\mathbf{s}_{t,\mathrm{v}}^n, w_t^n\}_{n=1}^{N_\mathrm{s}}, b = b_0\lambda^i)$.

6: Perturb and transform $\{\mathbf{s}_t^{n*}\}_{n=1}^{N_\mathrm{s}} \leftarrow \{A^{-1}\mathbf{s}_{t,\mathrm{v}}^{n*} + \bar{\mathbf{s}}_t + bA^{-1}\mathbf{e}\}_{n=1}^{N_\mathrm{s}}$ with $\mathbf{e} \sim \mathcal{N}(0, \mathbf{I}_d)$.

7: Update weights $\{w_t^{n*}\}_{n=1}^{N_\mathrm{s}} \leftarrow \{\frac{\hat{p}(\mathbf{y}_t|\mathbf{s}_t^{n*})}{\hat{q}_\mathrm{ms}(\mathbf{s}_t^{n*})}\}$.

8: Normalize $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}} \leftarrow \{\mathbf{s}_t^{n*}, \frac{w_t^{n*}}{\sum_{n=1}^{N_\mathrm{s}} w_t^{n*}}\}_{n=1}^{N_\mathrm{s}}$

9: **End For**

Output: Particles $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$ // For single images: stop here

apply the system model (4.1) at time step $t-1$ to the complete particle set in order to maintain an estimate of the posterior pdf. In our approach, this propagation step is carried out *before* the mean shift iterations, by means of CONDENSATION. Doing so guarantees a valid particle representation of the estimated posterior while being computationally cheaper.

In summary, performing a mean shift operation on a particle set that was initialized with CONDENSATION yields a consistent representation of the posterior pdf, if the weight update scheme from (4.24) is used. In such a case, kernel particle filtering can be categorized as mean shift guided CONDENSATION. Note that, in contrast to many other particle filtering approaches that attempt to steer particles to interesting regions of the state space, kernel particle filtering uses a deterministic approach to translate the particles. The good performance characteristics of the KPF algorithm result mainly from the fast convergence of the mean shift steps.

The basic kernel particle filtering steps are summarized in Alg. 2 which is discussed in the following. Line 1 expresses that a CONDENSATION step is used to propagate the particle set representation of the posterior from time step $t-1$ to t. Afterwards, I iterations of mean shift are performed on the particle set. In each iteration, a standard whitening transform matrix A is obtained by an eigenvalue decomposition of the empirical covariance matrix $\hat{C}$ of the samples $\{\mathbf{s}_t^n\}_{n=1}^{N_\mathrm{s}}$. The samples then undergo a variance normalization by subtracting their mean $\bar{\mathbf{s}}_t$ and applying the whitening transform A as indicated in line 4. This step rescales the sample space. After rescaling, the d vector components of the samples $\{\mathbf{s}_{t,\mathrm{v}}^n\}_{n=1}^{N_\mathrm{s}}$ are decorrelated and of unit variance. This is important because the untransformed samples contain translation and rotation parameters that vary on different scales. Within the unwhitened space, a spherical kernel such as (4.18) would tend to oversmooth parameters with small variance while it would tend to undersmooth parame-

ters with large variance. After whitening, the particles undergo a mean shift as indicated in line 5. Note that the bandwidth parameter b is decreased by $0 < \lambda < 1$ at each of the I iterations. The decrease aims at changing the smoothing behavior of kernel K. We will discuss this heuristic in Chap. 4.2.5 and present an alternative and theoretically more promising approach. Line 6 of the algorithm transforms the shifted samples back to their original parameter space. Additionally, the particle positions are perturbed by some small random noise, in order to push them away from local density plateaus. Afterwards, the particle weights are updated according to Eqn. (4.24) and normalized such that they sum up to 1.

Formally, kernel particle filtering is strongly related to SIR particle filtering. In fact, we have seen that kernel particle filtering is mean shift guided CONDENSATION. The fast conversion of the mean shift to local modes of the posterior estimate explains why, compared to SIR particle filtering, the KPF algorithm successfully increases the particle survival rate α from (4.14). However, Chang & Ansari restrict the evaluation of their algorithm to state spaces of dimension $3 \leq d \leq 9$. And in our application context, kernel particle filtering still needs intractably many particles to localize even simple assemblies with competitive accuracy and precision. This thesis therefore contributes several extensions to Alg. 2. The extended kernel particle filter (EKPF) facilitates the computationally tractable localization of multi-part assemblies with competitive accuracy and precision.

4.2.4 Weighting Function Manipulation

The first KPF extension that is provided by this thesis is aimed at further increasing the particle survival rate α from (4.14). In order to achieve this, an extension is incorporated to the KPF framework that addresses two fundamental problems of mean shift based mode detection under sparse sampling conditions. The posterior pdf estimate that is illustrated in Fig. 4.7(a) provides an instructive example to explain these problems.

Figure 4.7(a) depicts a typical posterior pdf with narrow peaks and regions of almost zero density. It also shows the sample positions of a typical particle set that populates the state space sparsely. Consider the highlighted particle at the figure's center whose kernel range is indicated above it. Like the surrounding particles, it is positioned in a region of almost zero posterior density. A mean shift step doesn't affect this particle because there is no density gradient within its kernel range. It is trapped in a *density plateau*. Thus, the particle can't contribute to locating local modes of the posterior density. And the more particles get stuck in plateaus of low density, the lower the resulting particle survival rate will be. This problem is alleviated somewhat by the random noise that the KPF algorithm adds to particles (cf. line 6 of Alg. 2). However, the noise must be kept at a small level, in order not to induce too much variance to the estimation results.

A second problem that can be understood from Fig. 4.7(a) arises from the fact that sparse sampling from the state space can lead to an undersampling of narrow density peaks.

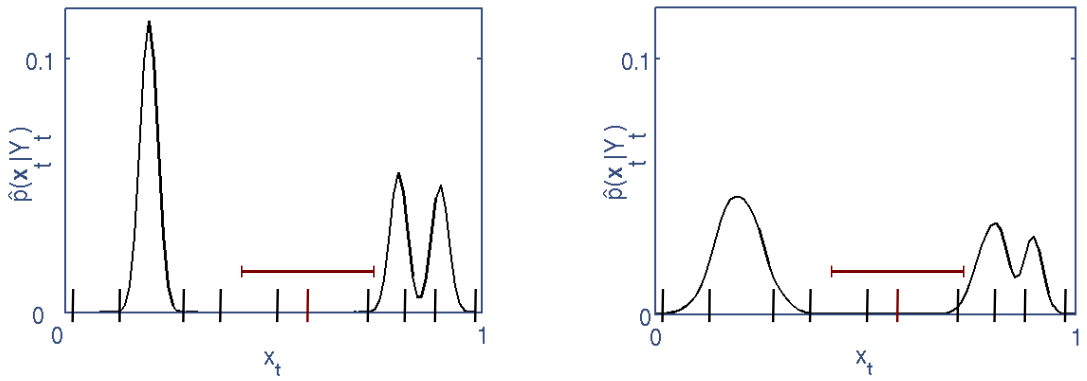


Figure 4.7: The effects of weighting function manipulation. a) Typical posterior pdfs exhibit narrow peaks and regions of near zero density. The vertical bars denote particle positions (a 1D state space was chosen for illustrative purposes). For the particle highlighted in red, an exemplary kernel range is indicated. A mean shift with this bandwidth wouldn't move the highlighted particle because no density gradient is detectable within its kernel range. b) The posterior after weighting function manipulation, with the same set of particles. A mean shift step would now shift the highlighted particle towards the nearest density mode

Such peaks occur because the standard deviations σ_c of the Gaussian cue weighting functions (4.10) are typically chosen close to zero. As a result, very narrow peaks are induced to the observation density, which in turn leads to narrow peaks in the estimated posterior. An illustrative example is given with the highest peak at the figure's left side. Note that the particles completely miss the peak. Therefore, a mean shift step doesn't detect density gradients towards it.

For assemblies with many DOF, sparse sampling is inevitable because the associated particle state space is extremely large. The question thus is how to deal with the problem of narrow peaks and density plateaus. Chang & Ansari [CA03] proceed by using a large initial kernel bandwidth b_0. The larger b_0 is, the more likely particles of non-zero density fall into the kernel range and draw other particles away from plateaus. However, the problem of missed narrow peaks remains. Also, the posterior pdf is now grossly oversmoothed. This flattens out peaks and blends them together. As a result, the KDE variance is decreased and the oversmoothed density yields a coarse indication of state space regions with density peaks. On the other hand side, oversmoothing increases the KDE bias. Chang & Ansari thus decrease the kernel bandwidth after each iteration of mean shift by a constant factor.

In this thesis, it was decided to decouple the bandwidth selection from the problems of dealing with density plateaus and narrow peaks. The solution to the former problem is

discussed in Chap. 4.2.5. Regarding the latter problem, the heuristic that is used in the following manipulates the Gaussian cue weighting functions from (4.10). The manipulation reshapes the peaks of the estimated observation density from (4.12) such that they are widened. The estimated posterior is affected likewise which is illustrated in Fig. 4.7(b). Here, the left-most peak is no longer missed. What is more, when applying the depicted example bandwidth, a non-zero posterior gradient can be determined for all particles. Thus, a mean shift step now moves more particles towards the nearest local modes of the estimated posterior density.

Our approach is related to the *annealed particle filtering* concept of Deutscher and colleagues [DBR00]. In their setting, estimates $\hat{p}(\mathbf{y}_t|\mathbf{s}_t^n)$ of the observation density are obtained by evaluating a weighting function $w(\mathbf{y}_t, \mathbf{s}_t^n)$ that arises from the evaluation of simple image features and yields values in the range of $[0, 1]$. In order to successively reshape the peaks of $w(\mathbf{y}_t, \mathbf{s}_t^n)$, the authors exponentiate the weighting function by defining $\hat{p}(\mathbf{y}_t|\mathbf{s}_t^n) = w(\mathbf{y}_t, \mathbf{s}_t^n)^\beta$. Starting with a value of β close to zero, the authors perform up to 10 iterations of CONDENSATION. For each iteration, β is increased which gradually narrows down the peaks of the estimated observation density and the associated estimate of the posterior.

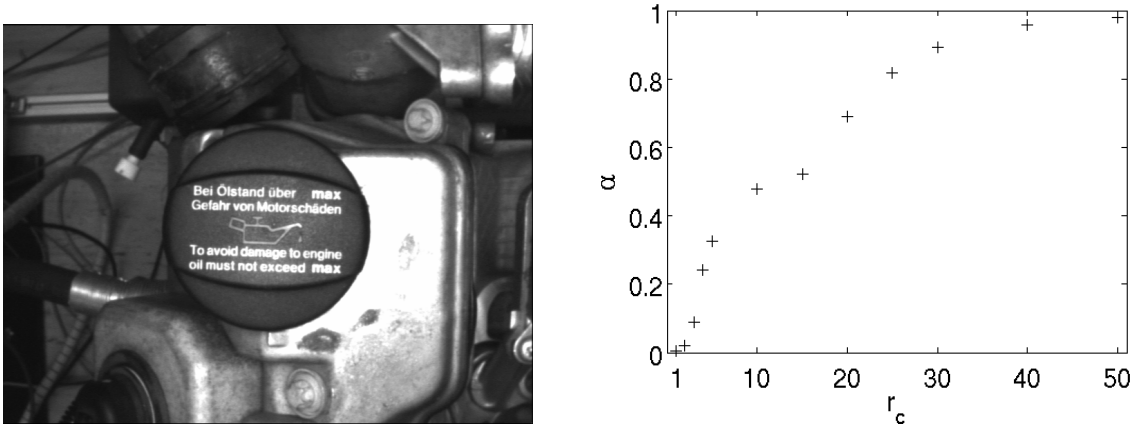
In our case, the observation density is estimated by combining multiple cues as defined in (4.12) and (4.10), i.e.

$$\hat{p}(\mathbf{y}_t|\mathbf{s}_t^n) = N_{\text{cues}}^{-1} \prod_{k=1}^{j} \sum_{c\in\{\text{fw,bw,col}\}} \exp\left(-\frac{(f_c(\mathbf{z}_t^k, \mathbf{y}_t))^2}{2\sigma_c^2}\right), \tag{4.26}$$

where N_{cues} is the number of cues and $f_c(\mathbf{z}_t^k, \mathbf{y}_t)$ denotes a cue response that is assumed to be normally distributed. As indicated above, the cue specific standard deviation σ_c is typically chosen to be close to zero. In other words, each of the summed cue likelihoods is very sensitive to small deviations of the cue response $f_c(\mathbf{z}_t^k, \mathbf{y}_t)$ from its optimum at 1. By artificially increasing σ_c, the sensitivity can be lowered in a controlled fashion. Noting this, we enhance σ_c in (4.26) with a rescale factor $r_c \geq 1$ that is varied for each of the $i = 1 \ldots I$ iterations of mean shift. From this we obtain

$$\hat{p}(\mathbf{y}_t|\mathbf{s}_t^n) = N_{\text{cues}}^{-1} \prod_{k=1}^{j} \sum_{c\in\{\text{fw,bw,col}\}} \exp\left(-\frac{(f_c(\mathbf{z}_t^k, \mathbf{y}_t))^2}{2(r_c\sigma_c)^2}\right). \tag{4.27}$$

The KPF Algorithm is altered in the following way. For the first iteration $i = 1$ of mean shift, the r_c are assigned large values. Consequently, $\hat{p}(\mathbf{y}_t|\mathbf{s}_t^n)$ exhibits comparatively broad peaks for the first application of mean shift. For the following iterations, the r_c are decreased by a constant factor. Furthermore, the initial value of the r_c is chosen such



(a) (b)

Figure 4.8: The effects of weighting function manipulation. a) An example image of an oil cap (courtesy of DaimlerChrysler AG). b) The posterior density of an oil cap model and the example image is estimated with one iteration of CONDENSATION, using the observation density estimate from (4.27) and varying scale factors r_c. The survival rate α of the resulting particle set is plotted against the scale factors r_c. The survival rate clearly increases with increasing values of r_c

that they converge to 1 at the final iteration $i = I$ of mean shift. At this final iteration of mean shift, the observation density estimate is thus determined from the original equation (4.26).

In contrast to Deutscher et al., our approach has the advantage that we can control the sensitivity of each cue independently. Another major difference between our KPF and their particle filter is the fact that we don't need to perform intermediate CONDENSATION iterations on the particle sets in order to migrate particles to local modes in the posterior. Our mean shift approach allows to do this much more efficiently. Typically, 2 or 3 iterations of mean shift are sufficient to locate the posterior modes while, in [DBR00], 10 annealing iterations are proposed.

The concept of weighting function manipulation is a heuristic that leads to initially broadly peaked estimates of the observation density and the posterior. The effect of this manipulation on the particle survival rate α is exercised in Fig. 4.8. It can be seen that the particle survival rate increases with increasing scale factors r_c. However, with increasingly large scale factors, the bias of the estimated posterior grows rapidly. The scale factors r_c are therefore forced to converge to 1 at the final iteration of mean shift, as stated above, in order to translate the particles gradually to the true modes of the estimated posterior pdf. The heuristic is evaluated in the second experimental investigation of the following chapter. The evaluation yields empirical evidence that the heuristic suc-

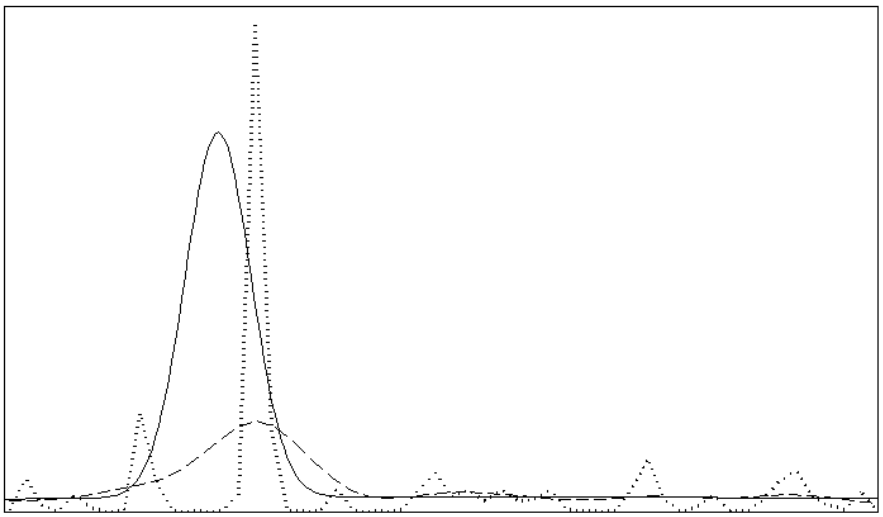


Figure 4.9: Antagonal driving forces of bandwidth selection. The true density (solid line) is estimated with a small (dotted line) and a large kernel bandwidth (dashed line). The small bandwidth yields a reasonable estimate of the density peak but induces strong false local modes at the sparsely sampled density tail. The large bandwidth yields better approximation of the tail but grossly oversmooths the peak

cessfully increases the particle survival rate and thus improves the assembly localization accuracy and precision.

4.2.5 Automatic Bandwidth Selection

The KPF algorithm from diagram 2 employs a global kernel bandwidth b_0 that is decreased in each iteration by a factor $0 < \lambda < 1$. Finding a suitable bandwidth is a very hard problem and in practice often includes a certain amount of guessing. The difficulties arise from two antagonal driving forces that are illustrated in Fig. 4.9. The figure shows that high density regions such as peaks are reconstructed best with a small kernel bandwidth which avoids oversmoothing. However, the figure also shows that small kernel bandwidths introduce strong variance in regions of low density such as the tail of the example density. For the tail, a large kernel bandwidth achieves superior results. Bandwidth selection techniques aim at finding a compromise.

Finding a compromise in the scenario described above is far from being trivial. What is more, in the original kernel particle filtering approach, it is complicated unnecessarily because bandwidth selection is mixed with the task of increasing the particle survival

rate. Within this thesis, it was decided to decouple the two problems. By this separation of concerns, both tasks can be solved more satisfactorily. Consequently, the following considerations focus entirely on performing a bandwidth selection that yields a fair continuous estimate of the posterior pdf.

When estimating multi-modal densities with peaks of varying strength and width, the above mentioned problems with selecting a global kernel bandwidth become notorious. In such cases, *fixed bandwidth* kernel density estimators as (4.16) and (4.17) perform rather poorly. An interesting alternative is offered by *adaptive kernel estimators* [Sil86] that are also known as *sample-point estimators* [BMP77]. When estimating the density represented by the particle set $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$, such estimators are of the form

$$\hat{f}_K(\mathbf{x}_t) = \sum_{n=1}^{N_\mathrm{s}} \frac{w_t^n}{(b\lambda_n(\mathbf{s}_t^n))^d} K\left(\frac{\mathbf{x}_t - \mathbf{s}_t^n}{b\lambda_n(\mathbf{s}_t^n)}\right). \tag{4.28}$$

Here, b is a global bandwidth parameter. However, this global bandwidth is now enhanced by the *local* bandwidth parameters $\lambda_n(\mathbf{s}_t^n)$. This means that each particle is associated with its individual local kernel bandwidth.

Regarding computer vision tasks, Comaniciu and colleagues [CRM01] have successfully employed sample-point estimators in the context of mean shift based image segmentation. However, the method hasn't been adapted to particle filtering yet, which is done in the following. Sample-point estimators have the advantage that the local bandwidths can be made small at points of high density and vice versa. Intuitively, this yields more accurate estimates of multi-modal densities because the bandwidths can be tuned locally to avoid over- and undersmoothing. A major disadvantage is that we now have to select N_s additional parameters. Silverman [Sil86] proposed the *adaptive kernel method* in order to solve this problem. It determines the local bandwidths as

$$\lambda_n(\mathbf{s}_t^n) = \left[g/\hat{f}_\mathrm{p}(\mathbf{s}_t^n)\right]^{\frac{1}{2}}, \text{ with} \tag{4.29}$$

$$g = \left[\prod_{n=1}^{N_\mathrm{s}} \hat{f}_\mathrm{p}(\mathbf{s}_t^n)\right]^{\frac{1}{N_\mathrm{s}}}. \tag{4.30}$$

In the above equations, $\hat{f}_\mathrm{p}(\mathbf{s}_t^n)$ is an initial local density estimate at $\mathbf{s}_t^n$, termed *pilot density*. Furthermore, g is the geometric mean of the pilot density.

The adaptive kernel method has the following effect. Suppose that a coarse pilot density $\hat{f}_\mathrm{p}(\mathbf{s}_t^n)$ is available. Then, this density estimate can be used in order to distinguish locally

high from locally low density values. The former are expected to be higher than the geometric mean g of the pilot density. The latter are expected to be lower than g. Note that the above equations guarantee $\hat{f}_{\mathrm{p}}(\mathbf{s}_t^n) < g \Rightarrow \lambda_n(\mathbf{s}_t^n) > 1$ and $\hat{f}_{\mathrm{p}}(\mathbf{s}_t^n) > g \Rightarrow \lambda_n(\mathbf{s}_t^n) < 1$. Thus, whenever the pilot density at a certain point drops below its geometric mean, (4.29) effects that the global bandwidth b is increased at that point. If the pilot density is above its geometric mean, the b is locally decreased.

At this point, it seems that we have extended the problem of selecting one global bandwidth with the burden of estimating a pilot density. Luckily, the literature indicates that the final estimate is rather insensitive to the specific details of the pilot density [Gis03]. For example, in [CRM01] a plugin-rule is used to obtain a fixed-bandwidth pilot density. The resulting final sample-point density estimate is empirically shown to be more accurate than a fixed-bandwidth estimator. Another possibility is to define $\lambda_n(\mathbf{s}_t^n)$ on the basis of the associated particle weight w_t^n

$$\lambda_n(\mathbf{s}_t^n) = (N_{\mathrm{s}} w_t^n)^{-\frac{1}{2}}. \tag{4.31}$$

According to Gisbert [Gis03], this yields a *bandwidth weighted* kernel density estimate. It effects that the global bandwidth b is increased at points where $w_t^n < \frac{1}{N_{\mathrm{s}}}$ and decreased whenever $w_t^n > \frac{1}{N_{\mathrm{s}}}$.

With either of the two approaches, sample-point estimation becomes applicable. All that remains to be done is to integrate a sample-point density estimate into our mean shift procedure. This can be done straight forward. We observe that the sample mean shift formulation in (4.21) can be interpreted as evaluating a uniform kernel with fixed bandwidth b that is centered at a specific sample $\mathbf{s}_t^r, 1 \leq \mathbf{s}_t^r \leq N_{\mathrm{s}}$ [CM99]. The fixed bandwidth determines the radius of the sphere $S_b(\mathbf{s}_t^r)$ by which the $\mathbf{s}_t^n \in S_b(\mathbf{s}_t^r)$ for the mean computation are selected. In order to use a variable bandwidth kernel within the mean shift procedure, all we have to do is to exchange the fixed-radius sphere $S_b(\mathbf{s}_t^r)$ with spheres of variable radius $b\lambda_n$ that are centered around the elements of comparison $\mathbf{s_t^n}$. Formally, the *variable bandwidth sample mean shift* $m_{b\lambda_n}(\mathbf{s}_t^r)$ is now written as

$$m_{b\lambda_n}(\mathbf{s}_t^r) = \frac{1}{w_S} \sum_{\{\mathbf{s_t^n} | \mathbf{s_t^r} \in S_{b\lambda_n}(\mathbf{s}_t^n)\}} \mathbf{s}_t^n w_t^n - \mathbf{s}_t^r, \tag{4.32}$$

where w_S again denotes the total accumulated weight of all samples $\mathbf{s}_t^n$ in the set $\{\mathbf{s_t^n} | \mathbf{s_t^r} \in S_{b\lambda_n}(\mathbf{s}_t^n)\}$.

Depending on the underlying density, this approach employs a large local bandwidth parameter, if $\mathbf{s}_t^r$ lies in a low-density region and vice versa. The global bandwidth parameter b must still be chosen somehow. In our application, this is done manually. However, because b is adapted to local fluctuations of the posterior density, the resulting estimate is

expected to be less sensitive against a particular choice of b. This expectation agrees with the finding that b could be held constant within the individual experimental investigations from the following chapter. In contrast to this, an EKPF with conventional mean shift had to be re-parameterized repeatedly in order to yield comparable performance. What is more, the second experimental investigation of the following chapter yields empirical evidence that the adaptive bandwidth selection improves the assembly localization and accuracy in comparison to the original KPF.

4.2.6 Dynamic State Space Decomposition

The extensions that have been contributed so far were all aimed at increasing the particle survival rate α of the EKPF, or at reducing the estimation bias and variance. However, the approach still doesn't scale to a high-dimensional state space. The basic problem is that the bias of kernel density estimates increases with a growing number of state space dimensions d, assuming that the bandwidth is enlarged in order to keep the variance of the estimates constant. The only possible compensation would be to increase the number of particles but this soon becomes computationally intractable. In the light of these considerations, it isn't surprising that Chang & Ansari [CA03] test the performance of their KPF in visual tracking scenarios with maximum 9 dimensional state spaces. But when localizing multi-part assemblies, each of the j parts contributes six dimensions to the state space, yielding $d = 6j$ dimensions. These considerations suggest that kernel particle filtering is tractable only for assemblies composed from very few parts.

The scalability problem is addressed in the following by proposing a final EKPF extension. It decomposes the state space into subspaces of tractable size which are then filtered individually. By means of this extension, the EKPF can localize assemblies composed from multiple components, at the prize of reduced robustness against inter-part occlusion. The extension is related to the *search space decomposition* approach of Gavrila & Davis [GD96] who use an hypothesize-and-test procedure for the model-based tracking of humans from multiple views. In order to track a 3D contour model with 22 DOF, they subdivide the search space into three disjunctive subspaces of 5, 9, and 8 dimensions. Each subspace is then searched individually by keeping the remaining parameters fixed at some predicted value. However, the authors employ deterministic best first search over a uniformly discretized search space. It is therefore discussed in the following, how a state space decomposition concept can be realized in the context of kernel particle filtering.

Let the estimated posterior of time step $t-1$ be represented by a particle set $\{\mathbf{s}_{t-1}^n, w_{t-1}^n\}_{n=1}^{N_\mathrm{s}}$. Then, this thesis proposes to perform state space decomposition in the context of kernel particle filtering as illustrated in Fig. 4.10. First, the particle samples $S_{t-1} = \{\mathbf{s}_{t-1}^n\}_{n=1}^{N_\mathrm{s}}$ are orthogonally projected along the assembly pose space $\mathcal{C}_\mathcal{A}$ onto

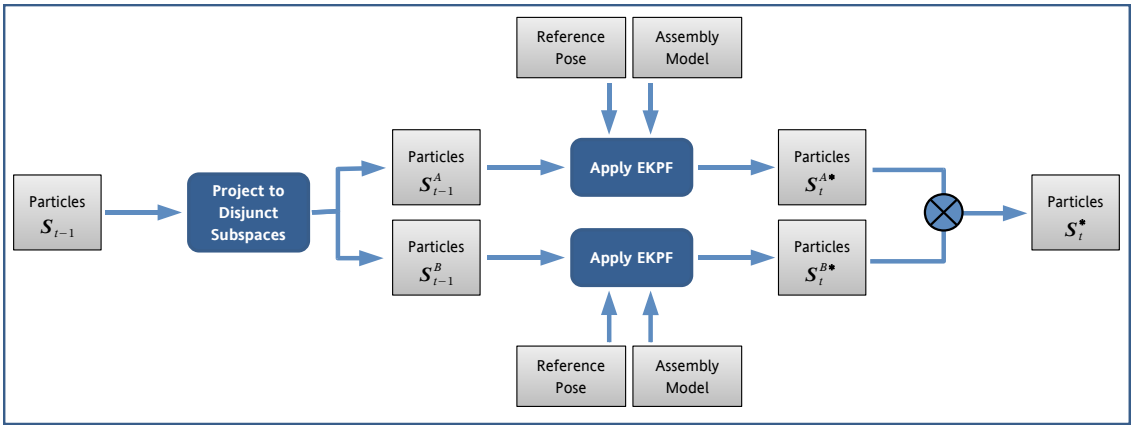


Figure 4.10: Kernel particle filtering on a decomposed state space

disjunctive subspaces A and B. This operation results in two sets S^A_{t-1} and S^B_{t-1} of projected samples. The samples in S^A_{t-1} are then updated to S^A_t and reweighted by means of a CONDENSATION step. Afterwards, mean shift iterations are performed that migrate the elements of S^A_t towards the gradient of the posterior confined to subspace A, resulting in S^{A*}_t. The same scheme is performed for S^B_{t-1} which produces S^{B*}_t. Finally, the positions of the k strongest modes within the obtained particle sets can be combined as cartesian product and reweighted. This results in a particle set that carries k^2 pose hypotheses.

The decomposition scheme seems to be simple but it provides a number of substantial caveats. For example, CONDENSATION is carried out in the individual subspaces A and B of the assembly pose space $\mathcal{C}_\mathcal{A}$. Consequently, the observation density estimate $\hat{p}(\mathbf{y}_t|\mathbf{s}^n_t)$ that is used for particle reweighting must be modified such that it can be applied within A or B. The same problem exists for the reweighting steps after mean shift iterations. We solve it in the following by evaluating (4.12) only for those parts whose pose parameters belong to the considered subspace.

A further potential caveat of the decomposition scheme becomes apparent when considering the sample points $\mathbf{z}^k_t$ from the kth assembly part model that appear in (4.12). We recall that they are placed along visible model features. In order to guarantee that the visibility prediction accurately accounts for inter-part occlusion, one would have to feed *all* assembly pose parameters into the visibility prediction process. This means that there are coupling effects between the assembly pose parameters on the level of visibility prediction. Decomposing the state space assumes that these coupling effects can be neglected. In order to alleviate this problem, we exploit the information that is provided by the kinematic tree of an assembly and use the reference parameters from (3.13) as initial guess for all pose parameters that aren't available in a given subspace. As soon as pose parameters have been recovered, they are used to update the initial guess which stabilizes the visibility prediction process of part model features to a large degree. In summary, state

space decomposition reduces the robustness of kernel particle filtering against inter-part occlusion. Despite this drawback, the evaluation chapter of this thesis shows empirically that, in practice, state space decomposition permits accurate and precise assembly pose localization for multi-part assemblies.

Of course, the state space decomposition scheme can divide $\mathcal{C}_\mathcal{A}$ into more than two disjunctive subspaces. For example, it has already been stated that Gavrila & Davis [GD96] chunk their original search space into three parts. The resulting subspaces correspond to the pose parameters of head+torso, arms, and legs. However, the authors don't indicate whether this setting is in some sense optimal. It thus remains unclear how many subspaces are desireable and how large their respective dimensionality should be.

With regard to the potential coupling effects, the best practice is to create subspaces of maximum possible dimensionality. Such subspaces are as large as kernel particle filtering can reliably manage. Therefore, it must be assessed how far kernel particle filtering might scale. To our knowledge, no theoretical bounds have been established for the KPF. But as mentioned before, Chang & Ansari suggest that their KPF works well within a range of 3 to 9 dimensions. To be on the safe side, $d = 6$ was chosen as subspace dimensionality. This coincides with the findings of Scott & Sain [SS04], who suggest that densities with 6 dimensions might be robustly estimated by means of kernel density estimation. The EKPF extension therefore decomposes the state space into 6-dimensional subspaces. In order to further exploit the spatial structure information of the assembly, this is done by recursively traversing the kinematic tree in a depth-first manner. For each node of the kinematic tree, a subspace is defined that subsumes the relative pose parameters of the part corresponding to the respective tree node.

The presented heuristic sequentially applies kernel particle filtering to subspaces of the assembly pose space. By means of this extension, an EKPF can localize assemblies that are composed from many parts. The following chapter provides empirical results on localizing different objects. For example, in the third experimental investigation the EKPF successfully localizes a subset of seven parts from a 20-part toy plane. To our knowledge, no approach has been published yet that scales equally well to an increasing number of parts. In order to give a complete overview of this new approach, the EKPF algorithm with all proposed extentions is presented in the next paragraph.

4.2.7 The Extended Kernel Particle Filter

This thesis contributes three important extensions to conventional kernel particle filtering. Algorithm 3 presents the pseudo-code of the resulting EKPF algorithm. Here, $\mathcal{T}$ is used to denote the kinematic tree of an assembly that is composed from j parts. Furthermore, B_k denotes the matrix that projects sample vectors to the subspace of the kth assembly part. Projected samples and their weights are marked with a high index k, accordingly.

Algorithm 3 Extended Kernel Particle Filter (EKPF)

Input: Particles $\{\mathbf{s}_{t-1}^n, w_{t-1}^n\}_{n=1}^{N_\mathrm{s}}$, image $\mathbf{y}_t$, bandwidth b, kinematic tree $\mathcal{T}$ with j nodes
// For single images, let $t = 1$ and initialize $\{\mathbf{s}_{t-1}^n, w_{t-1}^n\}_{n=1}^{N_\mathrm{s}}$ from prior $p(\mathbf{x}_0)$

1: $\hat{\mathbf{x}}_t \leftarrow$ `reference pose`$(\mathcal{T})$
2: **For all** $k \leftarrow$ `traverse`$(\mathcal{T})$ **do**
3: $\quad\{\mathbf{s}_t^{n,k}, w_t^{n,k}\}_{n=1}^{N_\mathrm{s}} \leftarrow$ `condensation`$(\{B_k\mathbf{s}_{t-1}^n, w_{t-1}^n\}_{n=1}^{N_\mathrm{s}}, \mathbf{y}_t)$ // Employs largest r_c.
4: $\quad$**For all** $i = 1 : I$ **do**
5: $\quad\quad(\{\mathbf{s}_{t,\mathrm{v}}^{n,k}\}_{n=1}^{N_\mathrm{s}},\ A^{-1},\ \bar{\mathbf{s}}_t^{n,k}) \leftarrow \mathrm{whitening}(\{\mathbf{s}_t^{n,k}\}_{n=1}^{N_\mathrm{s}})$
6: $\quad\quad\hat{f}_\mathrm{pilot}(\cdot) \leftarrow \mathrm{kde}(\{\mathbf{s}_{t,\mathrm{v}}^{n,k}\}_{n=1}^{N_\mathrm{s}}, b)$
7: $\quad\quad\{\lambda_n\}_{n=1}^{N_\mathrm{s}} \leftarrow$ `local bandwidths`$(\{\mathbf{s}_{t,\mathrm{v}}^{n,k}\}_{n=1}^{N_\mathrm{s}}, \hat{f}_\mathrm{pilot}(\cdot))$
8: $\quad\quad\{\mathbf{s}_{t,\mathrm{v}}^{n,k*}\}_{n=1}^{N_\mathrm{s}} \leftarrow$ `variable bandwidth mean shift`$(\{\mathbf{s}_{t,\mathrm{v}}^{n,k}, w_t^{n,k}\}_{n=1}^{N_\mathrm{s}}, \{\lambda_n\}_{n=1}^{N_\mathrm{s}}, b)$
9: $\quad\quad\{\mathbf{s}_t^{n,k*}\}_{n=1}^{N_\mathrm{s}} \leftarrow$ `retransform`$(\{\mathbf{s}_{t,\mathrm{v}}^{n,k}\}_{n=1}^{N_\mathrm{s}},\ A^{-1},\ \bar{\mathbf{s}}_t^{n,k})$
10: $\quad\quad\{w_t^{n,k*}\}_{n=1}^{N_\mathrm{s}} \leftarrow \{\frac{\hat{p}(\mathbf{y}_t|\mathbf{s}_t^{n,k*})}{q_\mathrm{ms}(\mathbf{s}_t^{n,k*})}\}$ // With decreasing r_c from 4.2.4.
11: $\quad\quad\{\mathbf{s}_t^{n,k}, w_t^{n,k}\}_{n=1}^{N_\mathrm{s}} \leftarrow \{\mathbf{s}_t^{n,k*}, \frac{w_t^{n,k*}}{\sum_{n=1}^{N_\mathrm{s}} w_t^{n,k*}}\}_{n=1}^{N_\mathrm{s}}$
12: $\quad$**End For**
13: $\quad\hat{\mathbf{x}}_t^k \leftarrow$ `strongest mode`$(\{\mathbf{s}_t^{n,k}, w_t^{n,k}\}_{n=1}^{N_\mathrm{s}})$
14: **End For**

Output: Recovered assembly pose $\hat{\mathbf{x}}_t = \sum_{k=1}^{j} \hat{\mathbf{x}}_t^k$ // For single images: stop here

Furthermore, a high index $*$ indicates mean shift optimized samples, whereas a low index v flags samples as variance normalized. The details of the EKPF algorithm are discussed in the following.

Line 1 of the EKPF algorithm initializes the estimated assembly pose $\hat{\mathbf{x}}_t$ by setting all part pose vectors $\hat{\mathbf{x}}_t^k$ with $k = 1 \ldots j$ to their reference pose values. These are obtained from the assembly pose specification $\mathcal{C}_\mathcal{A}$ that is associated to the kinematic tree $\mathcal{T}$. Afterwards, the index k of the next assembly part to consider is determined by recursing down the kinematic tree. In line 3, Matrix B_k projects all particle samples $\{\mathbf{s}_{t-1}^n\}_{n=1}^{N_\mathrm{s}}$ of the previous time step to the subspace that subsumes the pose parameters of the kth assembly part. An iteration of CONDENSATION subsequently updates the posterior estimate that is restricted to the current subspace. Note that this operation employs the same observation density estimate as the reweighting step in line 10 of the algorithm, namely $\hat{p}(\mathbf{y}_t|\mathbf{x}_t^k)$ from (4.11) with $\mathbf{x}_t^k = \mathbf{s}_t^{n,k*}$ for all $n = 1 \ldots N_\mathrm{s}$. Also remember that the associated feature visibility prediction employs the j part pose vectors $\hat{\mathbf{x}}_t^k$ as default values for all assembly pose parameters that don't belong to the subspace of the kth assembly part.

Once the posterior estimate of the current time step is available, lines 4-12 perform I iterations of mean shift. For this, the samples of the current time step undergo the same kind of variance normalization that was already detailed at the end of Chap. 4.2.3. Lines 6 and 7 perform the estimation of local bandwidth parameters which proceeds as described in Chap. 4.2.5. In line 8, the local bandwidths are finally employed for the variable

bandwidth mean shift which is calculated from (4.32). Afterwards, the samples are re-transformed to the original state space (line 9), reweighted (line 10) with the weighting function manipulations from Chap. 4.2.4, and normalized (line 11). Finally, after I iterations of mean shift, the position of the strongest local mode of the resulting particle set is determined. Line 13 expresses that this position is interpreted as the localized relative part pose of part k. The EKPF algorithm finally outputs the localized pose parameters of the assembly at time step t. This parameter set is obtained from the direct sum of the relative part pose vectors.

Regarding the details of the EKPF algorithm, it remains to be specified how the strongest mode of a posterior estimate is recovered. So far, this has only been detailed with regard to uni-modal posteriors. For them, an estimate of the assembly pose at time step t can be obtained from calculating (4.9). In the case of multi-modal posteriors, a robust estimate can be determined in a way that is related to calculating the variable bandwidth sample meanshift from (4.32). For this, the *potential modes* $\hat{\mathbf{x}}_t^k(\mathbf{s}_t^{r,k})$ near the particle samples $\mathbf{s}_t^{r,k}, 1 \leq r \leq N_\mathrm{s}$ are determined by calculating

$$\hat{\mathbf{x}}_t^k(\mathbf{s}_t^{r,k}) = \frac{1}{w_S} \sum_{\{\mathbf{s_t^{n,k}} | \mathbf{s_t^{r,k}} \in S_{b\lambda_n}(\mathbf{s}_t^{n,k})\}} \mathbf{s}_t^{n,k} w_t^{n,k}, \tag{4.33}$$

where w_S denotes the accumulated weight of all samples $\mathbf{s}_t^{n,k}$ that are selected for summation. Out of the potential modes $\hat{\mathbf{x}}_t^k(\mathbf{s}_t^{r,k})$, the one with the largest accumulated weight w_S is chosen as the strongest local mode $\hat{\mathbf{x}}_t$ of the posterior.

In its presented form, the EKPF algorithm outputs only the recovered assembly pose $\hat{\mathbf{x}}_t$ for time step t. This form was chosen because, within the experimental investigations of this thesis, the EKPF algorithm is only applied to single monocular images. However, the algorithm can also process image sequences. By retrieving the strongest m modes, the cartesian product over the associated relative part pose vectors yields an hypotheses set that can afterwards be reweighted. This results in a particle representation of the posterior at time step t that carries multiple assembly pose hypotheses and can be used as prior for the next time step $t + 1$.

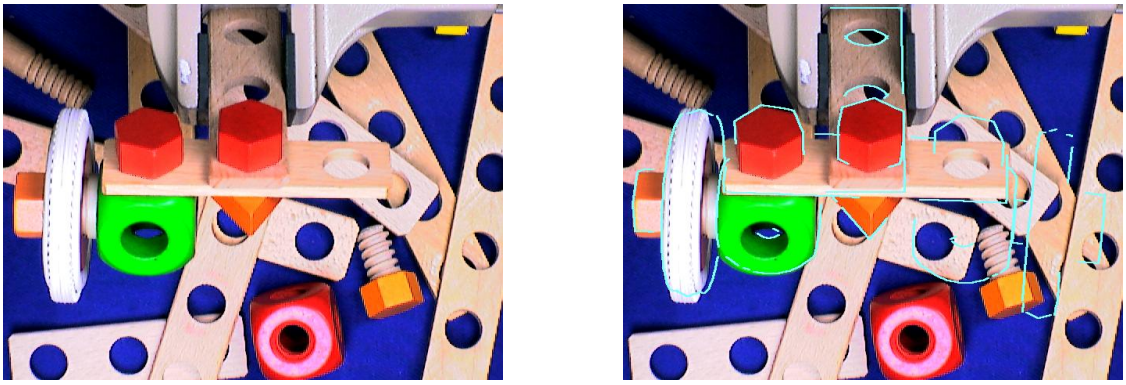
When operating with N_s particles and assembly models with a total of N_features contour edge features, the EKPF algorithm has a memory consumption in the order of $\mathcal{O}(N_\mathrm{s} + N_\mathrm{features})$, because the particle set of the current time step t and the time step before must be completely held in memory, together with the assembly representation. The overall memory consumption of assembly localization can therefore be characterized as moderate. For example, the memory representations of assemblies composed from up to 20 parts that are localized throughout the experimental investigations each consume less than 70MB of main memory. And for a 20-part assembly, two sets of 1000 particles approx. consume a total of 1MB memory.

In terms of computational effort, the worst-case time complexity of the EKPF depends on the number of mean shift iterations N_{ms}, the number of particles N_{s}, and the computational cost $C_{\mathrm{visibility}}$ of the two-stage visibility determination process for assembly model features that was analyzed in Chap. 3.3.1. With respect to these constants, the worst-case time complexity of the EKPF algorithm is expressed as $\mathcal{O}(N_{\mathrm{ms}} \cdot N_{\mathrm{s}}^2 + N_{\mathrm{ms}} \cdot N_{\mathrm{s}} \cdot C_{\mathrm{visibility}})$. The left part $N_{\mathrm{ms}} \cdot N_{\mathrm{s}}^2$ of the complexity expression results from the bandwidth selection and mean shift operations in lines 6-8 of Alg. 3. Both processes proceed by iterating over the complete particle set in an outer *and* an inner loop, which gives rise to the quadratic complexity in the number of particles. However, the operations that are performed within the inner loops are rather cheap, when compared to the computational effort of evaluating the observation density in lines 3 and 10. The latter task gives rise to the right part $N_{\mathrm{ms}} \cdot N_{\mathrm{s}} \cdot C_{\mathrm{visibility}}$ of the complexity expression. Thus, for particle sets with $N_{\mathrm{s}} \leq 1000$ which were used for the experimental investigations, the largest part of computing power is consumed by the process that projects model features to the image plane and predicts the visible part model features. This finding stresses the importance of employing a fast visibility determination process such as the one contributed in Chap. 3.3.1. Furthermore, it shows that optimizing the part model feature sets as proposed in Chap. 3.3.2 has the potential to yield a significant speed up when employing particle sets with several hundred elements.

4.3 Inspection Classification

The previous section has described the EKPF that recovers assembly poses from monocular images. Within the inspection system proposed by this thesis, the recovered pose information is the input of the classification module that was illustrated in figures 1.2 and 3.1. So far, this module hasn't been implemented in the inspection system prototype. Nevertheless, it is sketched in the following, how such a classification could be performed. In this way, a conceptual solution for the classification of part completeness and pose integrity is presented.

Figure 4.11 illustrates the problems that arise in the context of localizing the pose of assemblies with missing parts. The fundamental problem with this kind of situation is that the EKPF doesn't verify the existence of localized parts. It merely fits assembly models to given observations. If some parts of the currently inspected assembly are missing as in Fig. 4.11(a), the EKPF algorithm provides no means for noticing this problem. A resulting match is shown in Fig. 4.11(b). Clearly, an industrial inspection system needs to be able to verify whether the currently observed assembly has a complete set of parts. Given a recovered assembly pose, this task is much easier to solve than without pose information because the verification step must only classify whether the individual assembly parts are present at the recovered positions.

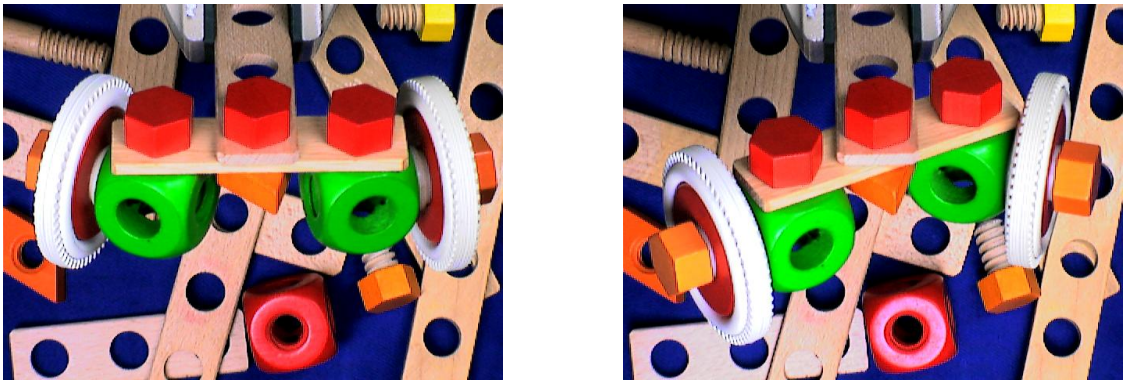


(a) (b)

Figure 4.11: The localization of assemblies with missing parts. a) The toy vehicle axle from experimental investigation 4. Six parts are missing on the right side of the assembly. b) The recovered the assembly pose. The hypothesized contour edges of the missing part models have been fitted to the background

As introduced in Chap. 2.5, classification in general is performed by assigning labels to sets of input features. For the verification of part completeness, the set of labels consists of the two elements C_{p} and C_{m}. The first label C_{p} denotes the event, or class, that an assembly part is present in the given scene. The label C_{m} indicates a missing assembly part. Concerning the input features, it would be desireable to reuse the image cues f_{fw}, f_{bw}, and f_{col} that are employed in the context of pose localization (cf. appendix C). They are well suited for the task because they are evaluated w.r.t. the sample points $\mathbf{z}_t^k$ of a specific assembly part with index k and the given image observation $\mathbf{y}_t$ at time step t. However, using multiple cues leads to the problem of cue integration.

A flexible and robust solution to the above formulated problem can be obtained from employing a Bayes classifier. The latter was already introduced in Chap. 2.5. This approach has the very appealing property of providing a probabilistic framework for the integration of different image cues into the classification process. As input, it operates on the class priors $P(C_{\mathrm{p}})$ and $P(C_{\mathrm{m}})$, and the conditional probability densities $p(f_c|C_{\mathrm{p}})$ and $p(f_c|C_{\mathrm{m}})$ with $c \in \{\mathrm{fw}, \mathrm{bw}, \mathrm{col}\}$. The class priors model the probability of the two classification events "part present" and "part missing" in the absence of any further information. The conditional probability densities represent, how well the image cues comply to the assumption that the assembly part for which the cues were evaluated is present in the image or missing. The class priors and the conditional probability densities can be estimated from representative labeled test sets. Such sets can be created with the assistance of the assembly localization module proposed in this thesis. This is done by localizing assemblies automatically, and subsequently performing a manual classification that picks out those parts that aren't present in the processed images. Afterwards, the Bayes classifier



(a) (b)

Figure 4.12: A toy vehicle axle in two different poses

verifies the existence of the kth assembly part at the recovered position and orientation by choosing

$$\underset{l\in\{\mathrm{p,m}\}}{\operatorname{argmax}} \; P(C_l) \prod_{c\in\{\mathrm{fw,bw,col}\}} p(f_c(\mathbf{z}_t^k, \mathbf{y}_t)|C_l), \tag{4.34}$$

where $\mathbf{z}_t^k$ again denotes sample points from the projected visible model features of the kth assembly part, and $\mathbf{y}_t$ is the current image. The classification rule assumes that the image cues depend conditionally on the output classes, only.

Concerning the verification of pose integrity, the classification task is to decide, whether a given pose adheres to the design specification of an assembly. Consider for example the toy vehicle axle from Fig. 4.12. Without external knowledge, it clearly is impossible to decide whether the two depicted assembly poses are valid or fault configurations. This knowledge must either be gathered from the assembly design process or from test sets that contain manually labeled examples of valid and fault configurations. From either of these information sources, two sets of assembly poses must be generated that correspond to valid and fault configurations of an assembly.

Once two reference sets of valid and fault pose configurations are available, new assembly poses can be classified against this reference. A simple approach would be to employ a NN classifier (cf. Chap. 2.5). However, this would imply to determine a suitable distance function. If the Euclidean distance doesn't yield reasonable results, an alternative would be to use decision trees. The latter have the advantage that tree training algorithms automatically generate short decision sequences that distinguish valid pose parameter ranges from those of fault configurations. Thus, in comparison to the NN classifier, a decision tree would be faster. On the other hand side, NN classifiers might perform better in cases where the input features from different classes are strongly mixed within the feature space.

4.4 Summary

This chapter has contributed a unique approach for the localization of multi-part assemblies. Together with the offline model preparation stage from the previous chapter and the sketched classification module, this thesis has thus described a complete system for the visual inspection of assemblies from monocular images. The system operation is automated to a large degree and envisioned to be integrated into a manufacturing environment. Except for the classification module that might be implemented based on existing techniques, a working prototype of the system has been implemented in Matlab and C++.

The first section of this chapter gave a detailed description of inspection task specifications. It specified precisely the minimal set of information that must be provided to the proposed inspection system such that the online inspection processes can proceed automatically. In other words, the section defined the interface between the proposed system and manufacturing environments.

The second section gave a full account of the new EKPF for assembly pose localization. It started with an overview of the assembly localization process that is followed by the system proposed in this thesis. Afterwards, promising literature in the field of particle filtering for visual tracking was presented. It was shown that interesting particle filtering approaches for the tracking of articulated objects have been published in the recent past. Furthermore, it was explained that neither of the techniques facilitates assembly pose localization due to the fact that the localization task doesn't allow to make use of a tracking assumption and due to a missing initialization of the pose parameters. Afterwards, the theoretical foundations of particle filtering were explained. An important subtype, SIR particle filtering, was considered in more detail. The CONDENSATION algorithm was presented as a well-known implementation, which allowed to give an in-depth account of the observation density model that is employed by the proposed system. Here, it was shown how the system combines different cues to obtain a likelihood function that measures how well the current image measurement agrees with specific pose hypotheses. The combination scheme is flexible to cue changes and modifications. Three example cues are detailed in the appendix.

After the cue combination scheme, kernel particle filtering was introduced. Its general concept was sketched first. Afterwards, this thesis contributed a thorough formal grounding of kernel particle filtering in kernel density estimation and gradient ascent through mean shift. What is more, the relationships between kernel particle filtering and other existing particle filters were defined clearly. This led to the conclusion that kernel particle filtering is mean shift guided CONDENSATION.

The second half of the section contributed three new extensions to conventional kernel particle filtering. Each extension was shown to aim at improving a specific weakness

of conventional kernel particle filtering. The first extension was motivated by the finding that density plateaus and narrow peaks can impact on the efficiency of mean shift based gradient ascent. The proposed extension manipulates weighting functions in a way that induces a coarse-to-fine behavior to the mean shift iterations. The second extension aims at selecting bandwidth parameters such that the bias and variance of kernel density estimates are minimized for a given number of particles. In order to achieve this minimization, the theoretically most promising adaptive kernel method was employed, together with the variable bandwidth mean shift for particle sets. The third extension aims at improving the scalability of kernel particle filtering. It was shown that the existing approach doesn't scale to the state space of multi-part assemblies, due to the fact that the underlying kernel density estimation process suffers from the curse of dimensionality. This problem was solved by contributing a state space decomposition scheme that divides the state space into subspaces of computationally tractable dimensionality. The theoretical limits and advantages of this heuristic were covered in detail.

The three extensions were finally combined within the extended kernel particle filtering (EKPF) algorithm. The algorithm was explained in detail and shown to exhibit moderate memory consumption. Furthermore, it was found to exhibit a worst-case time complexity that is quadratic in the number of particles, and linear in the number of mean shift iterations, assembly model features, and oriented bounding boxes. To our knowledge, this algorithm is the first particle filter that facilitates an accurate and precise localization of multi-part assemblies from monocular images. It has been reported in [SS06], which was published and presented at the DAGM 2006 Conference in Berlin[2]. Its performance aspects are evaluated in the following chapter.

Finally, the third section of this chapter discussed the classification of part completeness and pose integrity, while providing illustrative examples for both of the problems. As a conceptual solution to the problem of classifying part completeness, a Bayes classifier was proposed. It was shown how such an approach would allow to reuse the image cues from the pose localization module. Furthermore, NN classifiers and decision trees were identified as eligible techniques for the task of pose integrity classification.

[2]The paper presents central parts of the whole inspection system. Due to the space limitations, it only covers the variable bandwidth extension of the EKPF algorithm. Furthermore, the paper discusses the processing of single images, only.

5 Evaluation

Documenting the measurement accuracy and precision of a pose localization system such as the EKPF is difficult because many parameters influence the pose estimation process. For example, the imaging process might capture objects from different distances or with varying zoom settings. The resulting images are of different *scales*. With small image scales, each pixel represents a small area of the object space and objects appear large within the image. The larger the image scale grows, the smaller the respective objects will appear under projection to the image plane. Clearly, the EKPF will perform better for objects that appear large within an image than for apparently small objects. Further influences to the localization accuracy and precision arise e.g. from the perspective under which an object is perceived, lighting, clutter, the inspected objects, and the employed models.

In order to illustrate how well the EKPF can localize assemblies, four different experimental investigations were conducted that document the system performance under varying conditions. Table 5.1 presents an overview which shortly describes the key issues. Each experimental investigation involves the pose estimation of an individual object, with recovered DOF ranging from 5 up to 29. The localized objects are chosen from two application domains, namely a real industrial inspection scenario for experimental investigation 2, and assemblies built from the wooden building blocks provided by the `baufix`® construction set for experimental investigations 1, 3, and 4. The former domain allows to compare the achieved localization performance to an existing inspection system. The parts of the `baufix`® domain have the advantage of being widely available and standardized. Concerning the pose estimation task, they are very challenging because they are uniformly colored and thus provide no texture that could be exploited as image cue. Furthermore, the colored surfaces yield strong specular reflections and the edges

Table 5.1: Overview of the experimental investigations

Exp. No.	Assembly	Recovered DOF	Key issues
1	Screw-Cube	6	Varying perspective and image scale
2	Oil Cap	5	Industrial application, EKPF extensions
3	Toy Airplane	28	Multi-part assembly, no clutter
4	Toy Axle	29	Multi-part assembly, clutter, model optimization

(a) (b)

Figure 5.1: A screw-cube assembly under two different image scales and perspectives. a) The assembly is perceived under a $60°$ elevation from the yz-plane of the depicted coordinate system that is attached to the cube. The image scale is 0.1mm per pixel. b) The same assembly, perceived under an elevation of $0°$ and an image scale of 0.3mm per pixel

of all parts are rounded, which both impacts negatively on the quality of the resulting contour edges. With respect to the models, the true shape dimensions of the employed real wooden parts deviate up to 3% w.r.t. the largest model extent.

5.1 Experimental Investigation 1

The first experimental investigation is aimed at investigating the effects of the first two influencing factors stated above, namely image scale and perspective. In order to keep the effects of other influencing factors at the lowest possible level, a setting of low complexity is used. It is described and discussed in the following.

5.1.1 Methology and Data Sets

The inspected assembly is illustrated in Fig. 5.1. It simply consists of a wooden screw that is screwed into a wooden cube. The picture also shows that the cube is held fixed. Accordingly, the pose localization only needs to recover pose parameters of the screw. This is done relative to the cube. Among the recovered pose parameters, only the z-axis rotation and translation are considered in the following because only these two parameters could be reliably recorded as ground truth. The latter was carried out manually,

by means of a goniometer and vernier calipers. The accuracy of such measurements is expected to be better than 1° and 1mm.

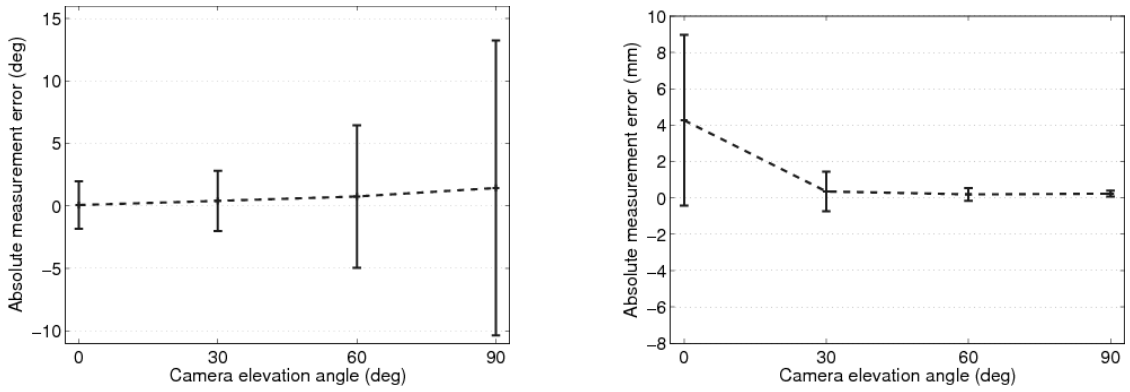
With this assembly, a total of 1000 image measurements was recorded in the following way. First, the cube was positioned at an elevation of 0° and a distance of 60cm from a statically placed camera. The camera's zoom lens was then adjusted to yield images with a scale of 0.1mm per pixel. Afterwards, the camera was calibrated and 125 images were captured that show the assembly with five different screw positions. The true screw translations and rotations w.r.t. the z-axis were recorded manually for each individual screw position. This procedure was repeated for an elevation of 30°, 60°, and 90°. Furthermore, the assembly was recorded under the same four elevation angles, but with an image scale of 0.3mm per pixel and a recalibrated camera. The scene was illuminated with two 110W cold light lamps that were statically placed to the left and right of the camera, in addition to the neon head lights of the lab.

In order to evaluate the data, the retrieved pose information was separated into 8 sets. These corresponded to the image measurements that were recorded under the two different image scales and four camera elevation angles. Each set consisted of 125 retrieved screw poses. For each set, the deviations of the recovered pose parameters from the manually measured ground truth were determined. Based on these deviations from the ground truth, the mean pose estimation errors and standard deviations were calculated. In the following, the mean error w.r.t. pose parameter deviations from the ground truth is used to document the absolute system accuracy, while the standard deviation is used to characterize the absolute precision.

5.1.2 Results

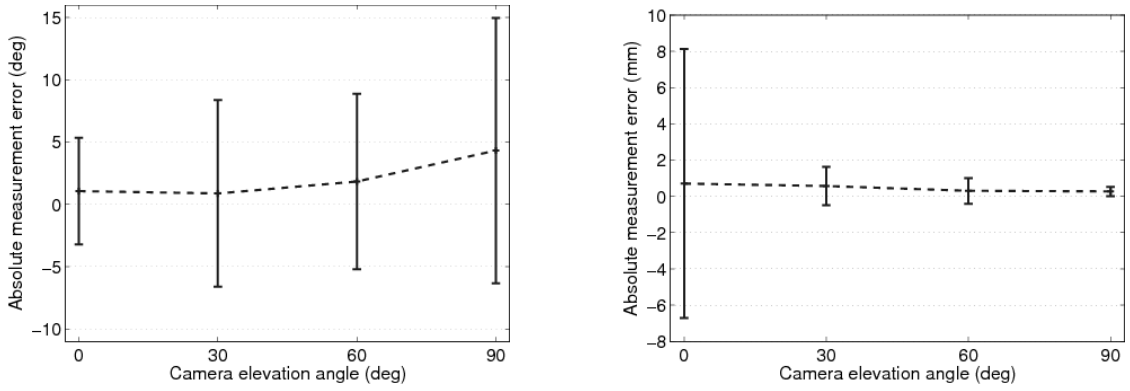
Figure 5.2 illustrates the results of the first experimental investigation, concerning the measurements with a small image scale of 0.1mm per pixel. It can be seen from Fig. 5.2(a) that the screw rotation is measured most accurately and precisely under a camera elevation angle of 0°. In this case, the hexagonal screw head is perceived straight from above, such as in Fig. 5.1(b). Given this perspective, the mean error of the screw rotation is smaller than 0.1°, with a standard deviation of 1.9°. For increasing camera elevation angles, the localization performance quickly decays, so that at an elevation of 90° no meaningful determination of the rotation parameter is feasible. The reason for this finding is that the screw shape exhibits strong rotational symmetries. The higher the elevation angle is, the smaller are the shape changes that result from a screw rotation around the z-axis.

Figure 5.2(b) shows that with regard to the screw translation, the measurement accuracy and precision develops quite differently. Here, a camera perspective associated with a 0° elevation yields the highest mean error of 4.3mm and a standard deviation of 4.7mm,



(a) (b)

Figure 5.2: The mean pose error and standard deviation at an image scale of 0.1mm per pixel. a) Recovering the screw rotation around the z-axis, under four different camera elevation angles. b) Recovering the screw translation along the z-axis



(a) (b)

Figure 5.3: The mean pose error and standard deviation at an image scale of 0.3mm per pixel. a) Recovering the screw rotation around the z-axis, under four different camera elevation angles. b) Recovering the screw translation along the z-axis

while the smallest values are achieved from a side-look position at $90°$ elevation (0.2mm mean error and standard deviation). The reason for this finding is that, from a side-look, a small translation of the screw along the z-axis yields a large change of the screw position within the image plane. In contrast to this, the only changes that a z-axis translation induces to an image perceived from $0°$ elevation result from depth changes which are comparatively small under the given setup.

Figure 5.3 shows that the same findings apply to the case of a larger image scale. The only difference is that the best achievable mean pose error and standard deviation are worse than for the small image scale. This is not surprising, given that the screw is now much less prominent in the image. For the rotation parameter, the mean error is now $1.1°$ with a standard deviation of $4.3°$ in the best case. For the translation parameter, we obtain a mean error and a standard deviation of 0.3mm each, which is still very accurate.

In comparison to the values that were presented in Tab. 2.1, the achieved performance is clearly competitive concerning the translation parameter. For the rotation parameter, the achieved performance is inferior to that of the systems reported in [vBGW03] and [HOW96]. However, the former system is specialized on single rigid objects, while the latter was tested on a desklamp with very elongated shape. The results of experimental investigation 3 show that our system performs equally well for parts that exhibit similar shape properties.

To conclude, the screw rotation and translation can both be recovered with satisfying accuracy and precision, given a suitable choice of the camera perspective. However, it must also be noted that it is impossible to measure the screw translation and rotation most accurately from the same camera perspective. Above all, this finding stresses the importance of conducting a proper inspection planning phase prior to any object localization. On a standard PC with a 2 GHz Pentium IV, running the EKPF with 500 Particles and 5 iterations of mean shift takes 5 seconds. The assembly representation consumes about 6MB of memory[1].

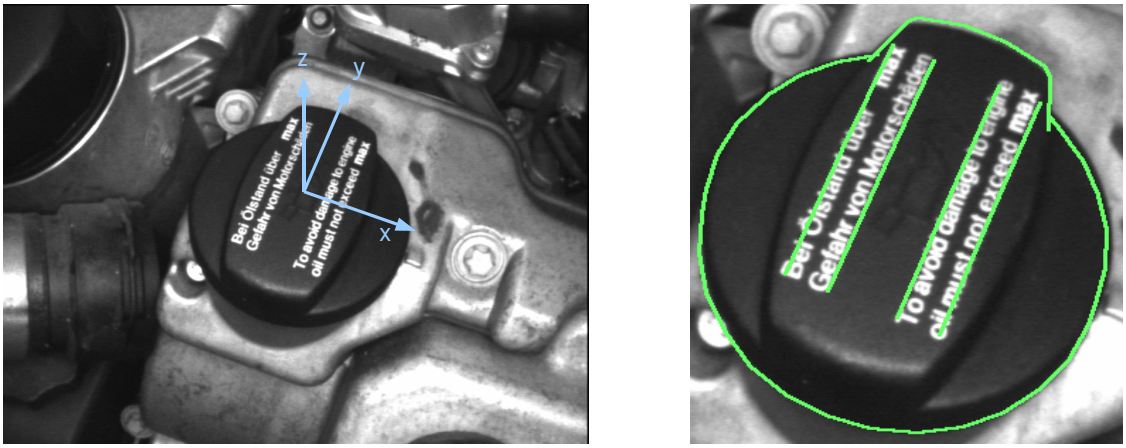
5.2 Experimental Investigation 2

The second experimental investigation is performed in a real industrial application domain. It provides the data for a direct comparison of the localization accuracy and precision with the performance of the system reported by Bank et al. [vBGW03] that was presented in Chap. 2.6. This comparison is feasible because the original image data, the object model, and the ground truth data from [vBGW03] have been kindly shared by the authors. Furthermore, the experimental investigation analyzes the effects of the individual EKPF extensions on the pose estimation performance.

5.2.1 Methology and Data Sets

The pose localization scenario of the second experimental investigation is visualized in Fig. 5.4. The left subfigure illustrates that the recorded image data shows an oil cap under

[1] All storage considerations presented in this thesis use the common but outdated convention of defining 1MB as 1024KB.



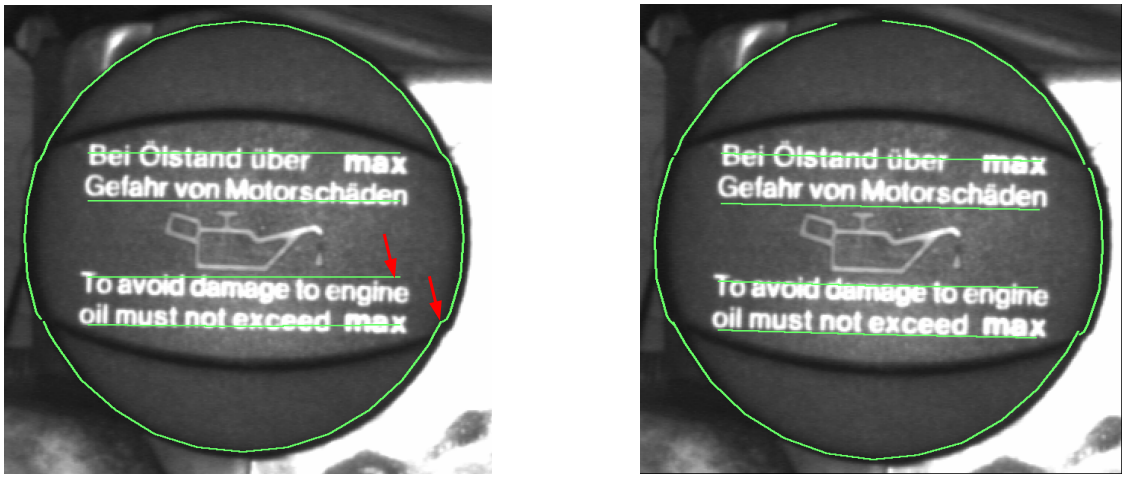
(a) (b)

Figure 5.4: The setup of experimental investigation 2. a) The inspected oil cap is observed from constant distance under different ZXZ fixed angle rotations of the depicted coordinate frame. The image scale is 2mm per pixel. b) The oil cap and a projection of its model. The automatically extracted contour edge model was manually annotated with lines that enclose the writing on the cap. All pictures and the model courtesy of DaimlerChrysler AG

different perspectives. The images were captured with a camera that was mounted to a robotic system, by placing the camera at equidistantly spaced positions along a hemisphere around the oil cap. The resulting images contain considerable clutter. The right subfigure shows that the oil cap model is annotated with four contour edge lines that enclose the written label on top of the cap. These contour edge lines were added manually to the automatically extracted features, in order to stabilize the rotation parameter estimation. As the oil cap is of circular shape, a pure automatically extracted contour edge model would otherwise have failed to provide enough information about the actual rotation of the cap within the image plane.

Bank and colleagues [vBGW03] report that their appearance based pose estimation approach is very accurate and precise in a small range of camera perspectives. In order to get comparable results, a range of camera perspectives was chosen that is similarly small. This range is represented by ZXZ fixed angle rotations in the intervals of $80°$ to $100°$ for the first Z-axis rotation, $30°$ to $50°$ for the rotation around X, and $-10°$ to $10°$ for the second Z-axis rotation. Within this range, 27 images were available. Each image was processed 70 times, yielding a total of 1890 pose measurements that were each compared to the ground truth[2]. Each recovered pose exhibits 5 DOF because the distance between camera and oil cap is known. Note that out of these 5 DOF, only the rotational parameters

[2]Bank et al. report that their ground truth has an accuracy of about $1°$.



(a) (b)

Figure 5.5: Mismatch of coordinate systems. a) The oil cap at a camera perspective of $(0°, 0°, 0°)$, together with the backprojected model. The arrows mark spots where the mismatch between the coordinate system of the image recording process and the model coordinate space is most obvious. b) Applying the compensation rotation to the ground truth values alleviates the mismatch. All pictures and the model courtesy of Daimler-Chrysler AG

are analyzed in the following because the ground truth doesn't contain any translation information. All pose measurements and the ground truth values were then converted to the corresponding XYZ fixed angle parameters, in order to maintain a constant representation throughout this thesis. The deviations of the rotation parameters from the ground truth were finally used to determine the mean pose error and its standard deviation for each of the three rotation parameters.

When calculating the mean pose errors and standard deviations, it became apparent that the provided ground truth refers to a coordinate system that is different from the coordinate system of the oil cap model. This can be seen very clearly from an image that shows the oil cap perceived straight from above as illustrated in Fig. 5.5(a). At the positions marked by the red arrows, the model deviates very clearly from the image allthough the correct ground truth values are used to project the model to the image plane. As a consequence, all recovered pose parameters are biased when compared to the ground truth, in addition to the inherent bias of the pose estimation process. In order to compensate for this systematic error, a compensatory rotation was determined, whose effects are visualized in Fig. 5.5(b). The compensation was determined as the rotation matrix that minimizes the sum of squared deviations of the recovered pose measurements from the ground truth values. The compensating rotation parameters were found to be the XYZ fixed angles $(-1.9°, -0.6°, -1.3°)$.

Apart from the intriguing possibility of obtaining a direct comparison of localization methods, the oil cap is an ideal test candidate to evaluate the individual effects of the various EKPF extensions that have been proposed in the previous chapter. The reason for this is that the state space decomposition heuristic is never applied for state spaces of less than 6 DOF. Thus, the EKPF can be used without weighting function manipulation or without adaptive bandwidth selection and the effects of the remaining extension can be assessed without possible side-effects from the state space decomposition. Even the KPF from from Alg. 2 can easily taken into the comparison. Therefore, the EKPF was run in different modes of operation and its localization accuracy evaluated as before. The KPF was employed, too. The gathered pose localization results are presented after the system comparison.

5.2.2 Results

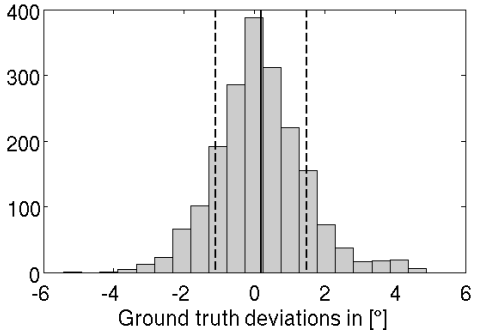
In Tab. 5.2, the mean pose estimation errors and standard deviations of the rotation parameter estimation with the EKPF are listed. When the compensation rotation is applied to the ground truth, the mean pose error of all rotation parameters is well below 1°, yielding equal performance to the system in [vBGW03] and all other systems that were introduced in Tab. 2.1 from Chap. 2.6. The standard deviations concerning the estimated Y- and Z-axis rotations are larger than the 1° achieved by Hel-Or & Werman in [HOW96]. Nevertheless, the results are competitive when taking into account that the oil cap exhibits strong rotational shape symmetries, in contrast to the elongated shape of the desklamp that was localized by the system of Hel-Or & Werman. The histograms of all rotation parameter deviations from the compensated ground truth are illustrated in Fig. 5.6. Without compensation, the mean pose error increases up to 3.7° for the Z-axis rotation because the mismatch in the model coordinate system and the ground truth origin introduces a systematic error to the pose measurements. But even in this case, the other two rotation parameters are recovered with reasonable accuracy.

In order to achieve the performance values indicated in Tab. 5.2, the EKPF was run with 1000 Particles and 2 iterations of mean shift. This takes about 6s on a 2GHz Pentium IV which is more than one order of magnitude slower than the approach reported in [vBGW03]. The memory consumption is 5MB. Figure 5.7 provides a visual impression of the achieved localization accuracy of the system, by means of the best and worst three results w.r.t. the sum of squared pose parameter deviations from the ground truth.

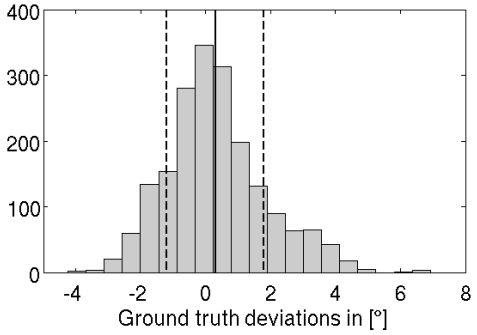
The results of performing the EKPF in different modes of operation and of the KPF performance are given in Tab. 5.3. The upper half lists the mean errors and standard deviations that were achieved together with applying the compensatory rotation to the ground truth. The lower half presents the same performance indicators for the uncompensated ground truth values. Throughout the table, *KPF* denotes the kernel particle filter from Alg. 2. Furthermore, *WFM-only* is an EKPF that was run with weighting function

Table 5.2: The mean error (μ) and standard deviation (σ) of the X,Y, and Z-axis rotation parameters w.r.t. the compensated and uncompensated ground truth

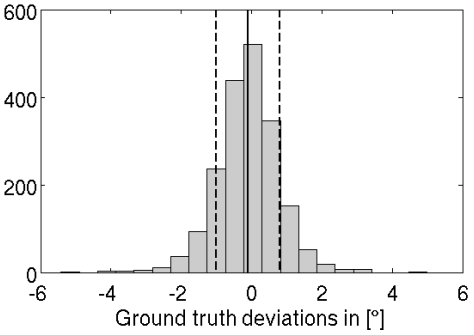
Compensation	μ_X	μ_Y	μ_Z	σ_X	σ_Y	σ_Z
On	$0.2°$	$0.3°$	$-0.1°$	$1.3°$	$1.5°$	$0.9°$
Off	$-1.4°$	$0.9°$	$3.7°$	$1.4°$	$1.6°$	$1.0°$



(a) X-axis rotation

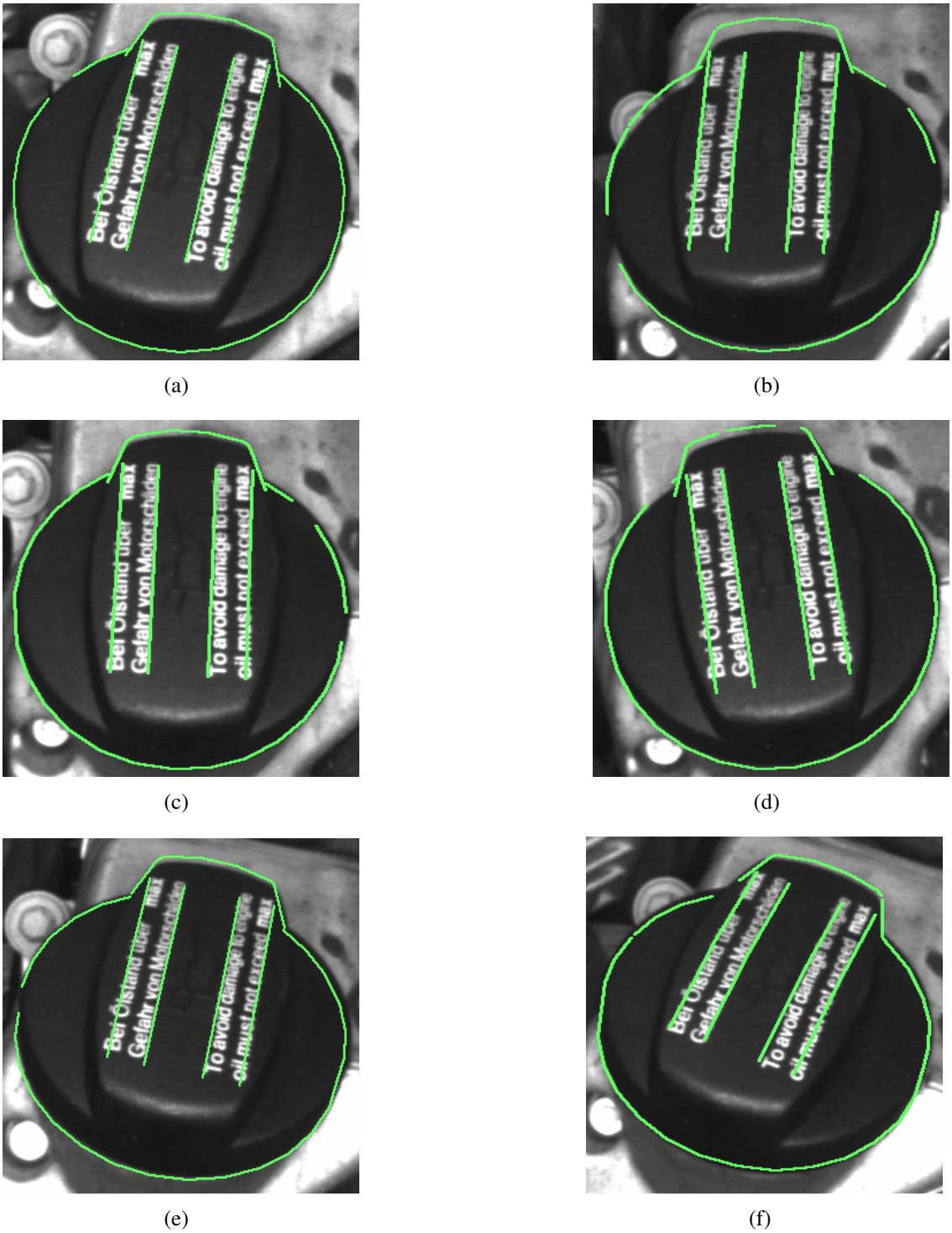


(b) Y-axis rotation



(c) Z-axis rotation

Figure 5.6: The histograms of the rotation parameter deviations from the compensated ground truth. Solid vertical lines denote the mean error. Dashed vertical lines show the standard deviation



(a) (b)

(c) (d)

(e) (f)

Figure 5.7: Results of the oil cap localization. The visible contour edges of the localized oil cap have been backprojected to the image plane under the recovered part pose. a,c,e) Best three results, ranked by the sum of squared deviations from the ground truth. The backprojected contour edges fit quite well to the physical object. b,d,f) Worst three results. All pictures and the model courtesy of DaimlerChrysler AG

Table 5.3: The mean error (μ) and standard deviation (σ) of the X,Y, and Z-axis rotation parameters w.r.t. the compensated and the uncompensated ground truth and different kernel particle filters

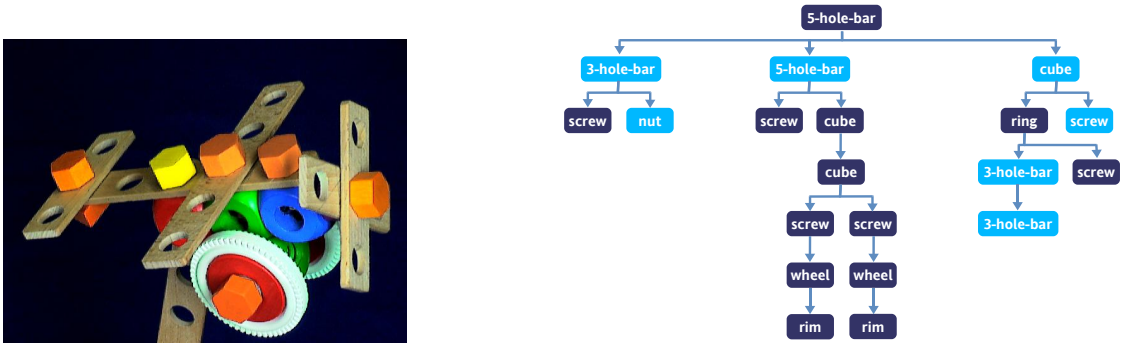
Compensation	Algorithm	μ_X	μ_Y	μ_Z	σ_X	σ_Y	σ_Z
yes	KPF	$-0.3°$	$1.0°$	$-0.1°$	$2.9°$	$3.3°$	$2.8°$
yes	WFM-only	$-0.5°$	$0.9°$	$0.1°$	$2.6°$	$2.9°$	$2.5°$
yes	ADAPT-only	$0.2°$	$0.5°$	$0.1°$	$1.6°$	$2.0°$	$1.3°$
yes	Full EKPF	$0.2°$	$0.3°$	$-0.1°$	$1.3°$	$1.5°$	$0.9°$
no	KPF	$-1.8°$	$1.6°$	$3.8°$	$3.0°$	$3.3°$	$2.8°$
no	WFM-only	$-2.0°$	$1.4°$	$3.8°$	$2.6°$	$3.0°$	$2.5°$
no	ADAPT-only	$-1.4°$	$1.1°$	$3.8°$	$1.7°$	$2.1°$	$1.3°$
no	Full EKPF	$-1.4°$	$0.9°$	$3.7°$	$1.4°$	$1.6°$	$1.0°$

manipulation but uses bandwidths like Alg. 2. In contrast to this, *ADAPT-only* denotes an EKPF that doesn't manipulate the weighting functions but performs adaptive bandwidth selection. Finally, *Full EKPF* stands for fully activated extensions.

When regarding the mean error of the recovered rotation parameters, it can be seen that the full EKPF always performs better than or at least equal to the KPF. Furthermore, when running the EKPF with only one extension, the achieved mean error lies in between the performance of the KPF and the EKPF, with the exception of the WFM-only case w.r.t. the X-axis rotation. However, the most remarkable result arises from analyzing the standard deviations of the rotation parameter deviations from the ground truth. For all three rotation parameters, the KPF measurements exhibit standard deviations that are at least twice as high as those of the full EKPF. Furthermore, each of the proposed extensions alone achieved better results than the KPF, but worse than for employing both extensions together. This finding indicates that the weighting function manipulation and the adaptive bandwidth selection schemes both significantly improve the measurement accuracy and precision of the EKPF.

5.3 Experimental Investigation 3

The first experimental investigation of this chapter investigated the pose estimation accuracy and precision of the EKPF in a low-complexity setting. Similar to this, the third experimental investigation aims at documenting the performance of the EKPF under favorable conditions, which in this case means scenes without clutter. However, unlike the first two investigations, the third experimental investigation deals with an assembly composed from 20 parts. It determines the pose estimation accuracy and precision based on a manually measured ground truth.



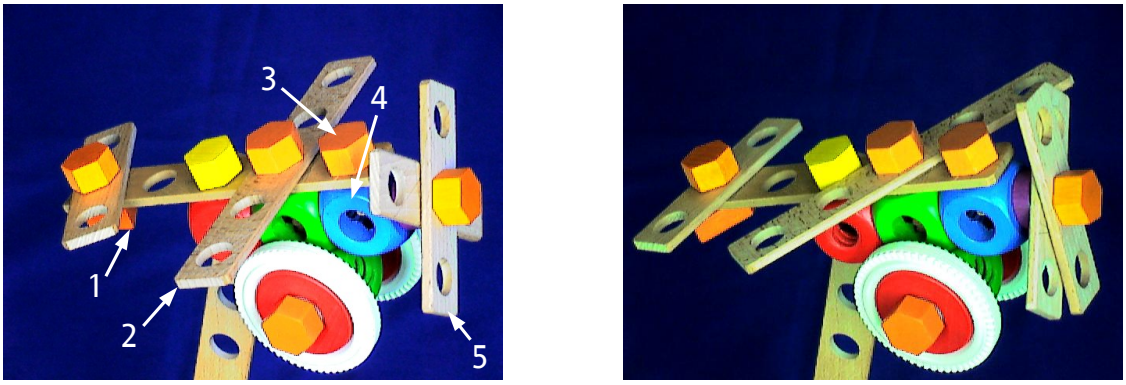
(a) (b)

Figure 5.8: The setup of experimental investigation 3. a) A 20-part toy airplane in its reference pose. b) A sketch of the airplane model's kinematic tree. Only the pose parameters of the parts in light blue are recovered

5.3.1 Methology and Data Sets

The setup of the third experimental investigation is illustrated in Fig. 5.8. It shows the assembly that is localized in the following, namely a 20-part toy airplane, together with a sketch of its kinematic tree. The assembly is mounted to a fixture consisting of a 5-hole-bar, a red cube and a yellow screw that are not part of the model. To manually record the ground truth data of each individual part would have been too difficult with the available vernier calipers and goniometer. Thus, only the part poses of the five best accessible parts are considered in the following. The respective parts are visualized in Fig. 5.9(a). As ground truth, the rotation parameter of parts 1, 2, 3, and 5 was recorded that was least constrained by the assembly structure. For example, the least constrained rotation parameter of the nut part 1 was the rotation around the screw thread to which the nut was attached. Furthermore, the least constrained translation parameter of part 1 and 4 was recorded, which was a translation along the individual screw thread to which the respective parts were attached. As in experimental investigation 1, the accuracy of the manual measurements is expected to be better than 1° for rotation parameters and better than 1mm for translation parameters.

The experimental investigation was conducted in the following way. First, a camera was statically placed at a distance of 80cm from the assembly. The camera was then zoomed to capture images with a scale of 0.3mm per pixel and calibrated to the fixture to which the airplane was mounted. Afterwards, 50 images of the assembly were captured in which the part poses vary systematically. The extreme points of variation are illustrated in Fig. 5.9. For each image measurement, the associated ground truth was recorded as indicated above. Then, the EKPF was used to estimate the pose parameters of parts 1 to 5, together with the pose parameters of their respective parents within the kinematic



(a) (b)

Figure 5.9: The range of recorded toy airplane poses. a) Pose parameters of the numbered parts are manually measured and recorded as ground truth. b) Throughout the experimental investigation, the assembly pose was varied in between the pose depicted in the left image and this pose

tree. Note that the root node part, i.e. the 5-hole-bar, was held by the fixture and its pose parameters known from the camera calibration. The complete set of seven localized parts is marked in light blue within the kinematic tree sketch in Fig. 5.8(b).

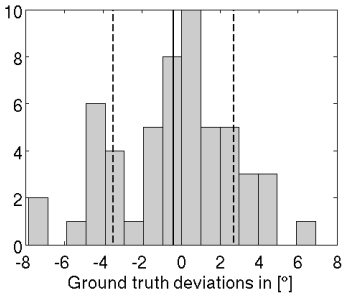
Taken together, the localized parts exhibit 42 DOF. However, 14 DOF are strongly constrained by the assembly structure. The structurally constrained parameters are part of the particle filtering state space but their values are nearly constant and known in advance from the assembly reference pose. Consequently, the number of DOF that were effectively recovered in this experimental investigation is 28. The pose parameters of the unlocalized parts were taken to be the values of the reference pose represented by the kinematic tree, though this constraint wasn't imposed on the real assembly when capturing the images. The scene illumination was again provided by two 110W cold light lamps that were placed to the left and to the right of the camera. Furthermore, the scene illumination was influenced by strongly varying amounts of daylight.

5.3.2 Results

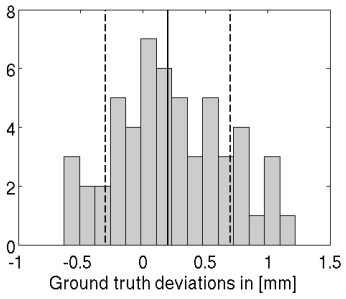
The mean error and standard deviation of the recovered pose parameters of parts 1 to 5 w.r.t. the ground truth is given in Tab. 5.4. For the rotation parameters, the mean error is $-1.4°$ in the worst case and $-0.3°$ in the best case, while the standard deviations range from $0.6°$ to $3.1°$. More specifically, the highest two standard deviations and mean errors are associated with the most challenging parts 1 and 3. The former experiences inter-part occlusion of up to 80% of its contour length while the latter exhibits strong

Table 5.4: The mean error (μ) and standard deviation (σ) of rotation and translation parameter measurements w.r.t. the ground truth

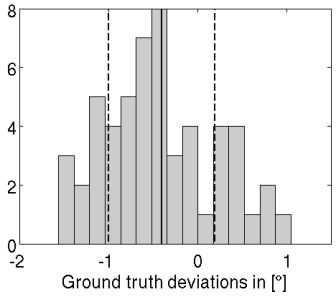
	Rotation				Translation	
	Part 1	Part 2	Part 3	Part 5	Part 1	Part 4
μ	-0.4°	-0.4°	-1.4°	-0.3°	0.2mm	0.3mm
σ	3.1°	0.6°	3.1°	2.1°	0.5mm	0.6mm



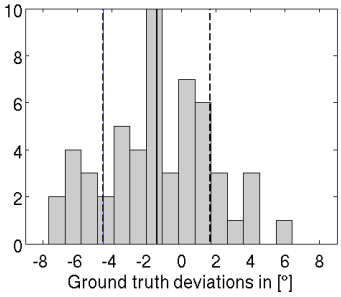
(a) part 1, rotation



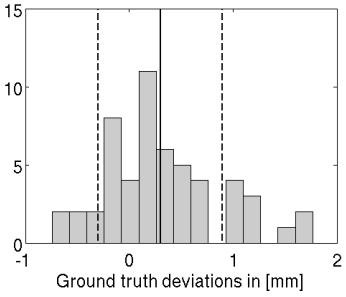
(b) part 1, translation



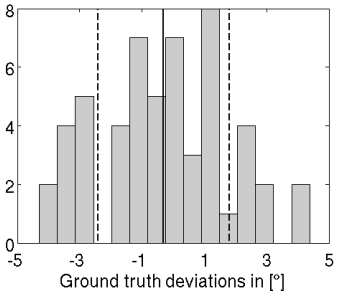
(c) part 2, rotation



(d) part 3, rotation

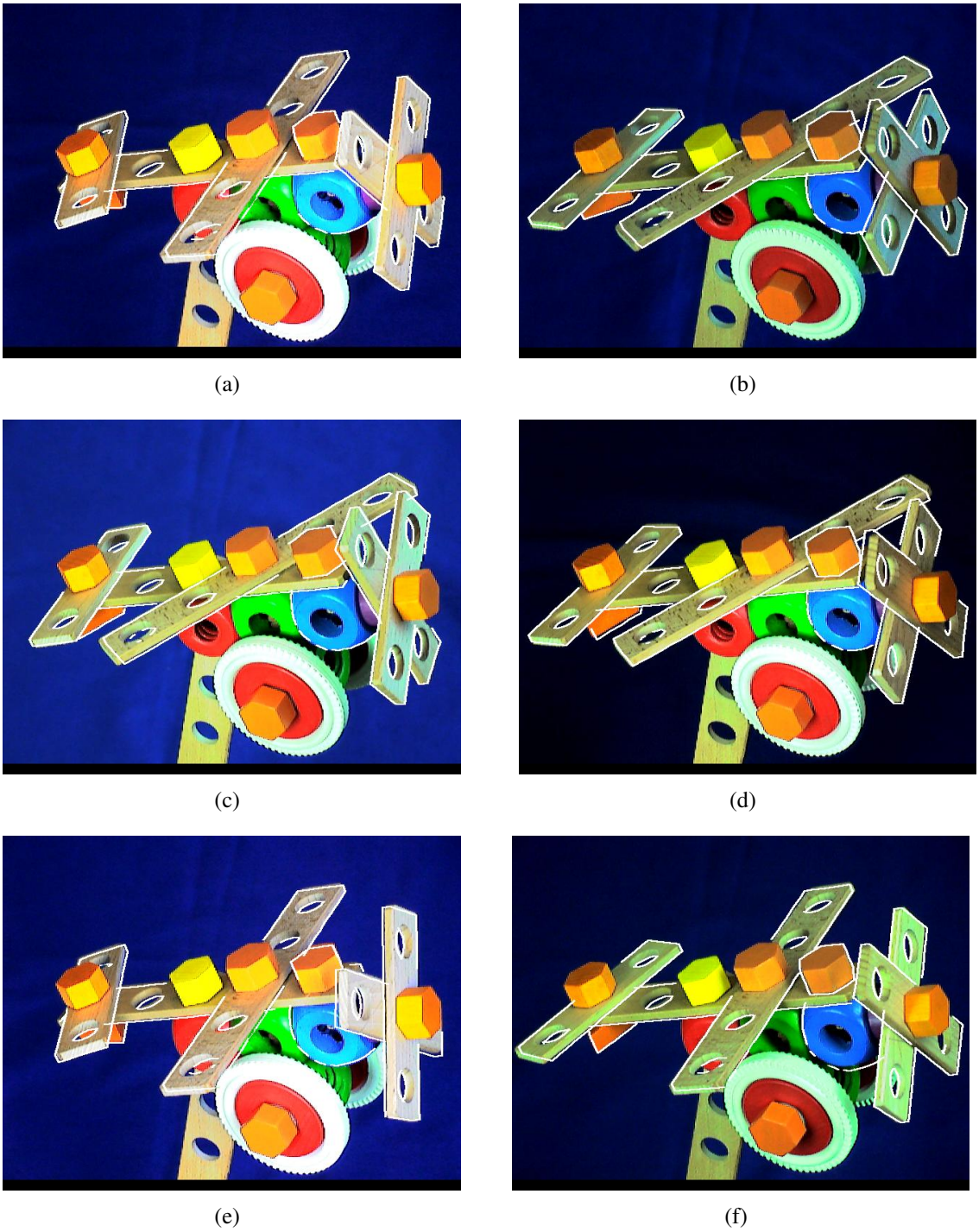


(e) part 4, translation



(f) part 5, rotation

Figure 5.10: Histograms of the pose parameter deviations from the ground truth. Solid vertical lines denote the mean error. Dashed lines visualize the standard deviation



(a) (b) (c) (d) (e) (f)

Figure 5.11: Results of the airplane localization. The visible contour edges of the localized airplane parts have been backprojected to the image plane under the recovered part poses. a,c,e) Best three results, ranked by the sum of squared deviations from the ground truth. The backprojected contour edges fit quite well to the physical objects. b,d,f) Worst three results

rotational shape symmetries w.r.t. the axis of the recovered rotation parameter. In contrast to this, the best localization performance is achieved for part 2 which provides favorable edge information due to its very elongated shape. For this part, the achieved results are competitive to the results of the system in [HOW96] which localized a desklamp that was of similarly elongated shape. Concerning the recovered translation parameters, our system again performs competitively well in comparison to all systems presented in Chap. 2.6, i.e. with a mean error and standard deviation of less than 1mm.

The results in Tab. 5.4 were achieved by executing the EKPF with 500 Particles and 5 iterations of mean shift. Recovering the 28 DOF takes 56s on a 2GHz Pentium IV. The memory consumption of the airplane assembly model is 41MB. Figure 5.11 provides a visual impression of the localization accuracy of the system, by means of the best and worst three results w.r.t. the sum of squared pose parameter deviations from the ground truth. Furthermore, Fig. 5.10 shows the resulting histograms of the pose parameter deviations from the ground truth.

5.4 Experimental Investigation 4

Experimental investigation 4 analyzes the localization accuracy and precision of the EKPF concerning multi-part assembly observations with cluttered backgrounds. The clutter is generated from a random distribution of parts that are also used within the observed assembly. Note that out of the five systems that have been presented in Chap. 2.6, only [vBGW03] and [Köl02] have been tested under similar conditions. The other three systems were only tested on images with a rather uniform background. As a cluttered background gives rise to many more contour edges than the ones arising from the observed assembly, there is a high potential for the localization process to be distracted from finding the correct part positions. Accordingly, the most important question within this investigation is, how well the system performs in such an environment. Additionally, the performance impact of model optimization is studied.

5.4.1 Methology and Data Sets

This experimental investigation performs the localization of a toy vehicle axle that is composed from 16 parts. It is illustrated in Fig. 5.12, together with a sketch of its kinematic tree. As in the previous investigation, the assembly is held by a fixture that is not part of the assembly model. But unlike the toy airplane from experimental investigation 3, the structure of the toy vehicle axle doesn't support the manual determination of the pose ground truth because the assembly is too small and to compact to reliably apply the goniometer. Therefore, a simulated ground truth was used in this investigation. The

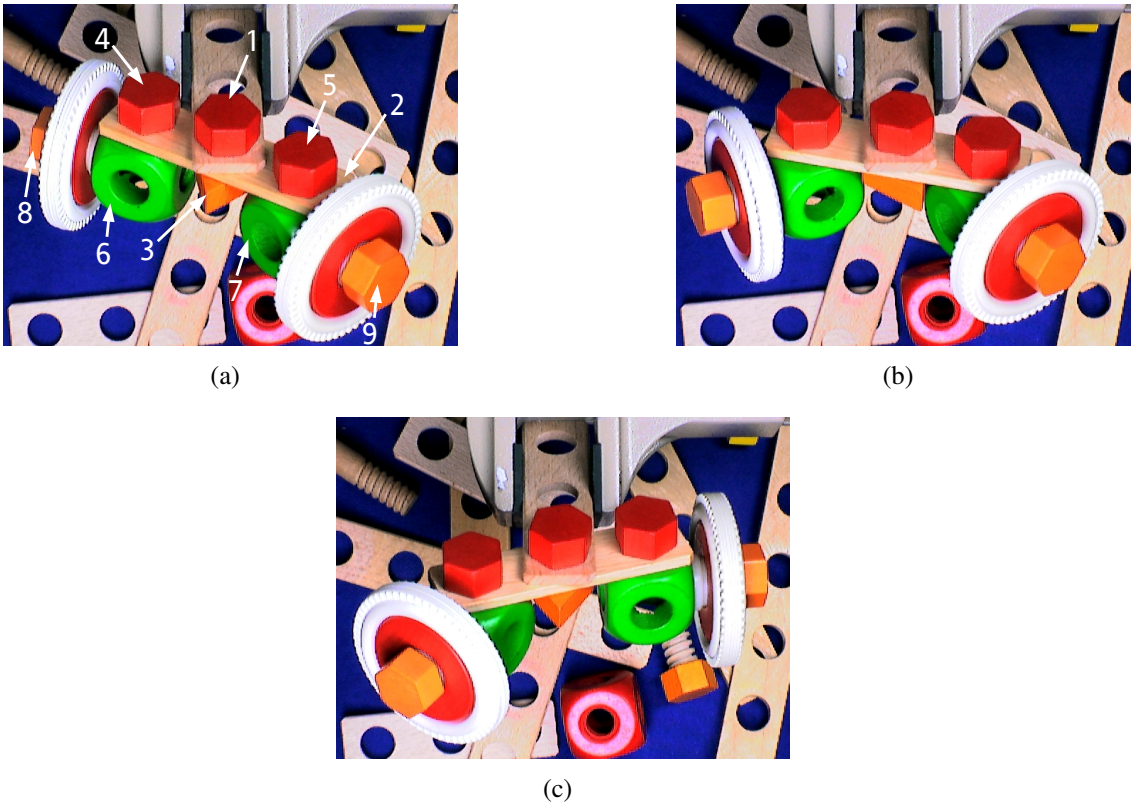
(a) (b)

Figure 5.12: The setup of experimental investigation 4. a) A 16-part toy vehicle axle in its reference pose. The image background is cluttered. b) A sketch of the vehicle axle's kinematic tree. Only the pose parameters of the parts in light blue are recovered

latter was obtained by performing a manual fit of the assembly model to all recorded images. Only the root node part position was determined from physical measurements, by means of placing a calibration target at the 3-hole-bar before the rest of the assembly was attached to it.

Out of the 16 parts of the toy vehicle axle, only 9 are considered in the following. These parts are marked in light blue in the kinematic tree sketch from Fig. 5.12(b) and also enumerated in Fig. 5.13(a). The wheels, bearings and rims are left out of the considerations, because the correct respective models weren't available. The models that were employed instead were just similar enough in shape to facilitate the correct localization of the remaining parts. Concerning the localized parts, only the most strongly varying pose parameter per part is analyzed in the following. For parts 1 to 7, this is the rotation around the screw thread, which either belongs to the respective part or to which the part is attached. For the screws 8 and 9, the analyzed parameter is the translation along their screw axis.

The experimental investigation was conducted in a way similar to that of the previous one. First, a camera was statically placed at a distance of 80cm from the assembly. The camera was then zoomed to capture images with a scale of 0.2mm per pixel and calibrated. Afterwards, 100 images of the assembly were captured in which the part poses vary systematically. The extreme points of variation are illustrated in Fig. 5.13. For each captured image, the associated simulated ground truth was determined as indicated above. Then, the EKPF was used to estimate the pose parameters of parts 1 to 9. This was done 50 times per recorded image, resulting in a total of 5000 assembly pose measurements.



(a) (b)

(c)

Figure 5.13: The range of recorded toy vehicle axle poses. Throughout experimental investigation 4, the assembly part poses were varied in between the depicted configurations. Only the numbered parts in a) were localized

Taken together, the localized part poses exhibit 54 DOF. But due to the constraints arising from the assembly structure, only 29 DOF had to be recovered effectively. All other pose parameters were set to the values of the respective reference pose parameters from the kinematic tree. The scene illumination was again provided by two 110W cold light lamps that were placed to the left and to the right of the camera. Furthermore, the scene illumination was influenced by strongly varying amounts of daylight.

In order to evaluate the pose localization performance in the context of model optimization, three vehicle axle models were used that had been generated from varying degrees of optimization. Model I was obtained without employing any contour edge feature optimization. Model II resulted from setting Δq^j in Eqn. (3.9) to accept a maximal loss of 30% of the expected number of active contour edge features per view. Model III was obtained from refining model II by optimizing the models of all unlocalized parts with a Δq^j of 90%. Furthermore, the contour edge features of the nut part in model III were completely unoptimized, for reasons that will be explained within the discussion of re-

Table 5.5: The mean error (μ) and standard deviation (σ) of rotation and translation parameter measurements w.r.t. the simulated ground truth and vehicle axle model I

	Rotation							Translation	
	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	Part 8	Part 9
μ	0.2°	-0.3°	0.4°	-0.1°	0.3°	-1.9°	-1.7°	0.4mm	-0.1mm
σ	2.5°	2.6°	2.9°	2.6°	3.3°	3.6°	3.2°	1.1mm	0.6mm

sults. Thus, vehicle axle model III was a hybrid model in which the localized parts were represented with a fair amount of contour edge features, while the unlocalized parts were represented with a feature set that was just large enough to provide some visual feedback for the visual analyzation of localization results.

5.4.2 Results

In the following, the localization results are first discussed w.r.t. the unoptimized vehicle axle model I. Afterwards, the achieved performance will be compared to the results of employing the optimized assembly models II and III.

The mean error and standard deviation of the recovered pose parameters of parts 1 to 9 w.r.t. the simulated ground truth and model I is given in Tab. 5.5. For the rotation parameters, the mean error is $-1.9°$ in the worst case and $-0.1°$ in the best case, while the standard deviations range from $2.5°$ to $3.6°$. For the translation parameters, the mean error ranges from 0.4mm to -0.1mm, while the standard deviations are 1.1mm and 0.6mm in the worst and in the best case. When comparing these results to the performance achieved in experimental investigation 3, it can be seen that the worst and best case standard deviations are higher than in the previous investigation. Thus, the clutter has clearly impacted on the pose estimation accuracy and precision. Nevertheless, no other system is known to us that performs a similarly accurate and precise pose localization of complex multi-part assemblies from monocular images with cluttered background.

The results in Tab. 5.5 were achieved by executing the EKPF with 200 Particles and 2 iterations of mean shift. Recovering the 29 DOF takes 91s on a 2GHz Pentium IV. The memory consumption of the vehicle axle assembly model is 69MB. Figure 5.14 provides a visual impression of the localization accuracy of the system, by means of three good and three bad results. The illustrated results were taken out of the 100 best and worst w.r.t. the sum of squared pose parameter deviations from the simulated ground truth. The histograms of the pose parameter deviations from the simulated ground truth are given in Fig. 5.10.

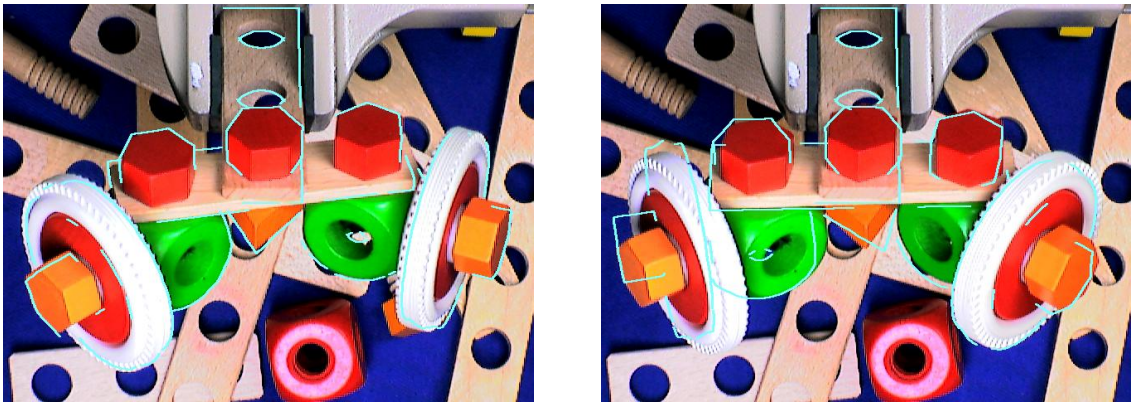
Table 5.6 compares the results that have been discussed so far to the pose localization results achieved with the other two vehicle axle models. Concerning model II, it can

Table 5.6: The mean error (μ) and standard deviation (σ) of rotation and translation parameter measurements w.r.t. the simulated ground truth and vehicle axle models I, II, and III

	Rotation							Translation	
	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	Part 8	Part 9
	Model I								
μ	0.2°	-0.3°	0.4°	-0.1°	0.3°	-1.9°	-1.7°	0.4mm	-0.1mm
σ	2.5°	2.6°	2.9°	2.6°	3.3°	3.6°	3.2°	1.1mm	0.6mm
	Model II								
μ	0.3°	-0.5°	-0.3°	-0.1°	0.4°	-2.1°	-1.8°	0.4mm	-0.1mm
σ	2.5°	2.6°	11.7°	2.5°	3.2°	3.7°	3.2°	1.1mm	0.6mm
	Model III								
μ	0.3°	-0.5°	-0.3°	-0.1°	0.3°	-2.1°	-1.8°	0.3mm	-0.1mm
σ	2.7°	2.8°	2.8°	2.4°	3.4°	3.7°	3.2°	1.1mm	0.6mm

be seen that the localization accuracy and precision stays quite close to the values of the unoptimized model, with the only extreme exception of the rotation parameter of part 3. In this case, model optimization has clearly exceeded a critical level of contour edge feature reduction. Note that part 3 is quite small and experiences strong inter-part occlusion within the captured images. This is why within model III, part 3 was completely unoptimized. As a result, the standard deviation w.r.t part 3 and model III is very similar to that of model II. To conclude, Table 5.6 shows that a careful optimization of contour edge feature sets only has a minor impact on the localization accuracy and precision of the EKPF.

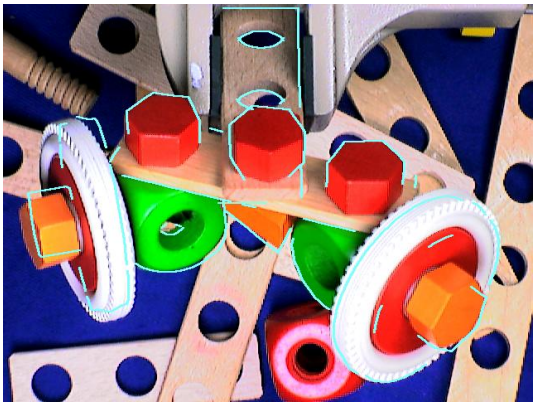
The real benefit from using model optimization becomes clear when comparing the execution speed of the EKPF w.r.t model I, II, and III. As indicated above, the EKPF needs 91s on a 2GHz Pentium IV when localizing the vehicle axle with model I. The latter vacates 69MB of main memory. Working with model II and the same parameterization of the EKPF takes 67s, which is a speed up of 26%. The memory consumption of model II is 57MB. Finally, performing pose localization with model III takes 41s, which achieves a speed up of 55%. The model consumes 31MB of memory. In summary, the model optimization stage that was introduced in Chap. 3.3 facilitates a reduction of computational load, while even improving the moderate memory consumption of the EKPF. When running the inspection system with model III, the total memory consumption of the system (including all image preprocessing results, particles, assembly model data, and loaded software libraries) is below 100MB.



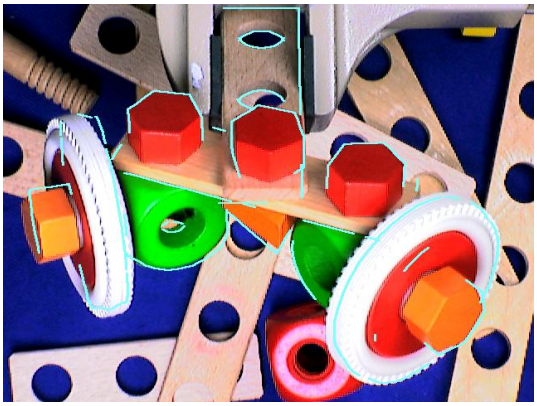
(a)

(b)

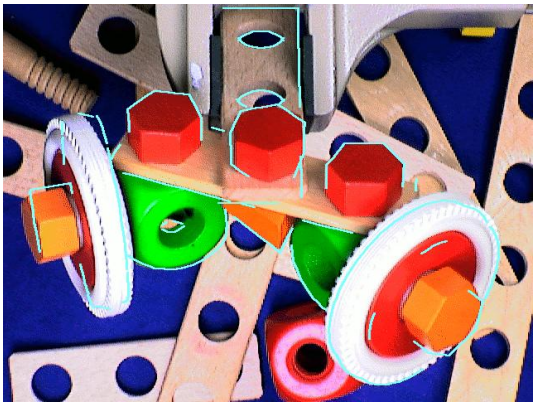
(c)



(d)

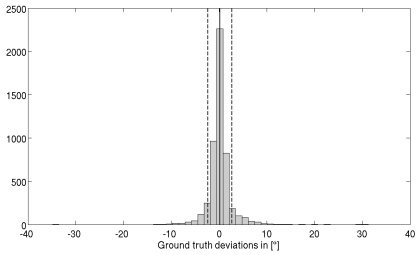


(e)

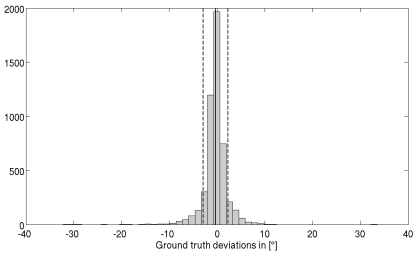


(f)

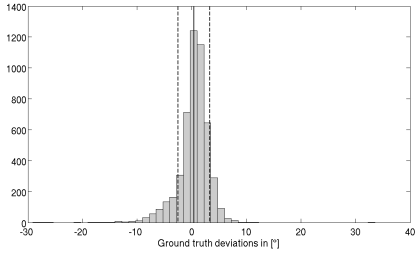
Figure 5.14: Results of the vehicle axle localization. The visible contour edges of the localized parts have been backprojected to the image plane under the recovered part poses. a,c,e) Three results out of the 100 best w.r.t the sum of squared deviations from the simulated ground truth. The backprojected contour edges fit quite well to the physical objects. b,d,f) Three results out of the 100 worst



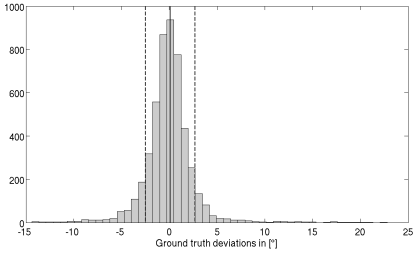
(a) part 1, rotation



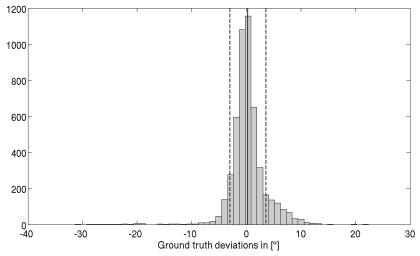
(b) part 2, rotation



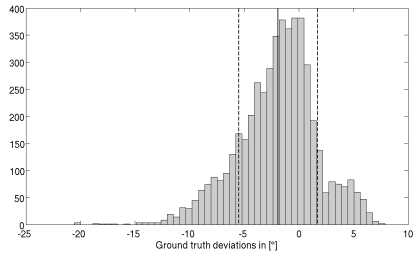
(c) part 3, rotation



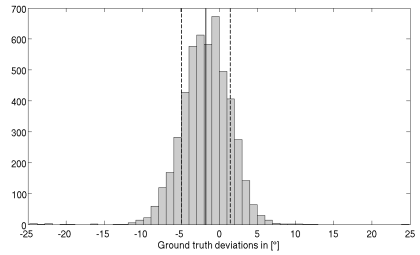
(d) part 4, rotation



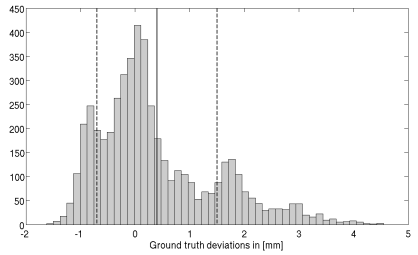
(e) part 5, rotation



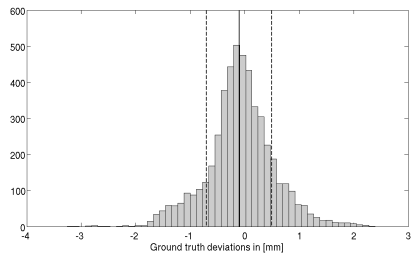
(f) part 6, rotation



(g) part 7, rotation



(h) part 8, translation



(i) part 9, translation

Figure 5.15: Histograms of the pose parameter deviations from the simulated ground truth. Solid vertical lines denote the mean error. Dashed lines visualize the standard deviation

5.5 Summary

Experimental investigation 1 documents the accuracy and precision of the localization module for a baufix® screw-cube assembly under varying perspective and image scale. The investigation is carried out w.r.t. a manually measured ground truth. The screw rotation around the screw thread is determined with an accuracy and precision that is slightly inferior to that of the reference systems reported in Chap. 2.6. Regarding the screw translation along its thread, the achieved localization accuracy and precision is competitive to that of the reference systems.

Experimental investigation 2 estimates the 5 DOF pose of an oil cap. The ground truth is given for the three oil cap rotation parameters. Within this industrial application domain, the EKPF pose estimation accuracy w.r.t. the three rotation parameters equals that of the reference system from [vBGW03], which has been tested on the same data. However, the system of [vBGW03] is specialized on estimating object poses with up to 6 DOF. Consequently, the processing speed of the specialized system is considerably higher than that of the EKPF. The EKPF pose estimation precision is comparable to that of the other reference systems, when taking into account that the oil cap exhibits strong rotational symmetries that principally impact negatively on the pose estimation accuracy and precision. By testing the EKPF repeatedly on the same data but in different modes of operation, it is demonstrated that the proposed weighting function manipulation and adaptive bandwidth selection schemes can both significantly improve the pose estimation accuracy and precision.

Experimental investigation 3 analyzes the pose estimation capabilities of the EKPF concerning image measurements with uniform background and a 7-part subset of a 20-part baufix® airplane. The seven localized parts effectively exhibit 28 DOF, out of which four rotation parameters and two translation parameters are available as ground truth. Regarding the achieved pose estimation accuracy, the EKPF performs as good as the reference systems for five out of the six ground truth pose parameters. Regarding the pose estimation precision, the EKPF achieves the reference system performance w.r.t. the two recorded translation parameters and one out of four rotation parameters.

Experimental investigation 4 documents the localization accuracy and precision of the EKPF concerning image measurements with cluttered background and a 9-part subset of a 16-part baufix® vehicle axle. A simulated ground truth of the assembly pose is available. For all of the seven rotation parameters and the two translation parameters that are analyzed, the mean error of the pose deviations from the simulated ground truth is smaller than 2° and 0.5mm. The corresponding standard deviations are all smaller than 3.7° and 1.2mm. Furthermore, by employing two additional assembly models that have been generated with varying degrees of contour edge feature optimization, it is demonstrated that the model optimization scheme facilitates a reduction of computational load, while further improving the moderate memory consumption of the EKPF.

The four experimental investigations are carried out on a 2GHz Pentium IV PC with 1GB of main memory. In all four cases, the total memory consumption of the whole image processing system is well below 150MB. Depending on the size of the employed particle sets, the number of mean shift iterations, and the number of contour edge features and oriented bounding boxes within the employed assembly models, the processing speed varies between 5 to 10 seconds per localized part.

6 Conclusion and Outlook

Automated visual inspection is fundamental in the endeavor to manufacture products of increasing complexity. In the context of quality assurance procedures, its chief purpose is to identify errors within production processes, possibly before they become defects. Furthermore, by tracing the identified errors and adapting the involved production processes accordingly, many errors can be prevented from being made at all.

This thesis contributes a new system for automated visual inspection. Regarding prior work in this field, the proposed system is unique in the following ways.

1. The system extends the localization of objects composed from single or few rigid parts to the case of true multi-part assemblies. It employs a new assembly pose estimation technique, namely the extended kernel particle filter (EKPF). The EKPF determines assembly poses from monocular images and is robust against occlusion between parts.

2. The EKPF integrates and extends a number of existing techniques. Individually, the integrated techniques don't facilitate multi-part assembly pose estimation with equal accuracy and precision, when allocated comparable amounts of system resources.

3. In addition to online localization, the system also performs offline assembly model preparation. It automatically extracts contour edge features from 3D CAD models of rigid parts. Furthermore, it supports the optimization of part feature models w.r.t. storage, and their combination to representations of multi-part assemblies. The resulting assembly models efficiently and accurately represent feature visibility under different assembly pose configurations and perspective occlusion.

Except for its classification module that is presented only conceptually, the proposed system has been fully implemented. The offline model preparation is written in MATLAB®, while the EKPF is written in C++. As image processing platform, the iceWing toolkit of Frank Lömker is used, which is licensed under the GNU General Public License and freely available at sourceforge.net.

The evaluation at the end of this thesis reports four experimental investigations that document the pose localization performance of the system prototype under varying conditions. The localized assemblies exhibit up to 29 recovered DOF. Regarding the results, no

other system is known to us that localizes complex multi-part assemblies from monocular images with comparable accuracy and precision. In this respect, the proposed system therefore defines the state-of-the-art. At the same time, the evaluation yields empirical evidence which indicates that the system offers near state-of-the-art localization accuracy and precision when localizing single parts.

Concerning future work, the most important step clearly is to implement the classification module that is sketched in this thesis. Furthermore, there is still untapped potential for speeding up the performance of the pose localization module. For example, the oriented bounding boxes of the assembly models could be organized hierarchically. By testing against the hierarchically organized bounding boxes in a coarse-to-fine fashion, many of the contained part model features could be ruled out as invisible much faster than with the current approach. Consequently, the number of operations that have to be performed for the online feature visibility prediction could be significantly reduced. Finally, the localization module could be extended to deal with rather flexible objects like cables and tubes. This could for example be achieved by integrating the approach that was recently proposed in [Ell05].

A OBB Generation

The generation scheme of oriented bounding boxes (OBBs) that is sketched in the following is a simplified version of the approach proposed in [GLM96, pp. 6-8]. It starts by considering the vertices of the normalized 3D CAD model of a rigid part. For this part, a set of OBBs can be generated by the following decomposition principle.

1. Calculate the mean μ and the three-by-three covariance matrix C of the vertices of the normalized CAD part.
2. Create an initial OBB. Its center is at μ, while the three eigenvectors of C define its basis. In order to size the new OBB, find the extremal vertices along the OBB axes and set the box size to tightly bound all vertices.
3. Split the OBB in two. The split is performed with a plane that is orthogonal to the longest OBB axis and includes the box center. Assign the vertices within the unsplit box to either of the new ones. Then update the center points, bases, and sizes of the two resulting OBBs.
4. Keep splitting the OBBs according to the previous step, until a maximum total number of OBBs is reached or until all box axes are smaller than a given threshold.

B Importance Sampling

The following considerations largely follow the discussion presented in [AMGC02]. The *importance sampling principle* is concerned with the problem of sampling from the posterior pdf $p(\mathbf{x}_t|\mathcal{Y}_t)$. Recall that the posterior can be approximated as

$$p(\mathbf{x}_t|\mathcal{Y}_t) \approx \sum_{n=1}^{N_\mathrm{s}} w_t^n \delta(\mathbf{x}_t - \mathbf{s}_t^n), \tag{B.1}$$

where $\mathbf{x}_t$ is the current system state, $\mathcal{Y}_t$ is the history of image observations, δ is the Dirac delta function, and $\{\mathbf{s}_t^n, w_t^n\}_{n=1}^{N_\mathrm{s}}$ is a particle set representation of the posterior. As stated in Chap. 4.2.2, in the context of particle filtering the posterior pdf usually can't be sampled directly. However, usually a pdf $\pi(\cdot)$ can be evaluated at given sample positions which is known to be proportional to the posterior pdf, i.e. $p(\mathbf{x}_t|\mathcal{Y}_t) \propto \pi(\mathbf{s}_t^n|\mathcal{Y}_t)$. Assume that such a $\pi(\cdot)$ is given, and that an *importance density* $q(\cdot)$ is known, from which one can easily sample the particles $\mathbf{s}_t^n \sim q(\mathbf{x}_t|\mathcal{Y}_t)$. If this is done, choosing the particles weights according to

$$w_t^n = \frac{\pi(\mathbf{s}_t^n|\mathcal{Y}_t)}{q(\mathbf{s}_t^n|\mathcal{Y}_t)} \propto \frac{p(\mathbf{s}_t^n|\mathcal{Y}_t)}{q(\mathbf{s}_t^n|\mathcal{Y}_t)} \tag{B.2}$$

yields a valid particle set representation of the posterior.

In order to employ this approach within particle filtering, the distribution $q(\cdot)$ is chosen to factorize according to

$$q(\mathbf{x}_t|\mathcal{Y}_t) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathcal{Y}_t) q(\mathbf{x}_{t-1}|\mathcal{Y}_{t-1}). \tag{B.3}$$

In that case, the samples $\mathbf{s}_t^n$ of the current time step t can be obtained by augmenting those of the previous time step with the new state $\mathbf{s}_t^n \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathcal{Y}_t)$. However, the weight update scheme must be adapted to the factorized version of $q(\cdot)$, too. For this, it can be shown that the posterior is proportional to

$$p(\mathbf{x}_t|\mathcal{Y}_t) \propto p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathcal{Y}_{t-1}), \tag{B.4}$$

if the observation of the current time step only depends on the current system state, and if the same assumption is made as in (B.3) that the current state $\mathbf{x}_t$ only depends on the state of the previous time step. Plugging (B.3) and (B.4) into (B.2) yields

$$w_t^n \propto \frac{p(\mathbf{y}_t|\mathbf{s}_t^n)p(\mathbf{s}_t^n|\mathbf{s}_{t-1}^n)p(\mathbf{s}_{t-1}^n|\mathcal{Y}_{t-1})}{q(\mathbf{s}_t^n|\mathbf{s}_{t-1}^n,\mathcal{Y}_t)q(\mathbf{s}_{t-1}^n|\mathcal{Y}_{t-1})}. \tag{B.5}$$

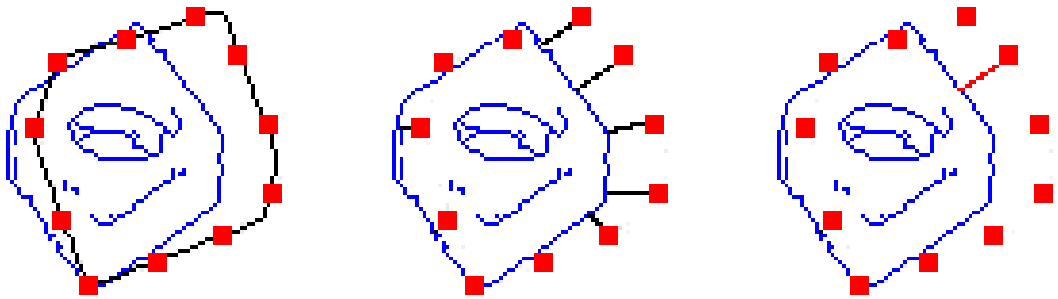
Under the additional assumption that the importance density $q(\cdot)$ only depends on the latest image measurement $\mathbf{y}_t$ instead of the whole history, this can finally be regrouped to the recursive weight update scheme

$$w_t^n \propto w_{t-1}^n \frac{p(\mathbf{y}_t|\mathbf{s}_t^n)p(\mathbf{s}_t^n|\mathbf{s}_{t-1}^n)}{q(\mathbf{s}_t^n|\mathbf{s}_{t-1}^n,\mathbf{y}_t)}. \tag{B.6}$$

C Image Cues for Assembly Pose Estimation

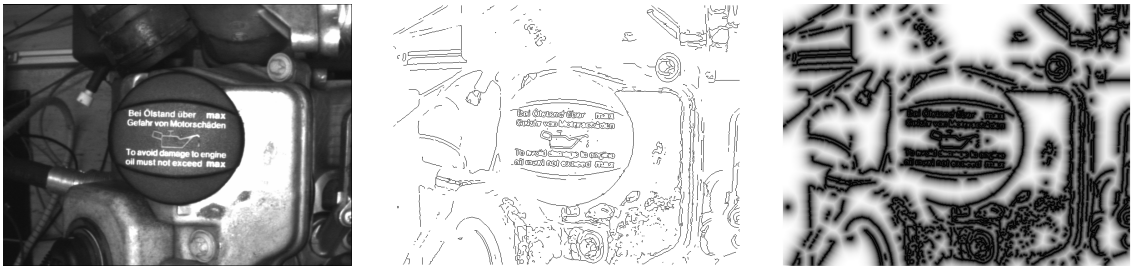
All particle filters that are described in this thesis approximate the observation density at specific particle positions. As detailed in 4.2.2, this approximation is based on the evaluation of different image cues. For the inspection system prototype, three cues were implemented, namely the forward distance cue, the backward distance cue, and the color cue. They are described in the following.

The *forward distance cue* is based on the partial directed Hausdorff distance (cf. [Ruc96, p.38]). The latter rates the distance between two point sets. In our case, the first point set is $\mathbf{z}_t^k$ which results from sampling 2D points along the projected features of the kth assembly part model, given a specific particle that is interpreted as pose hypothesis. The second point set is the set of edge pixels $E(\mathbf{y}_t)$ that results from the application of an edge detection filter to the current image measurement. Both point sets are illustrated in Fig. C.1(a). The forward distance cue is evaluated by first establishing the distance of each point in $\mathbf{z}_t^k$ to the nearest edge pixel as illustrated in Fig. C.1(b). The largest such distance yields the *directed Hausdorff distance*.



(a) (b) (c)

Figure C.1: The directed Hausdorff distance. a) Point set $\mathbf{z}_t^k$ (red) is sampled from the part model features (black) of a nut whose pose is hypothesized according to a given particle. The edge pixels $E(\mathbf{y}_t)$ (blue) arise from the physical object. b) A line associates each $\mathbf{z} \in \mathbf{z}_t^k$ with the nearest edge pixel. c) The largest distance (red line) is the directed Hausdorff distance



(a) (b) (c)

Figure C.2: The effects of the chamfer distance transform. The image a) is treated with a SUSAN edge detection filter [SB97]. The resulting edge image b) then undergoes a chamfer distance transform. Within the chamfer distance image c), dark pixels denote positions that are close to edge pixels in the untransformed edge image. Image a) courtesy of DaimlerChrysler AG

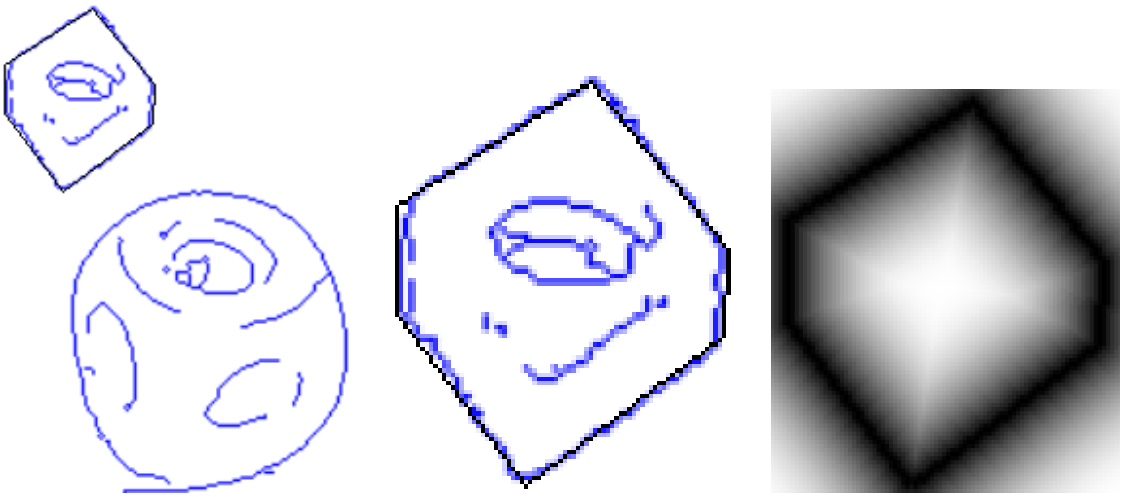
In order to compensate for outliers due to image noise, one can rank the shortest distances between the point sets and choose the l-th quantile value among them. For example, the 0-th quantile value would select the smallest of all ranked values and the $\frac{1}{2}$-th quantile value their median, while the 1-th quantile value yields the largest distance as chosen in Fig. C.1(c). For $0 \leq l \leq 1$, this procedure returns the *partial directed Hausdorff distance*

$$h(\mathbf{z}_t^k, E(\mathbf{y}_t)) = l^{\text{th}}_{\mathbf{z} \in \mathbf{z}_t^k} \min_{\mathbf{e} \in E(\mathbf{y}_t)} \|\mathbf{z} - \mathbf{e}\|. \tag{C.1}$$

Note that $\|\cdot\|$ within this thesis denotes the Euclidean distance or an approximation thereof. The $\min \|\cdot\|$ operation can be implemented as a simple look-up, if the edge image $E(\mathbf{y}_t)$ is filtered with a *chamfer distance transform*. The latter is an integer approximation to the Euclidean distance of any point on the image grid to the nearest edge pixel. An efficient algorithm to compute this transform has e.g. been proposed by Borgefors [Bor86]. Its outcome is illustrated in Fig. C.2(c). At each 2D position $\mathbf{z}$, one can now read out the corresponding chamfer distance value in order to obtain the shortest distance to the next image edge pixel.

The forward cue can be normalized to $[0, 1]$ by taking into account that the chamfer distance transform operates on images of finite dimensions. Thus, a value $D_{\max}$ exists that denotes the largest possible distance of any image point to an edge pixel within the same image. We define the forward distance cue as the normalized partial directed Hausdorff distance

$$f_{\text{fw}}(\mathbf{z}_t^k, \mathbf{y}_t) = \frac{1}{D_{\max}} h(\mathbf{z}_t^k, E(\mathbf{y}_t)). \tag{C.2}$$



(a) (b) (c)

Figure C.3: Clipping image regions for the backward distance cue. a) The edge image of an observed nut and cube (blue) is overlaid with the projected model features of an hypothesized nut model (black). b) Clipped region that fully contains the projected model features. c) The projected model features have undergone a chamfer distance transformation

The *backward distance cue* is similar to the forward distance cue but operates in the reverse direction. While the forward distance cue rates how near model feature sample points are to image edge pixels, the backward distance cue indicates how close image edge pixels are to predicted part model features. Note that, for two point sets A and B to be nearly identical, both forward and backward distance cue would have to close to 0. However, the comparison of all image edge pixels to part feature points is problematic in our case because the feature points arise from only one assembly part. Accordingly, they usually account for a small part of the observed edge pixels. Those edge pixels arising from other structures in the image must be excluded from consideration, or they would bias the backward distance. This is why, as illustrated in Fig. C.3(b), the edge image is clipped to a rectangular region that is tightly bounding the model feature points. If $C(E(\mathbf{y}_t))$ denotes such a clipped image region, the backward distance cue is expressed as

$$f_{\mathrm{bw}}(\mathbf{z}_t^k, \mathbf{y}_t) = \frac{1}{D_{\max}} l^{\mathrm{th}}_{\mathbf{e}\in C(E(\mathbf{y}_t))} \min_{\mathbf{z}\in\mathbf{z}_t^k} \|\mathbf{z}-\mathbf{e}\|. \tag{C.3}$$

Analogue to the forward distance cue, the predicted model feature points within the clipped region undergo a chamfer distance transform as illustrated in Fig. C.3(c). Afterwards, the $\min\|\cdot\|$ operation can again be realized by a fast look-up. However, the backward cue is computationally much more expensive than the forward cue. The reason

for this is that the latter depends on a distance transform of $E(\mathbf{y}_t)$ which must only be calculated once for any new image measurement. In contrast to this, the backward cue employs the distance transform of predicted part features, which must be computed for each new part pose hypothesis.

The *color cue* evaluates the mean color of polygonal image patches. For this, the assembly model must be annotated not only with contour edge features but with polygonal surface regions of a certain uniform color. In the prototype system of this thesis, such an annotation can only be carried out manually because the employed CAD models don't contain any color information. The polygons can then be decomposed to straight line segments and treated with the same visibility prediction, transformation, and projection concepts as the contour edge features. Samples $\mathbf{z}_t^k$ are positioned within the image regions that are enclosed by projected visible polygons. Given that each color region of the kth assembly part is assigned a mean color $\overline{Col}^{\,k}$ in the course of some system calibration procedure, the mean color cue of that part is evaluated as the Euclidean distance

$$f_{\mathrm{col}}(\mathbf{z}_t^k, \mathbf{y}_t) = \frac{1}{N_{\mathrm{z}}} \sqrt{\sum_{\mathbf{z} \in \mathbf{z}_t^k} (Col(\mathbf{z}, \mathbf{y}_t) - \overline{Col}^{\,k})^2}, \tag{C.4}$$

where $Col(\mathbf{z}, \mathbf{y}_t)$ denotes the color within the current image measurement at sample point $\mathbf{z}$ and $N_{\mathrm{z}} = card(\mathbf{z}_t^k)$ is the number of 2D sample points. The prototype implementation represents colors in the uv subspace of the yuv color space.

D Publication List

- D. Stößel, M. Hanheide, G. Sagerer, L. Krüger, and M. M. Ellenrieder. Feature and Viewpoint Selection for Industrial Car Assembly. In *DAGM 2004*, volume 3175 of *Lecture Notes in Computer Science*, pages 528–535, 2004.

- M. M. Ellenrieder, L. Krüger, D. Stößel, and M. Hanheide. A Versatile Model-Based Visibility Measure for Geometric Primitives. In H. Kalviainen, J. Parkkinen, and A. Kaarna, editors, *SCIA 2005*, volume 3540 of *LNCS*, pages 669–678, Heidelberg, Germany, 2005. Springer.

- D. Stößel and G. Sagerer. Kernel Particle Filter for Visual Quality Inspection from Monocular Intensity Images. In *DAGM 2006*, volume 4174 of *Lecture Notes in Computer Science*, pages 597–606, 2006.

Bibliography

[ACB96] H. Araújo, R. L. Carceroni, and C. M. Brown. A Fully Projective Formulation for Lowe's Tracking Algorithm. Technical report, University of Rochester, November 1996.

[AK89] J. Arvo and D. Kirk. A Survey of Ray Tracing Acceleration Techniques. In A. S. Glassner, editor, *An Introduction to Ray Tracing*, pages 201–261. Academic Press, London, 1989.

[AKSA05] S. Ando, Y. Kusachi, A. Suzuki, and K. Arakawa. Appearance Based Pose Estimation of 3D Object Using Support Vector Regression. In *IEEE International Conference on Image Processing*, volume 1, pages 341–344, September 2005.

[AMGC02] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on Particle Filters for On-line Non-Linear/Non-Gaussian Bayesian Tracking. *IEEE Trans. on Signal Processing*, 50(2):174–188, 2002.

[BA83] A. B. Badiru and B. J. Ayeni. *Practitioners Guide to Quality and Process Improvement*. Chapman and Hall, London, 1983.

[BA98] J. H. M. Byne and J. A. D. W. Anderson. A CAD-Based Computer Vision System. *Image and Vision Computing*, 16:533–539, 1998.

[Bal81] D. H. Ballard. Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[Bas93] R. Basri. Viewer-Centered Representations in Object Recognition. In C. H. Chen, L. F. Pau, and P. S. P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, pages 863–882. World Scientific, 1993.

[Bau02] C. Bauckhage. *A Structural Framework for Assembly Modeling and Recognition*. PhD thesis, University of Bielefeld, 2002.

[BB82] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, 1982.

[BDKS04] B. Bank, G. Diubin, A. Korbut, and I. Sigal. The Average Behaviour of Greedy Algorithms for the Knapsack Problem: Computational Experiments. Preprints aus dem Institut für Mathematik 6, Humboldt Universität, Berlin, 2004. ISSN: 0863-0976.

[Bec98] J. Beckford. *Quality: a critical introduction*. Routledge, New York, 1998.

[BFF+06] C. Bauckhage, G. A. Fink, J. Fritsch, N. Jungclaus, S. Kronenberg, F. Kummert, F. Lömker, G. Sagerer, and S. Wachsmuth. *Situated Communication*, chapter Integrated perception for cooperative human-machine interaction, pages p.325–356. Trends in linguistics. Mouton de Gruyter, Berlin, 2006.

[BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Monterey, CA: Wadsworth, 1984.

[BHH83] R. C. Bolles, P. Horaud, and M. J. Hannah. 3DPO: A Three-Dimensional Part Orientation System. In *Proc. of the 8th International Joint Conf. on Artificial Intelligence*, pages 1116–1120, Karlsruhe, West Germany, 1983.

[BM98] C. Bregler and J. Malik. Tracking People with Twists and Exponential Maps. In *CVPR 1998*, pages 8–15, Santa Barbara, California, 1998. IEEE.

[BMP77] L. Breiman, W. Meisel, and E. Purcell. Variable Kernel Estimates of Multivariate Densities. *Technometrics*, 19:135–144, 1977.

[Bor86] G. Borgefors. Distance Transformation in Digital Images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[Bou84] A. Bourjault. *Contribution à une approche méthodologique de l'Assemblage Automatisé: Elaboration Automatique des Séquences Opératoires*. PhD thesis, Université de Franche-Comté, 1984.

[Bro66] D. C. Brown. Decentering Distortion of Lenses. *Photogrammetric Engineering*, 32(3):444–462, 1966.

[Bro81] R. A. Brooks. Symbolic Reasoning Among 3-D Models and 2-D Images. *Artificial Intelligence*, 17:285–348, 1981.

[Bro83] R. A. Brooks. Model-Based Three-Dimensional Interpretations of Two-Dimensional Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):140–150, March 1983.

[BW91] A. Beinglass and H. J. Wolfson. Articulated Object Recognition, or to Generalize the Generalized Hough Transform. In *Proc. of the IEEE Computer Vision and Pattern Recognition Conference*, pages 461–466, 1991.

[CA03] C. Chang and R. Ansari. Kernel Particle Filter: Iterative Sampling for Efficient Visual Tracking. In *ICIP 2003*, pages 977–980. IEEE, 2003.

[CA05] C. Chang and R. Ansari. Kernel Particle Filter for Visual Tracking. *IEEE Signal Processing Letters*, 12(3):242–245, March 2005.

[Can86] J. F. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[CB89] C. K. Cowan and A. Bergman. Determining the Camera and Light-Source Location for a Visual Task. In *ICRA 1989*, volume 1, pages 509–514, Scottsdale, Arizona, USA, May 1989.

[CH67] T. M. Cover and P. E. Hart. Nearest Neighbour Pattern Classification. *IEEE Transactions on Information Theory*, 13:21–27, 1967.

[Che91] H. Chen. Pose Determination from Line-to-Plane Correspondances: Existence Condition and Closed-Form Solutions. *IEEE Transaction on Pattern Analysis And Machine Intelligence*, 13(6):530–541, 1991.

[CL02] S. Y. Chen and Y. F. Li. A Method of Automatic Sensor Placement for Robot Vision in Inspection Tasks. In *ICRA 2002*, pages 2545–2550, Washington, DC, May 2002.

[CM99] D. Comaniciu and P. Meer. Mean Shift Analysis and Applications. In *Proceedings of the International Conference on Computer Vision*, pages 1197–1203, September 1999.

[CM02] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[Cow91] C. K. Cowan. Automatic Camera and Light-Source Placement Using CAD Models. In *IEEE Workshop on Directions in Automated CAD-Based Vision*, pages 22–31, Maui, Hawaii, June 1991.

[Cra89] J. J. Craig. *Introduction To Robotics: Mechanics and Control*. Addison-Wesley, 2nd edition, 1989.

[CRM01] D. Comaniciu, V. Ramesh, and P. Meer. The Variable Bandwidth Mean Shift and Data-Driven Scale Selection. In *IEEE International Conference on Computer Vision*, volume 1, pages 438–445, 2001.

[CSH91] O. I. Camps, L. G. Shapiro, and R. M. Haralick. PREMIO: An Overview. In *IEEE Workshop on Directions in Automated CAD-Based Vision*, pages 11–21, Maui, Hawaii, June 1991.

[DBNB99] J. Deutscher, A. Blake, B. North, and B. Bascle. Tracking Through Singularities and Discontinuities by Random Sampling. In *Proc. 7th Int. Conf. on Computer Vision*, volume 2, pages 1144–1149, 1999.

[DBR00] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 1111–1119, 2000.

[DC00] T. W. Drummond and R. Cipolla. Real-Time Tracking of multiple Articulated Structures in Multiple Views. In *European Conference on Computer Vision*, pages 20–36, 2000.

[DD95] D. DeMenthon and L. S. Davis. Model-Based Object Pose in 25 Lines of Code. *International Journal of Computer Vision*, 15:123–141, 1995.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

[DK80] P. A. Devijver and J. Kittler. On the Edited Nearest Neighbour Rule. In *Proc. 5th Int. Conf. on Pattern Recognition*, pages 72–80, Miami, Florida, 1980.

[dMS90] L. S. Homem de Mello and A. C. Sanderson. AND/OR Graph Representation of Assembly Plans. *IEEE Transactions on Robotics and Automation*, 6(2):188–199, 1990.

[DPR92] S. J. Dickinson, A. P. Pentland, and A. Rosenfeld. 3-D Shape Recovery Using Distributed Aspect Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):174–198, 1992.

[DW88] X. Dong and M. Wozny. FRAFES - a Frame-Based Feature Extraction System. In *International Conference on Computer Integrated Manufacturing*, pages 296–305, Troy, NY, 1988. IEEE.

[EKH05] S. Ekvall, D. Kragic, and F. Hoffmann. Object Recognition and Pose Estimation using Color Cooccurrence Histograms and Geometric Modeling. *Image and Vision Computing*, 23(11):943–955, 2005. October.

[EKSH05] M. M. Ellenrieder, L. Krüger, D. Stößel, and M. Hanheide. A Versatile Model-Based Visibility Measure for Geometric Primitives. In H. Kalviainen, J. Parkkinen, and A. Kaarna, editors, *SCIA 2005*, volume 3540 of *LNCS*, pages 669–678, Heidelberg, Germany, 2005. Springer.

[Ell05] M. M. Ellenrieder. *Optimal Viewpoint Selection for Industrial Machine Vision and Inspection of Flexible Objects*. PhD thesis, University of Bielefeld, July 2005.

[FB81] M. A. Fischler and R. C. Bolles. RANdom SAmple Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Commun. of the ACM*, 24(6):381–395, 1981.

[FH75] K. Fukunaga and L. D. Hostetler. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. *IEEE Transactions on Information Theory*, 21:32–40, 1975.

[Fri03] J. N. Fritsch. *Vision-Based Recognition of Gestures With Context*. PhD thesis, Bielefeld University, March 2003.

[Gav00] D. M. Gavrila. Pedestrain Detection from a Moving Vehicle. In *Proc. 6th European Conference on Computer Vision*, volume 2, pages 37–49, Dublin, Ireland, 2000.

[GBCS00] N. Giordana, P. Boutherny, F. Chaumette, and F. Spindler. Two-Dimensional Model-Based Tracking of Complex Shapes for Visual Servoing Tasks. In M. Vincenze and G. Hager, editors, *Robust Vision for Vision-Based Control of Motion*, pages 67–77. IEEE Press, 2000.

[GD96] D. M. Gavrila and L. Davis. 3D Model-Based Tracking of Humans in Action: A Multi-View Approach. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[Gis03] F. J. Goerlich Gisbert. Weighted Samples, Kernel Density Estimators and Convergence. *Empirical Economics*, 28:335–351, 2003.

[GJW94] B. K. Gosh, M. Jankovic, and Y. T. Wu. Perspective Problems in System Theory and its Application to Machine Vision. *Journal of Mathematical Systems, Estimation and Control*, 4(1):3–38, 1994.

[GLM96] S. Gottschalk, M. C. Lin, and D. Manocha. OBB-Tree: A Hierarchical Structure for Rapid Interference Detection. In *Proc. ACM SIGGRAPH*, volume 30, pages 171–180, August 1996.

[GLPH90] W. E. L. Grimson, T. Lozano-Pérez, and D. P. Huttenlocher. *Object Recognition by Computer: The Role of Geometric constraints*. MIT Press, Cambridge, Massachusetts, 1990.

[Goa83] C. Goad. Special Purpose Automatic Programming for 3-D Model-Based Vision. In *Proc. of DARPA Image Understanding Workshop*, pages 94–104, 1983.

[Goa86] C. Goad. Fast 3D Model-Based Vision. In A. P. Pentland, editor, *From Pixels to Predicates*, Ablex Series in Artificial Intelligence, pages 371–391. Ablex, 1986.

[God97] J. S. Goddard. *Pose and Motion Estimation from Vision using Dual Quaternion-Based Extended Kalman Filtering*. PhD thesis, University of Tennessee, Knoxville, December 1997.

[Han01] M. A. Hanheide. Objektbezogene 3D-Erkennung automatisch generierter Konturmodelle in Intensitätsbildern. Master's thesis, University of Bielefeld, July 2001.

[HB86] P. Horaud and R. C. Bolles. 3DPO: A System for Matching 3-D Objects in Range Data. In A. P. Pentland, editor, *From Pixels to Predicates*, Ablex Series in Artificial Intelligence, pages 359–370. Ablex, 1986.

[HCG90] M. R. Henderson, S. H. Chuang, and G. P. Gavankar. Graph-Based Feature Extraction. In *Proceedings of NSF Design and Manufacturing Systems Conference*, pages 183–189, Tempe, AZ, 1990.

[HEG+91] T. C. Henderson, J. Evans, L. Grayston, A. Sanderson, L. Stoller, and E. Weitz. CBCV: A CAD-Based Computer Vision System. In *IEEE Workshop on Directions in Automated CAD-Based Vision*, pages 11–21, Maui, Hawaii, June 1991.

[Hen84] M. R. Henderson. *Extraction of Feature Information from Three Dimensional CAD Data*. PhD thesis, Purdue University, 1984.

[HJ88] R. M. Haralick and H. Joo. 2D-3D Pose Estimation. In *ICPR 1988*, pages 385–391, 1988.

[HKT89] R. Hoffman, H. R. Keshavan, and F. Towfiq. CAD-Driven Machine Vision. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1477–1488, November/December 1989.

[HLZ97] A. Hauck, S. Lanser, and C. Zierl. Hierarchical Recognition of Articulated Objects from Single Perspective Views. In *CVPR 1997*, pages 870–876, Puerto Rico, 1997. IEEE.

[Hom91] H. C. Homer. Pose Determination from Line-To-Plane Correspondences: Existence Condition and Closed-Form Solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):530–541, 1991.

[HOW96] Y. Hel-Or and M. Werman. Constraint Fusion for Recognition and Localization of Articulated Objects. *International Journal of Computer Vision*, 19(1):5–28, 1996.

[HS96] A. Hauck and N. O. Stöffler. Video-Based Determination of the Joint States of Articulated Objects. In *Int. Conf. on Robotics, Vision and Parallel Processing for Industrial Automation*, pages 1018–1023, Ipoh, Malaysia, 1996.

[HU86] D. P. Huttenlocher and S. Ullman. Object Recognition using Alignment. In *Proceedings of the 1st Internation Conference on Computer Vision (ICCV)*, pages 102–111. IEEE, 1986.

[HZ03] R. Hartley and A. Zisserman. *Multiple View Gometry*. Cambridge University Press, second edition, 2003.

[IB98a] M. Isard and A. Blake. Condensation: Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(3), 1998.

[IB98b] M. Isard and A. Blake. ICONDENSATION: Unifying Low-Level and High-Level Tracking in a Stochastic Framework. In *Proc. European Conference on Computer Vision (ECCV)*, volume 1406 of *LNCS*, pages 893–909, 1998.

[Imm05] S. Immen, editor. *Jahresbericht 2005*. Kraftfahrt-Bundesamt, 2005.

[Jak82] R. Jakubowski. Syntactic Characterization of Machine-Parts Shapes. *Cybern. Syst. Int. J.*, 13:1–24, 1982.

[JC88] S. Joshi and T. C. Chang. Graph-Based Heuristic for Recognition of Machined Features from a Solid 3D Model. *Computer-Aided Design*, 20(2):58–66, 1988.

[JM97] Q. Ji and M. M. Marefat. Machine Interpretation of CAD Data for Manufacturing Applications. *ACM Computing Surveys*, 24(3):264–311, September 1997.

[KBG97] H. Klingspohr, T. Block, and R.-R. Grigat. A Passive Real-Time Gaze Estimation System for Human-Machine Interfaces. In G. Sommer, K. Daniilidis, and J. Pauli, editors, *Computer Analysis of Images and Patterns (CAIP)*, volume 1296 of *LNCS*, pages 718–725, 1997.

[KDN93] D. Koller, K. Daniilidis, and H.-H. Nagel. Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993.

[KLW94] A. Kong, J. S. Liu, and W. H. Wong. Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association*, 89:278–288, 1994.

[KMTB94] K. W. Khawaja, A. A. Maciejewski, D. Tretter, and C. Bouman. Automated Assembly Inspection Using a Multiscale Algorithm Trained on Synthetic Images. In *ICRA 1994*, volume 4, pages 3530–3536. IEEE, 1994.

[Köl02] T. Kölzow. *System zur Klassifikation und Lokalisation von 3D-Objekten durch Anpassung vereinheitlichter Merkmale in Bildfolgen*. PhD thesis, University of Bielefeld, October 2002. in German.

[KVP92] D. J. Kriegman, B. Vijayakumar, and J. Ponce. Constraints for Recognizing and Locating Curved 3D Objects from Monocular Image Features. In G. Sandini, editor, *Proc. European Conference on Computer Vision (ECCV)*, volume 588 of *LNCS*, pages 829–833, 1992.

[KW97] L. Kettner and E. Welzl. Contour Edge Analysis for Polyhedron Projections. In W. Strasser, R. Klein, and R. Rau, editors, *Geometric Modeling: Theory and Practice*, pages 379–394. Springer, 1997.

[LHM00] C.-P. Lu, G. D. Hager, and E. Mjolsness. Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000.

[Li96] R. K. Li. *A Conceptual Framework for the Interaction of Computer-Aided Design and Computer-Aided Manufacturing*. PhD thesis, Arizona State University, 1996.

[Low87] D. G. Lowe. Three-Dimensional Object Recognition from Single Two-Dimensional Images. *Artificial Intelligence*, 31(3):355–395, 1987.

[Low89] D. G. Lowe. Fitting Parameterized 3-D Models to Images. Technical Report 89-26, Dept. of Computer Science, University of British Columbia, 1989.

[Low04] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–104, 2004.

[LTCP02] A. J. Lacey, N. A. Thacker, P. Courtney, and S. B. Pollard. *TINA 2001: The Closed Loop 3D Model Matcher*. Imaging Science and Biomedial Engineering Division, Medical School, University of Manchester, February 2002.

[Luo93] Q.-T. Luong. Color in Computer Vision. In C. H. Chen, L. F. Pau, and P. S. P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, pages 311–368. World Scientific, 1993.

[LW88] Y. Lamdan and H. Wolfson. Geometric Hashing: A General and Efficient Model-Based Recognition Scheme. In *Proc. of the IEEE Computer Vision and Pattern Recognition Conference*, pages 238–249, 1988.

[MI00] J. MacCormick and M. Isard. Partitioned Sampling, Articulated Objects, and Interface-Quality Hand Tracking. In *Proc. of the European Conference on Computer Vision*, volume 2, pages 3–19, 2000.

[MN95] H. Murase and S. K. Nayar. Visual Learning and Recognition of 3-D Objects from Appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[MT90] S. Martello and P. Toth. *Knapsack Problems. Algorithms and Computer Implementation*. Chichester: Wiley, 1990.

[Nie83] H. Niemann. *Klassifikation von Mustern*. Springer, 1983.

[OH97] C. F. Olsen and D. P. Huttenlocher. Automatic Target Recognition by Matching Oriented Edge Pixels. *IEEE Transactions in Image Processing*, 6(1):103–113, 1997.

[Per78] W. A. Perkins. A Model-Based Vision System for Industrial Parts. *IEEE Transactions on Computers*, C-27(2):126–143, February 1978.

[Pet03] G. Peters. Efficient Pose Estimation Using View-Based Object Representations. In J. L. Crowley et al., editor, *ICVS 2003*, volume 2626 of *LNCS*, pages 12–21, Berlin Heidelberg, 2003. Springer.

[PHYP93] T. Q. Phong, R. Horaud, A. Yassine, and D. T. Pham. Optimal Estimation of Object Pose from a Single Perspective View. In *International Conference on Computer Vision*, pages 534–539, February 1993.

[Pop94] A. Pope. Model-Based Object Recognition - A Survey of Recent Research. Technical report, University of British Columbia, January 1994.

[Qui93] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, 1993.

[Ris01] I. Rish. An Empirical Study of the Naive Bayes Classifier. In *IJCAI Workshop on Empirical Methods in Artificial Intelligence*, 2001.

[RKRS01] B. Rosenhahn, N. Krüger, T. Rabsch, and G. Sommer. Tracking with a Novel Pose Estimation Algorithm. In R. Klette, S. Peleg, and G. Sommer, editors, *Robot Visio 2001*, volume 1998 of *LNCS*, pages 9–18, Berlin Heidelberg, February 2001. Springer.

[Rob65] L. G. Roberts. Machine Perception of Three-Dimensional Solids. In J. Tippett, D. Berkowitz, L. Clapp, C. Koester, and A. Vanderburgh, editors, *Optical and Electro-Optical Information Processing*, chapter 9, pages 159–197. MIT Press, 1965.

[Ros03] B. Rosenhahn. *Pose Estimation Revisited*. PhD thesis, Christian-Albrechts-Universität Kiel, April 2003.

[RP96] M. Rabemanantsoa and S. Pierre. An Artifcial Intelligence Approach for Generating Assembly Sequences in CAD/CAM. *Artifcial Intelligence in Engineering*, 10(2):97–107, 1996.

[Ruc96] W. Rucklidge. *Efficient Visual Recognition Using the Hausdorff Distance*, volume 1173 of *LNCS*. Springer, 1996.

[RW91] A. A. G. Requicha and T. W. Whalen. Representations for Assemblies. In L. S. Homem de Mello and S. Lee, editors, *Computer-aided mechanical assembly planning*, pages 15–39. Kluwer Academic Press, 1991.

[SB90] M. Swain and D. Ballard. Indexing via Color Histograms. In *ICVS 1990*, pages 390–393, Osaka, Japan, 1990. Springer.

[SB97] S. M. Smith and J. M. Brady. SUSAN - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, May 1997.

[SB01] H. Sidenbladh and M. J. Black. Learning Image Statistics for Bayesian Tracking. In *Proc. Int. Conf. on Computer Vision*, volume II, pages 709–716. IEEE, 2001.

[SGPK98] R. Sinha, S. K. Gupta, C. J. J. Paredis, and P. K. Khosla. Capturing Articulation in Assemblies from Component Geometry. In *Proceedings of DETC'98*, Atlanta, Georgia, USA, September 1998. ASME.

[Shi75] Y. Shirai. Edge Finding, Segmentation of Edges and Recognition of Complex Objects. In *IJCAI 1975*, pages 674–681, Tbilisi, Georgia, USSR, 1975.

[Shi78] Y. Shirai. Recognition of Man-Made Objects using Edge Cues. In A. Hanson and E. Riseman, editors, *Computer Vision Systems*. Academic, New York, 1978.

[Shi86] S. Shingo. *Zero Quality Control: Source Inspection and the Poka-Yoke System*. Productivity Press, 1986.

[SHS+04] D. Stößel, M. Hanheide, G. Sagerer, L. Krüger, and M. M. Ellenrieder. Feature and Viewpoint Selection for Industrial Car Assembly. In *DAGM 2004*, volume 3175 of *Lecture Notes in Computer Science*, pages 528–535, 2004.

[Sil86] B.W. Silverman. *Density Estimation for Statistics and Data Analysis*, volume 26 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 1986.

[SKF06] J. Schmidt, B. Kwolek, and J. Fritsch. Kernel Particle Filter for Real-Time 3D Body Tracking in Monocular Color Images. In *Proc. of Automatic Face and Gesture Recognition*, pages 567–572, Southampton, UK, April 2006. IEEE.

[Soc97] G. Socher. *Qualitative Scene Descriptions from Images for Integrated Speech and Image Understanding*. PhD thesis, University of Bielefeld, June 1997.

[SRTBS91] R. Safaee-Rad, I. Tchoukanov, B. Benhabib, and K. C. Smith. Accurate Parameter Estimation of Quadratic Curves from Grey-Level Images. *CVGIP: Image Understanding*, 54(2):259–274, September 1991.

[SS04] D.W. Scott and S.R. Sain. *Data Mining and Computational Statistics*, volume 23, chapter Multi-Dimensional Density Estimation. Elsevier, 2004.

[SS06] D. Stößel and G. Sagerer. Kernel Particle Filter for Visual Quality Inspection from Monocular Intensity Images. In *DAGM 2006*, volume 4174 of *Lecture Notes in Computer Science*, pages 597–606, 2006.

[SVD03] G. Shakhnarovich, P. Viola, and T. Darrel. Fast Pose Estimation With Parameter Sensitive Hashing. In *IEEE International Conference on Computer Vision*, volume 2, pages 750–757, October 2003.

[Szc58] W. Szczepanski. *Die Lösungsvorschläge für den räumlichen Rückwärtseinschnitt*. PhD thesis, Deutsche Geodätische Kommission, 1958. Reihe C: Dissertationen – Heft Nr. 29.

[Tay00] C. J. Taylor. Reconstruction of Articulated Objects from Point Correspondences in a Single Uncalibrated Image. *Computer Vision and Image Understanding*, 80(3):349–363, December 2000.

[TvBDK00] D. M. J. Tax, M. van Breukelen, R. P. W. Duin, and J. Kittler. Combining Multiple Classifiers by Averaging or by Multiplying? *Pattern Recognition*, 33(9):1475–1485, 2000.

[vBGW03] C. v. Bank, D. Gavrila, and C. Wöhler. A Visual Quality Inspection System Based on a Hierarchical 3D Pose Estimation algorithm. In B. Michaelis and G. Krell, editors, *DAGM 2003*, volume 2781 of *LNCS*, pages 179–186. Springer, 2003.

[WB89] L. Wixson and D. Ballard. Color Histograms for Real-Time Object Search. In *Proc. SPIE Sensor Fusion II: Human and Machine Strategies Workshop*, pages 435–446, Philadelphia, PA, 1989.

[Woo82] T. C. Woo. Feature Extraction by Volume Decomposition. In *Proceedings of Conference on CAD/CAM Technology in Mechanical Engineering*, pages 76–94, MIT, Cambridge, MA, 1982.

[WWH97] S. Winkler, P. Wunsch, and G. Hirzinger. A Feature Map Approach to Real-Time 3D Object Pose Estimation from Single 2D Perspective Views. In E. Paulus and F. M. Wahl, editors, *Mustererkennung 1997*, Informatik aktuell, pages 129–136. Springer, Heidelberg, 1997.

[YMK94] C. C. Yang, M. M. Marefat, and R. L. Kashyap. Active Visual Inspection Based on CAD Models. In *ICRA 1994*, pages 1120–1125, San Diego, CA, May 1994.

[ZN96] M. Zerroug and R. Nevatia. Pose Estimation of Multi-Part Curved Objects. *Image Understanding Workshop (IUW)*, pages 831–835, 1996.