
Model Based Object Classification and Localisation in Multicocular Images

Lars Krüger
November 2007

Dissertation

Abdruck der genehmigten Dissertation zur Erlangung des akademischen Grades *Doktor der Ingenieurwissenschaften (Dr.-Ing.)* an der Technischen Fakultät der Universität Bielefeld.

Dipl.-Inf. Lars Krüger
email: lars_e.krueger@gmx.de

Gutachter:
Prof. Dr. Gerhard Sagerer, Universität Bielefeld
Prof. Dr. Rainer Ott, Universität Stuttgart

Prüfungsausschuß:
Prof. Dr. Jens Stoye, Universität Bielefeld
Prof. Dr. Gerhard Sagerer, Universität Bielefeld
Prof. Dr. Rainer Ott, Universität Stuttgart
Dr. Marc Hanheide, Universität Bielefeld

Gedruckt auf alterungsbeständigem, holz- und säurefreiem Papier nach DIN-ISO 9706.
Printed on non-aging, woodfree, acidfree paper according to DIN-ISO 9706.

Acknowledgement

I would like to thank my parents, Karin and Horst Krüger, for their love and support.

This thesis was instigated by my advisor Prof Dr Ott, who kept encouraging me for about half a year, before I was finally convinced. Dr Ott also provided me with most valuable feedback, resulting in a clearer presentation of this thesis. Thank you.

Prof Dr Sagerer, my doctoral advisor, introduced an outside view on the topic. His suggestions are appreciated.

My office mate, Dr Marc Ellenrieder, cheerfully endured me for more than four years, both at work and sometimes even in his home. We had many lively discussion about many different subjects, which I cherish as fun time.

Dr Christian Wöhler worked with me on various projects, great and small. He taught me to aim for the scientific merits of a project, beyond the task at hand.

Last but not least, I would like to thank Dr Pablo d'Angelo for discussions on photography, programming, and Open Source software; Mr Kia Hafezi for his help in various projects; Mr Frank Lindner for barbecue invitations, finding the simple solution, and condemning C++; and my other colleagues at the DaimlerChrysler Research Centre for allowing me to ask them questions at any time.

Contents

I Introduction

1 Motivation	1
1.1 Context	1
1.2 State of the Art	2
2 About This Thesis	17
2.1 Aim and Scope	17
2.2 Data and Methods Used	19
2.3 Notational Conventions	19
2.4 Section Overview	20

II The Multicocular Object Recognition System

3 Methodical Framework	22
3.1 System Overview	22
3.2 Camera Calibration	24
4 Appearance Based Methods	53
4.1 Multicocular Template Matching	53
4.2 Feature Pose Maps	64
5 Segmentation Methods	68
5.1 Multicocular Active Contours	68
5.2 The Contracting Curve Density Algorithm	71
6 Closest Point Methods	76
6.1 Object Recognition using Characteristic Local Features	76

6.2	Object Recognition using Gradient Sign Tables	79
-----	---	----

III Experiments and Applications

7	Evaluating the Camera Calibration	89
7.1	Goal of Investigation	89
7.2	Experimental Setup	89
7.3	Pose Repeatability	90
7.4	Calibration Repeatability	93
7.5	Choice of External Calibration Algorithm	101
7.6	Summary	104
8	Oil Cap Inspection	105
8.1	Goal of Investigation	105
8.2	Experimental Setup	105
8.3	Pose Estimation Accuracy	109
8.4	Object Classification Performance	126
8.5	Summary	134
9	Obtaining the Trajectory of Tubes and Cables	135
9.1	Goal of Investigation	135
9.2	Experimental Setup	136
9.3	Accuracy of Trajectory Estimation	140
9.4	Summary	145

IV Closing Remarks

10	Summary	146
10.1	A-Priori Questions Answered	146
10.2	Additional Insights	147
11	Outlook	149
11.1	Calibration	149
11.2	Template Matching	149
11.3	Feature Pose Maps	150

11.4 Gradient Sign Tables	150
11.5 Trajectory of Tubes and Cables	150
11.6 Other Issues	151

V Appendix

A Definitions	152
A.1 Image Tuple	152
A.2 Object Recognition Algorithms	152
A.3 Camera Identifier	153
A.4 Mesh, Facet, Vertex	153
A.5 Appearance Based Object Recognition	153
A.6 Segmentation Based Object Recognition Algorithm	153
A.7 Constraint, Soft Constraint, Hard Constraint	154
A.8 Distance Transform	154
A.9 Epipolar Line	154
A.10 Run Length Encoding	155
Bibliography	156

Glossary

The following symbols are used in this thesis:

Symbol	Description
a	A scalar is denoted by a lower case letter.
\vec{a}	A vector is denoted by a lower case letter.
\vec{A}	A matrix is denoted by an upper case letter.
c	Index of a camera or its image in a multiocular camera system.
d	Index of the current dimension.
\vec{H}	Homogeneous transformation matrix (4×4).
i	Index of a feature point.
j	Index of an image in a sequence or set.
k	Number of the current iteration.
l	Index of the lattice point.
s	Position along a curve. For example $s = 0$: Beginning of the curve, $s = 1$: End of the curve
t	Index into a set of curves. For example $t = -1$: left side of a tube model in the image, $t = +1$: right side of the tube.
$\vec{\Phi}$	Parameter of an optimisation, a pose.
$\tilde{\vec{\Phi}}$	Result of an optimisation, the best matching pose.
$\tilde{\vec{p}}$	Measured point (image coordinates).
\vec{p}	Projected model point (image coordinates).
\mathcal{P}	Projection function (including distortions).
$\tilde{\kappa}$	Calibration parameter.
T	Template.
$\vec{\Lambda}, \overline{\Lambda}$	Predetermined limits of the pose.
I	Grey level image.
D	Distance image.
θ	Predetermined threshold. Subscript gives type.

Additionally entities have super- and subscripts, prefixes and postfixes. If all of them are present a symbol ${}^{\beta}_{\gamma}\alpha_{\delta}$ means:

Script	Description
α	Symbol. May be scalar, vector, or matrix.
β	Target coordinate system (transformation matrix) or current coordinate system (point).
γ	Source coordinate system (transformation matrix).
δ	Index in a vector, set, or sequence.

Abstract

In this thesis various approaches to object recognition are investigated, mainly recorded by multiocular imaging systems. The strengths and weaknesses of the approaches are analysed. Several different methods — Template Matching, Feature Pose Maps, Contracting Curve Density, Active Contours — are implemented, thoroughly investigated, and evaluated. These methods make use of images synchronously taken by multiple calibrated cameras in order to improve the location accuracy and to overcome ambiguities.

Based on these information a new object recognition and localisation algorithm is designed, implemented, and tested. This algorithm is based on the sign of the gradient of the feature-model distance in the image. Since the model function is reduced to very little information (its sign) at a given feature position, the evaluation is very fast as it can be reduced to a simple table lookup. Thus we named the new method Gradient Sign Tables.

In order to conveniently obtain a calibration of an arbitrary number of cameras a new calibration procedure is designed, implemented, and tested that is centred around a reliable, automatic calibration pattern finder.

Based on the methods above a 3D object recognition system is designed, implemented, evaluated, and adjusted for selected practical applications. It has the following properties:

- The object pose is obtained with application specific degrees of freedom.
- The type of an object is automatically determined out of a set of given types with rejection of unknown objects.
- Corresponding points between the images and the model of an object are automatically determined.
- The implementations of the algorithms fulfil the timing requirements of typical industrial image processing applications. Processing times of at most seconds, but not minutes are achieved.

-
- Moderate requirements are made to the hardware in terms of image resolution and memory consumption. Usually VGA sized camera resolutions are sufficient.
 - No algorithm or implementation has arbitrary limits in image size or model complexity, other than those imposed by memory consumption and desired run time.

The camera calibration is evaluated using 100 images per calibration rig position. So we can obtain the influence of the image noise on the calibration parameters. We could verify the rules of thumb regarding rig placement and found one new rule that is important for multiocular cameras: Depicting the rig in the corners of the calibration volume improves the external calibration parameters. Our correlation based corner detector achieves an accuracy of 0.018 pixels (average) with a worst case error of 0.3 pixel near to overexposed portions of the image.

The methods in this thesis are evaluated using two example application from automated quality assurance. The first application estimates the pose of a rigid object — an oil cap — in order to verify its correct mounting. Additionally the class of the oil cap is obtained in order to verify the mounting of the correct type of oil cap.

We found that the multiocular methods achieve a depth error that is 2–3 times smaller than that of the monocular methods for rigid objects. We also found that performing the distance transform of the image and storing a small model is preferable to a distance transformed — and therefore large — model or the computation of the distance transform on-the-fly.

The second application deals with the recognition of non-rigid objects. We obtain the trajectory of a tube or cable by starting from a fixed end (“dangling rope problem”) and follow it using object recognition methods. Using example images of varying contrast, three different methods are evaluated: Contracting Curve Density, Active Contours, Gradient Sign Tables.

Gradient Sign Tables are more robust and have a simpler implementation. They achieve a slightly higher error due to the edge quantisation to integer pixel coordinates. The Contracting Curve Density algorithm is the most accurate of the three methods, but cannot cope very well with small object. This is the domain of the Active Contours, but at a higher error.

Tubes and cables can be traversed for the complete length if a suitable contrast is present. In this case the average error is about 1.5 mm for a tube-camera distance of about 700–

1000 mm and a tube diameter of 7–25 mm. If only one edge of the tube or cable has a poor contrast, the methods can follow a shading related edge to some extent. In this case the average error is about 20 mm.

Based on these findings we present some possible improvements and directions of future research.

Part I

Introduction

1 Motivation

1.1 Context

In the field of image processing, industrial image processing is probably the topic of the largest economic, ecological and social importance. Using image based quality control methods the quality of products is enhanced, which immediately leads to more profit, more investments and more jobs. With less rejects due to improved product quality, less natural resources are used per product and the competitiveness of the enterprise is improved.

Manual quality control is a mentally exhausting job, especially if 100% of the parts are to be checked. Workers doing such tasks are usually exchanged after about two hours to maintain a sufficient level of attention. Relieving these workers from their both boring and extremely responsible job improves the quality of their lives and the quality of the product.

Additionally robust and precise object localisation methods allow a much higher degree of automation during production. A large percentage of jobs in manufacturing consists of taking things out of containers and placing them somewhere. In some cases this is currently technically impossible (e.g. where fine sensomotrics are required as during assembly) or commercially not viable. The continuing progress in computer and sensor performance leads to cheaper hardware, which in turn make once expensive set-ups now economically interesting alternatives.

Using only one camera and no metric information about the observed object (e.g. the length between two points in metres), the absolute values of size and position cannot be

obtained. This problem can be solved by using further cameras capturing the scene from different viewpoints, or geometrical information about the real-world object. Given the relative positions of the cameras, size and position of an observed object can be obtained by triangulation.

Due to these advantages this thesis is concerned with the use of multiple images for object recognition. The methods investigated here can be used for solving either, or both, of the aforementioned problems.

1.2 State of the Art

1.2.1 Overview

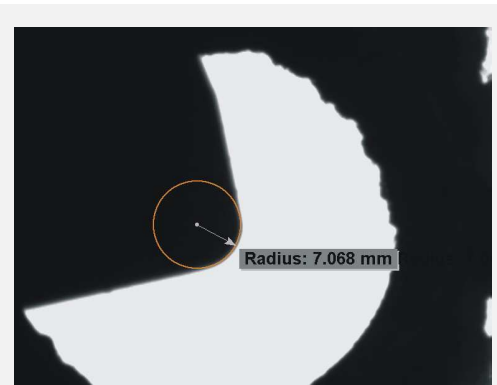


Figure 1.1: 2D image processing software. Source: www.mvtec.com, 12 Feb 2007

Current commercial computer vision systems and applications usually record and interpret the outside world in only two dimensions (Fig. 1.1). Naturally only two dimensional positions, size and planar rotations of real world objects are obtained from the image, usually by algorithms that are easily adjusted to a new task even by laypersons. This is usually supported by constructive restrictions, e.g. the inspected objects are to be placed on a conveyor instead of leaving them in the storage box they were transported in. Fig. 1.2 depicts such a setup.

If three dimensional measures are to be obtained by these state of the art algorithms, e.g. size and positions of holes or mounting points, this is reduced to two dimensional methods by using a relatively large number of cameras. Each camera inspects one e.g. hole, obtains its two dimensional size and position, and eventually from the individual data and the camera poses the final measure is computed. It is obvious that contraptions like this cause an enormous calibration and maintenance effort. Over the last few years the situation improved, and now commercial systems [50] are available that reduce the number of cameras by observing multiple features (e.g. holes in the example above) per camera, thus reducing the maintenance effort (Fig. 1.3, page 4).

More interesting than these methods are 2.5 dimensional object recognition and localisation algorithms (Def. A.2), a subject of past and current research. The methods obtain the three dimensional orientation and position of an object by recognising a change in appearance, e.g. the object became larger in the image, therefore it must be closer to the camera. Methods of this type heavily rely on the distance scaling property of perspective optics [28].



Figure 1.2: Conveyor, camera (background), gripper of an industrial inspection system. Source: www.mvtec.com, 12 Feb 2007

From an engineering point of view this is acceptable because telecentric optics — which create a distance independent image of the object — are large and heavy, which make them unsuited for many applications. The size of those lenses is caused by the fact the observed area is limited by the size of the aperture [80]. For many industrial inspection tasks this would require a prohibitively large optics. See [80] for a cheap method to convert perspective optics to telecentric optics for small observed areas.

Out of the three dimensional methods stereo vision based scene reconstruction is probably the best known. Corresponding points from two camera images are identified and using projective geometry a triangulation yields the three dimensional coordinate of the point. Using homogeneous coordinates the underlying epipolar geometry (Def. A.9) can be expressed in a compact manner. Various methods exist to obtain a more or less dense reconstruction

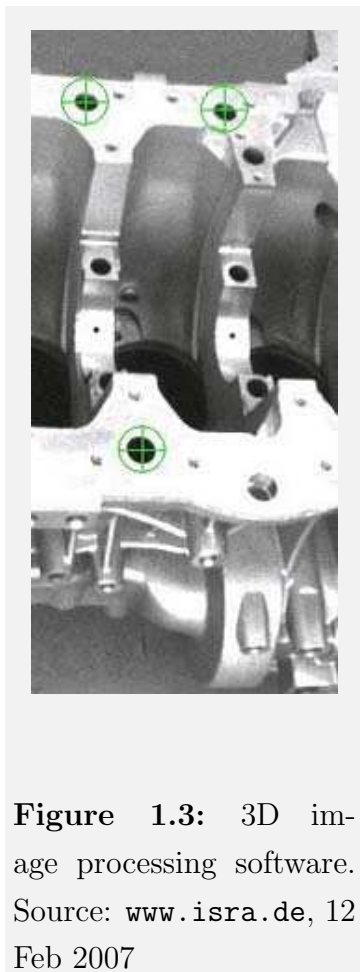
of the scene [56, 72] or modelling of objects [16, 67].

These scene reconstruction methods have the drawback that they have to solve the correspondence problem before solving the localisation or recognition problem. Although methods exist that integrate these problems into one algorithm, even mutually compensating the error of each other, this intermediate representation is of secondary interest for the task of object recognition. True, the object's pose and type can be obtained from such a reconstructed scene [16], but a more direct path from pixel matrix to object pose is desirable in order to reduce the sources of error.

The principle that models of the object can be integrated into the recognition process is

also widespread. Early approaches reformulate the model as a knowledge base and use classical Artificial Intelligence methods [5]. Later on more robust approaches were pursued (e.g. [48, 75, 84]) that relied less on symbolic logic, and more on numerical approaches. In this thesis we will analyse methods from three groups: appearance based methods, segmentation methods, and closest feature methods.

1.2.1.1 Appearance Based Methods



Appearance based methods use the object model to create a database of object appearances. This is usually done by simulating the image acquisition process (e.g. using technologies originally developed for Virtual Reality or computer games) or by placing object or camera on a computer controlled actuator (e.g. industrial robot). In both cases the recording process is cumbersome to set up and time consuming. Appearance-based methods use a so-called *viewer-centred data representation* [79].

The recognition process of appearance based methods consists of a suitable comparison [5, 79] between the appearance database and the object's appearance in the image. Each entry in the database is attributed with the pose parameter used during the creation of the entry. A suitable interpolation yields the pose even more accurate than the resolution of the database.

Appearance based methods can be further subdivided by the frequency of occurrence of the features they use: If a feature exists for almost every object pixel the algorithm is said to use *dense features*. *Sparse features* are used if only a minority of pixels is used for matching. Image template matching [5] uses dense features, since complete image matrices are compared. Edge template matching uses sparse features, as relevant edges make up only a small portion of the image [79].

Classically, appearance based methods refer only to dense feature [5, 15]. We'd like to extend the term *appearance based method* to sparse features if a database of object images is involved and this database is checked against the image to be matched. Refer to Sec. 1.2.2

for an analysis of a recent dense appearance based matching procedure. Refer to Sec. 1.2.3 for an analysis of a recent sparse appearance based matching procedure.

According to the definition above, Active Appearance Models [20] are dense appearance methods too, although they reconstruct the appearance not from a discrete set of appearances, but from a statistical model of them. This is basically a way to introduce intermediate views, just as the method analysed in Sec. 1.2.2 is. The difference is the choice of the reconstruction base.

Sparse appearance based methods which include generalised Hough transform, Geometric Hashing, and Image Registration/Pose Estimation extract two dimensional features (points, lines, arcs, ...) from the images, and compare them with the database. In this process primarily the pose is obtained, the recognition is done by computing a measure of similarity [84].

The main advantage of appearance based methods is their ability to store the image positions of features and their visibility in the same object representation: Only the positions of visible feature are stored. Deducing the occlusion state of a feature such as a line from pose and 3D model either requires a time-consuming analysis of the geometrical relations or computer graphics hardware.

The disadvantage of appearance based methods is their memory consumption. As the pose space is to be quantised such that each point in pose space is assigned individual appearance data, the available memory must be managed such that the recognition task can be solved. This either requires elaborate caching schemes or a reduction of the degrees of freedom that can be handled by the application.

1.2.1.2 Model Based Segmentation Methods

Model based segmentation methods such as Active Contours [12, 45, 68, 83], or the Contracting Curve Density Algorithm [37, 38, 39, 40] adjust the parameters of a suitable geometric representation such that it divides the image in two distinct set of positions: inside and outside the models outline. This distinction is iteratively improved. These methods use a so-called *object-centred data representation* [79].

Active Contours maximise an energy function that depends on the shape of segmentation boundary and the image content. The model information may be integrated either as

additional energy terms that prohibit larger deformations, or limitations of the search space during the maximisation, or both.

The main advantage is that Active Contours can accommodate flexible objects or slight deformations of rigid objects. In order to use them for model-based object recognition, the outline of the object must be easily obtainable as operations on it are very frequent. Furthermore they suffer from a limited radius of convergence, which is usually corrected by search methods such as particle filters [3].

The less well-known Contracting Curve Density Algorithm [37, 38, 39, 40] maximises the probability that the model curve separates the pixel value statistics on either side. Using the Bayes Theorem and suitable approximations of the probabilities involved, the posterior probability $p(I|\Phi)$ with I being the image and Φ being the model parameter is maximised and therefore the curve is fitted to the image. Therefore, it can be seen as a model-driven segmentation procedure.

The advantages of this method are that it converges within a few iterations, is very robust to cluttered background, and offers an easy exchange of the model curve. Its drawback is that for matching a self-occluding object the model curve generation becomes the limiting factor in regard to performance.

Refer to Sec. 5.1 and 5.2 for the detailed analyses of Active Contours and the Contracting Curve Density Algorithm.

1.2.1.3 Closest Feature Methods

As the name indicates, closest feature methods search the closest feature from model to image or vice versa and use it to establish a preliminary correspondence. Using these correspondences the current pose of the model is improved by minimising a distance measure (e.g. the average distance of the model features projected to the image). This process is iterated to convergence.

The best known method in this field is the Iterative Closest Point (ICP) Algorithm [6]. It operates in exactly the aforementioned way and is used to refine the transformation of pre-registered geometrical entities embedded in the same space (e.g. 3D points to 3D points, 3D points to 3D lines). There are numerous variants and applications of this algorithm.

The largest advantage of the classic ICP is its simplicity. Its disadvantages are a small radius of convergence and a strong sensitivity to outliers or missing data.

One approach to speed up the closest point computation is the use of a distance transform [30] of the image features. This avoids establishing explicit correspondences which may be subject to combinatorial explosion. Furthermore the distance transform allows the elegant treatment of missing data, which only locally increase the average distance.

Another closest feature method is presented in [75], where the object model is distance transformed by expressing its surface as the zero level-set of an implicit function. Points outside the object have an e.g. positive value, points on the inside a negative value. One constrained optimisation per image feature position is performed to find the closest points on the surface and on the projection ray. If these 3D correspondences are obtained, a new pose is computed (3D-3D pose estimation). The process repeats to convergence.

The advantages of the method are its ability to directly integrate multiple views into the recognition process and the large radius of convergence. The disadvantages are the long run-times due to the optimisation for each point and the choice of polynomials as the implicit function to express the distance transform.

Simple and smooth objects can be represented well by implicit polynomials [75], which allow a large radius of convergence. The polynomials of lower degrees are not sufficient to represent the sharp corners of man made objects, the polynomials of higher degrees suffer from stability problems such as spurious zero sets. Even sophisticated fitting procedures as from [46] offer only little help there. Functions of local support [69] suffer from inflexion points near the surface and an object enclosing zero set. These properties prohibit its use in practical tasks.

We developed the method of Characteristic Localised Features [54] prior to this thesis. This method projects salient points of the 3D model in the image and, by means of a local search, find the closest points. It allows very different types of features to be included in one model, such as points and lines on one hand, and image templates mapped to planar surfaces on the other hand. This is advantageous as the combination of different kinds of features improves the robustness of the matching. The disadvantage is that a uniform treatment of the different objective functions for finding the closest point is only possible with a large amount of prior knowledge [55].

Refer to Sec. 1.2.5 for an analysis of a closest feature algorithm with explicit correspondences.

1.2.1.4 Multiocular Object Recognition and Camera Calibration

Most of the methods above (aside from stereo and [75]) are monocular methods. Most of them have well-known properties, advantages and disadvantages alike. Quite a number of publications show the more or less successful application of these methods to object recognition tasks. To the best of the authors knowledge only Geometric Hashing [78] and Hough Transform [36] have been investigated so far in multiocular applications.

All the methods above require detailed knowledge about the geometrical properties of the cameras. This knowledge is obtained by camera calibration. Various camera models [11, 28, 77] can be found in the literature, along with calibration procedures [11, 77, 85, 86]. As the state of the art in this field does not influence the general scope of this thesis, it's analysis is postponed to Sec. 3.2.

1.2.2 Analysis of a Dense Appearance Based Matching Method

1.2.2.1 Algorithm Summary

In [71] Reinhold et al. propose an appearance based object recognition and localisation method that is based on two basic ideas:

1. The silhouette of an object varies with the pose. In order to improve the recognition only object pixels are to be matched, background pixels are to be ignored.
2. Similar poses yield similar appearances. Given a suitable function base, both the object outline and the grey/colour/feature image can be expressed as linear combinations of the basis functions depending on the pose parameters.

The basis functions are chosen to be the sine-cosine-decomposition $\{\cos \omega\Phi, \sin \omega\Phi\}$ for $\omega \in \mathbb{Z}_+$. Φ is comprised of e.g. the two angles azimuth and elevation. This is closely coupled to the image generation process: The object rotates on a turntable (azimuth) and the camera rotates around the up and down around the same point (elevation). Scaling (distance of camera from centre of rotation) is modelled in the same way using the cosine only. The functions base v_r consists of all N_r products of the sine and cosine elements. For azimuth (Φ_1) and elevation (Φ_2) this results in the functions base summarised in Eq. 1.1.

$$v_r(\Phi) = \langle 1, \cos \Phi_1, \sin \Phi_1, \cos \Phi_2, \dots, \cos 2\Phi_1 \cdot \cos 2\Phi_2 \rangle^T \quad (1.1)$$

Whether a pixel at image coordinate m belongs to the object or not is determined by Eq. 1.2.

$$\xi_m(\Phi) = \sum_{r=1}^{N_r} a_{\xi,m,r} \cdot v_r(\Phi) \quad (1.2)$$

Any pixel for that $\xi_m(\Phi)$ that is larger than a threshold e.g. 0.5 is assumed to be part of the object, all other pixel are assumed to be background pixel. The pixel coordinates of an object form its bounding region and will be denoted by O , with $N_O = |O|$. The bounding region is therefore different for every pose.

The grey/colour/feature values c_m of the pixels are assumed to be from independent Gaussian distributions with per-pixel mean μ_m and standard deviation σ_m . In order to obtain a suitable statistics to compute μ_m and σ_m , a set of images under different lighting conditions, but with the same pose is recorded. The model of the mean of a pixel value itself is a linear combination of the base functions as Eq. 1.3 details, the standard deviation is independent of the pose.

$$\mu_m(\Phi) = \sum_{r=1}^{N_r} a_{\mu,m,r} \cdot v_r(\Phi) \quad (1.3)$$

The model parameters $a_{\cdot,m,r}$ are computed from a set of training images. Each image exhibits a black background so that the object can easily be segmented and the training values for O are easily obtained. There are multiple training images per pose, each with a different lighting.

The probability of an object at image position t is the joint probability of all pixels having the current grey/colour/feature value. Eq. 1.4 denotes this probability. The matching objective function F is the geometric mean of the individual probabilities as denoted in Eq. 1.5.

$$p(C_O|B, \Phi, t) = \prod_{m \in O} p(c_m | \mu_m, \sigma_m, \Phi, t) \quad (1.4)$$

$$F(\Phi, t) = (p(C_O|B, \Phi, t))^{\frac{1}{N_O}} \quad (1.5)$$

B comprises of the per-pixel mean and standard deviations. C_O is the vector of the object features c_m .

Matching is performed by finding the maximum of Eq. 1.5 using a two level hierarchic search: First $F(\Phi, t)$ is evaluated at coarse steps of Φ and t , then at promising positions a finer search is performed. The objective function $F(\Phi, t)$ is the geometric mean of the individual probabilities in order to normalise it to the different number of pixels in each pose.

1.2.2.2 Analysis

The algorithm is rather economical concerning memory consumption, compared to the original images: approx. 3500 images can be expressed by 52 basis functions. This stems from the fact that neighbouring poses have similar feature images. This allows the impressive run times of 1.7 seconds per localisation*. The method scales linear with the number of objects.

Due to the inclusion of various illumination situations during training the classification rate exceeds 67% for scenes of heterogeneous background and 20% occlusion of the objects. Nonoccluded objects are found at least approx. 77% of the time in heterogeneous backgrounds. These values apply to three databases (office items, e.g. stapler, hospital objects, e.g. pillbox, and household items, e.g. cups).

For some applications in industrial image processing this recognition rate is sufficient to reduce the number of workers by three fourths as occlusion can be prevented in almost any case. In bin picking application this is done by having the camera look down into the bin. Objects can be taken only from the top of the bin, which are therefore not occluded by other objects.

Some objects (cutlery) can't be recognised well with this algorithm, but this probably stems from the parameter settings which were identical for all objects. We know from experience with template matching algorithms used prior to this thesis that the form of an object (long and slim vs. roundish) influences the choice of the search parameter: Assuming an algorithm with a fixed radius of convergence (e.g. 50% overlap between object and template), the search grid must be much finer for a small object, otherwise the matcher skips over the object without noticing it. This will of course increase the matching time.

*2.4 GHz Pentium IV

A further drawback of the algorithm is its poor localisation performance: Again this might be a problem of the matching parameters as the finest resolution was 4 pixels. This does not explain why for some experiments the algorithm 50% of the time yields poses that departed more than 10 pixels, 15° spatial orientation, or 2 cm in depth from the ground truth.

Such a localisation accuracy is useless for e.g. grasping or correctness checking. Without knowledge about the camera it is quite hard to obtain the spatial accuracy. As it is not explicitly stated in the paper, we estimate it to about 3 mm, as the front image of a stapler consist of approx. 4000 pixel according to [71], and the front of a stapler is approx. 2 cm high and wide. This leads to approx. 63 pixel per two cm, or approx. 3.2 pixel per mm.

We assume that both the angular and spatial accuracy can be improved at the expense of processing time and memory consumption by using more basis functions. This is the biggest drawback of dense appearance based matching algorithms: Because there are so many features, the reference values for the features have to be stored — which takes space — and compared — which takes time.

Only the relative position of these values is of interest: For an object of approximately uniform colour and planar surface (e.g. stapler from the top) each model pixel and image pixel correspond. The position can only be determined from the fact that more area overlaps at the correct position. If parts of the object are occluded (e.g. the right edge) the positional information contributed by these pixels is ambiguous: Only the vertical direction exhibits a sharp maximum, a horizontal ridge is formed.

1.2.3 Analysis of a Sparse Appearance Based Matching Method

1.2.3.1 Algorithm Summary

In [79] an object recognition and localisation method based on edge templates is presented. An edge template is a list of pixel coordinates where edge pixels are located, relative to a reference point.

In order to match a single edge template T with an edge image, the edge image is distance transformed (Def. A.8): each pixel is assigned a value that is related to the distance to the closest edge pixel. For each pixel position $p \in T$ and each template position t the mean (or Chamfer) distance is computed, with D denoting the distance image:

$$f(t, T) = \frac{1}{\|T\|} \sum_{p \in T} D(p + t) \quad (1.6)$$

The best matching position b of the best matching template B out of the template set \mathcal{T} is obtained by finding the minimum of f :

$$b, B = \operatorname{argmin}_{t \in \|D\|, T \in \mathcal{T}} f(t, T) \quad (1.7)$$

The resulting pose is interpolated to sub-grid accuracy using the 30 best templates.

The matching procedure is implemented as a hierarchical search that groups templates that are known to yield similar values of f for the same image. The template hierarchy is build off-line using a “partitional clustering algorithm based on simulated annealing” [79].

For a suitable set of poses the object’s images are rendered by PoVRay [27] and the edges are extracted.

1.2.3.2 Analysis

The method yields an accuracy of better than 4° (worst case) and better than 2° (most cases). If the point of view is carefully chosen (which is simple, as the camera is robot mounted) the accuracy is about 1° .

An oil cap was located within the image to 5 degrees of freedom stored as 1331 or 4550 templates. The computing time[†] is about 200 ms. In a different application two objects (ignition plug and socket) were classified and located with an accuracy of 0.5 mm. For the oil cap the recognition rate is excellent (better than 96.5% at 0% false positives).

Although the method uses the templates directly instead of a functional approximation as in Sec. 1.2.2 it achieves a better accuracy at shorter computation time. This is due to the vastly reduced number of pixel accesses: Usually less than 1% of the pixels are edge pixels.

The implementation, however, as presented in [79] is strictly monocular. The use of pixel coordinates instead of spatial coordinates indicates this.

We chose the edge template matching as one of the methods to be extended to multiple cameras for its simplicity in both training and application, its processing speed and high accuracy. Refer to Sec. 4.1 for this extension.

[†]2.4 GHz Pentium IV

1.2.4 Analysis of a Multiocular Appearance Based Method

1.2.4.1 Algorithm Summary

Harrie van Dijk's Ph.D. thesis [78] extends the Geometric Hashing method to a stereo camera system.

Geometric Hashing (GH) is an *Indexing Method*. It is used to detect objects in images based on generic features (e.g. corner points). Doing so it handles the correspondence problem and the pose estimation problem in an integrated way: If the correspondences are set up correctly, the pose can be computed correctly and a consistent solution is found.

This is accomplished by computing a hash table from the reference image (the indexing, or model building, step) and a voting scheme using this hash table (the recognition step). GH is therefore most efficient for model-based object recognition, where the hash table can be prepared off-line.

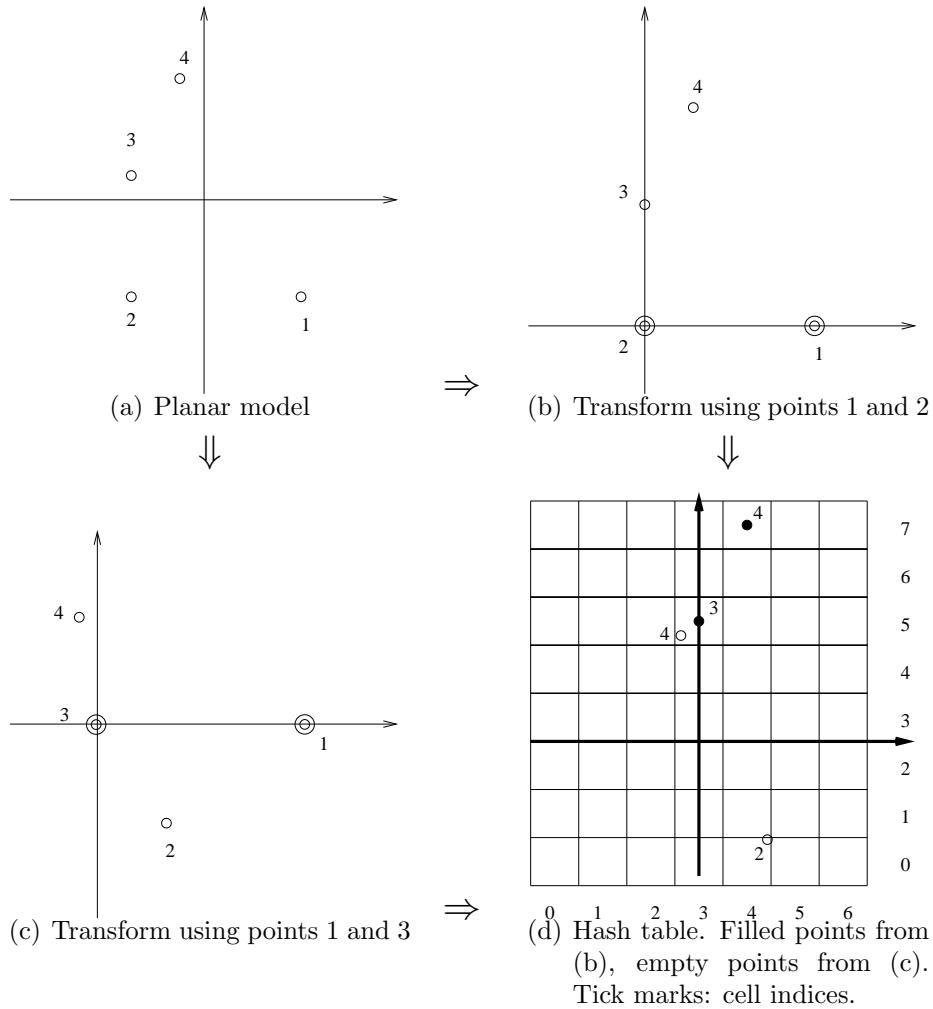
We will explain the construction of the hash table using a single planar model. The three degrees of freedom used in the following example are planar rotation and translation. Fig. 1.4(a) depicts the model to be preprocessed.

For all point combinations (e.g. 1–2, Fig. 1.4(b) or 1–3, Fig. 1.4(c)) a base coordinate system is computed and the resulting points (i.e. 3, 4 for Fig. 1.4(b)) are transformed into this coordinate system. A grid of cells is defined (Fig. 1.4(d)), the hash table. For each occupied cell the points are stored together with its base co-ordinate system.

The matching procedure performs similar steps: Given a set of feature points all combinations of two of these features are used to form a new coordinate system and the remaining points are transformed to it. For each cell in the hash table that is hit by the transformed points, an accumulator is incremented. There is one accumulator per basis and the bases that exceed a percentage of votes are accepted matches.

The multiocular extension of GH (MOGH) is performed by computing *all* 3D coordinates that can be formed from one point in one camera and the other points close to the epipolar line (Def. A.9) in the other camera. This produces false-positive points, which in turn are handled by the GH.

The MOGH poses are regarded as initial hypotheses only, they are checked by reprojecting the 3D model in the image. This reduces false-positive matches.



Cell	Content (Point, Base)
(3, 5)	(3, 1 - 2), (4, 1 - 3)
(4, 0)	(2, 1 - 3)
(4, 7)	(4, 1 - 2)

(e) Hash table in numerical form.

Figure 1.4: Geometric Hashing: Hash table generation. Each point pair forms a new coordinate system, the remaining points are transformed to it. The transformed point coordinates are stored together with the basis in a discrete grid: the hash table. For illustration only two bases are shown. After [78]

1.2.4.2 Analysis

In a sense [78] is the most similar work in literature to this thesis: A well-known monocular object recognition algorithm is extended to multiple cameras. It is, however, different in that it computes 3D correspondences from the images and uses them as the primary matching coordinates.

The MOGH is rather fast as less than 200 points were generated for the test scenes (white polyhedral objects on grey background). A hash table implementation on recent hardware and using sophisticated index computation [73] can definitely be done in real-time.

The recognition rate of 70% at 13% false positives for these simple scenes is rather low. One would expect at least 90% percent recognition at a few percent false positives for the homogeneous background. No information is to be found on the accuracy of the resulting poses.

The approach shows that multiple cameras improve the recognition rate compared to monocular matching of the same scenes. It also shows very clearly that computing all possible 3D correspondence first and filtering them later does not yield sufficiently robust matching results.

In the chapter describing related works *Enhanced Geometric Hashing* [60] is mentioned. The method computes the resulting transform from assigned feature points and casts a vote in the hough accumulator of the transform parameter space. At the end, peaks in the accumulator are identified and the resulting transform and object is returned. A method like this is trivially extended to multiple views if performed for each camera independently. All cameras vote in the same accumulator while taking into account their own transformation to e.g. camera 0.

1.2.5 Analysis of a Closest Feature Method

1.2.5.1 Algorithm Summary

Beveridge describes in [7] a simple and efficient closest feature algorithm. It consists of a method to hypothesise model/feature assignments, a pose estimation based on that assignment, and a fitness measure given pose and assignment. The assignments are then modified in order to improve the fitness measure. As only one assignment is changed at

a time this method is a local search, a combinatorial optimisation. The assignments are selected from image features that are close to the model features at the starting pose.

The features used this implementation are 2D lines obtained from binary edge images and 3D lines in the model. No visibility checking is performed on the lines. No indexing phase is performed, the method tries to find the best match, even in presence of multiple instances of the model.

In order to overcome the local optima, random assignments are built, optimised and the best solution yields the matching result. Further optimisation methods (such as simulated annealing) are evaluated in [62]. Beveridge [8] gives a few more examples.

1.2.5.2 Analysis

Determining a recognition rate is not done in [7], because given enough runs a random search will find the global optimum. Instead the number of runs to achieve a 99% recognition rate was computed, for multiple objects present and additional noise lines. This investigation was performed on 6 models, 48 scenes per models and 100 runs each.

In some cases (very distinctive objects) only two runs were sufficient. For other, highly symmetric objects (flower) 228 runs were required. The wall-clock times of these experiments is basically meaningless, the implementation was done in Lisp. However, [62] used a C implementation, which requires approx. 3.2 sec per run[‡]. This is approx. 0.13 sec per run on a 2.4 GHz Pentium IV if the computer was a DEC, or approx. 0.057 sec if it was a SPARC 2.

Explicitly establishing the correspondences first requires that a lot of combinations between model and image features have to be tried out. There are only a few options for approximations or optimisations because the use or omission of a single model/feature assignment changes the pose, which in turn changes the objective function value. The run-times for a given recognition rate are therefore quite high.

It is possible to extend the pose estimation to multiple calibrated cameras, as the 2D-3D pose estimation is basically a 3D-plane-point iterative closest point algorithm. This allows the image lines, which are expressed as a spatial plane, to be transformed into the coordinate system of one camera.

[‡]either Sun SPARC 2, SPARC 10 or DEC Alpha 3000

2 About This Thesis

2.1 Aim and Scope

The analyses above provided valuable insights into the problem of object recognition:

The two template matching methods (Sec. 1.2.2 and 1.2.3) showed the inherent flexibility of the appearance based methods: a new object can be added by storing its template data. Furthermore Sec. 1.2.2 indicated that the redundancies of neighbouring views on the viewing sphere can be expressed as a function of the object pose.

This fact is also implicitly used by the clustering of similar edge templates (Sec. 1.2.3). This implementation, however, does not make use of the pose values of the templates due to its origins in detecting people, where training is performed from sample images.

Indexing methods (Sec. 1.2.4) suffers from the need to establish correspondences first as they are bottom up methods. This causes problems due to the number of possible correspondences which have to be reduced by using high-level features such as lines instead of low-level features such as edge pixels.

The Closest Feature method (Sec. 1.2.5) taught us that the average distance of a feature point to the closest model point is a relatively smooth function with some local minima. This measure is therefore a valuable objective function for pose refinement applications if an economic way for computing can be found.

There are some points which are — to the best of our knowledge — not yet investigated:

- Given the same object representation, which is better: a bottom-up or a top-down approach to object recognition?
- Is it better to perform the distance transform of the model or the current image?
- Does multiocular evaluation always perform better, regardless of the object recognition method?

- Can we simplify the Closest-Feature Method, especially regarding the matching procedure and the model?

We would like to investigate these and similar points in this thesis while meeting a few basic requirements: Algorithms should provide low processing times and have a low memory consumption. To avoid the combinatorial explosion connected to explicit correspondences the generation of image-model correspondences should be performed in a nearest neighbour like approach. The generation of model representation from CAD data should be as simple as possible. Object shapes and poses should not be limited by the algorithm. It should also be possible to account for non-Euclidean degrees of freedom such as internal deformation (e.g. the angle between truck and trailer). The required prior knowledge of an algorithm about the problem should be as small as possible, however, all prior knowledge should result in a speed-up of the recognition process.

The aim of this thesis is therefore to investigate how 2.5 dimensional (Def. A.2) object recognition and pose estimation algorithms may be extended to multiple calibrated cameras such that all cameras are of equal importance and explicit correspondences do not have to be generated. This is accomplished by either integrating the 2.5 dimensional contributions of all images into one global function or performing matches on the images individually and integrating the spatial results. The 3D information about the scene is extracted only implicitly and on object level. We do not reconstruct the visible surface of an object, we obtain the pose of the object directly.

The methods to be investigated are selected to cover a wide range of operating principles: Bottom-Up methods (Feature Pose Maps, Sec. 4.2), Top-Down methods (Template Matching, Sec. 4.1), snakes (Sec. 5.1) and local statistics optimisation (Contracting Curve Density, Sec. 5.2).

Based on these investigation a novel recognition scheme (Sec. 6.2) using the signs of the gradient of a distance function is presented. It integrates all the favourable aspects of the investigated methods.

User interface issues such as operator guidance, scene planning etc. are beyond the scope of this thesis.

Applications to range images are not considered in this thesis, as the data quality and resolution of current range sensors is sufficient only for a selected group of problems.

2.2 Data and Methods Used

The object recognition system as proposed in this thesis relies on the availability of CAD models of the objects to be recognised. This is not a severe problem as modern production facilities are based on exactly those models. For simple objects the models can be re-engineered in short time using Open Source tools only.

A CAD model is assumed to describe the visible surface of the object and to be stored in form of a list of planar polygons (usually triangles). This so-called tessellated description may be obtained from the structural description such as constructive solid geometry. This format is widely used, almost all CAD programs export it, and arbitrary object shapes can be represented. A concrete file format is not required by any of the algorithms presented here. Most implementations make use of VRML 1 or VRML 2 files.

The following mathematical concepts and procedures are used in this thesis: linear algebra, homogeneous coordinates and coordinate transformations, calculus and Newton's method of optimisation*, projective geometry and the pinhole camera, statistics, basic graph theory. We assume the reader is familiar with these topics. Otherwise [21, 28, 51] give a sufficient overview of these methods.

The algorithms in this thesis use basic image processing methods such as filtering, edge extraction, and thresholding. More advanced methods for e.g. object recognition and localisation, correspondence building, and feature extraction are introduced in the respective sections.

All the algorithms presented here operate on extracted features. Therefore they may be applied to grey level images, colour images, infra-red images etc. as long as the respective feature may be extracted and the model fits the observation of the objects. The requirements of the aforementioned features, which limits the algorithms to extract them from the images, are stated in the respective algorithm descriptions. Throughout this thesis grey level images are used.

2.3 Notational Conventions

Certain notational convention outside the usual mathematical conventions will be listed in this section. Established notational conventions are listed here only if multiple notations

*Also the derived methods Gauss-Newton and Newton-Raphson

are used in other works. The notation of cited works is unified to identical or at least similar symbols to increase the consistency of this thesis.

Further definitions may be found in Sec. A in case a concept is not intuitively named.

The notation of *transformations* is similar to that of [21]: ${}^B_A H$ denotes a transform of points defined in coordinate system A into coordinate system B. The homogeneous (4×4) matrix ${}^B_A H$ consists of a rotation matrix and a translation vector. In some cases it is more convenient to use the rotation matrix and the translation vector separately. In these cases they are named ${}^B_A R$ and ${}^B_A T$. Keep in mind that the construction of H from R and T depends on the ordering of the individual transformations as matrix multiplication is not commutative.

Spatial points are denoted by \vec{P} . Projected points are denoted as \vec{p} . Computed values are denoted x , measured values as \tilde{x} .

2.4 Section Overview

Even though this thesis is supposed to be read in sequential order of the chapters this section serves as a directory to topics of particular interest to the reader.

Part II describes the *architecture* of the proposed multi-ocular vision system, its calibration and various algorithms for object recognition and localisation.

In Chap. 3 we illustrate the *foundations* of all methods in this thesis. Sec. 3.1 describes the vision *system at coarsest level*. The relationships between the four phases of object recognition are illustrated and the data flow from camera to result is depicted.

Sec. 3.2 takes a deeper look at the *camera calibration* procedure with respect to multi-camera systems. The values obtained by camera calibration describe a specific camera system and are used in all multiocular algorithms in this thesis. We describe *our design of the calibration rig* that is used throughout this thesis. A *novel robust rig finding algorithm* is presented that copes with a wide range of cameras, including fish-eye and catadioptric cameras. Based on the method by Bouguet a calibration procedure is designed and implemented to cope with multiple, arbitrarily positioned cameras. The similarity between calibration and object recognition is analysed.

Chap. 4 describes the *appearance based methods* used in this thesis: Template Matching and Feature Pose Maps. Sec. 4.1 gives an overview of *Template Matching*, as found in the

literature and *our modifications*. The section also describes *our extension* of the methods to multiocular images. Special emphasis is put on edge templates applied to distance transformed edge images. Sec. 4.2 provides a bottom-up method, the *Feature Pose Map*, corresponding to the top-down template matching from Sec. 4.1. Several *novel ways to improve the robustness* of the method are presented.

Chap. 5 explains how to use *segmentation* methods for object recognition, with emphasis on multiocular images. Sec. 5.1 describes Active Contours, also known as *snakes*, and their *extension to 3D and multiple images*. Sec. 5.2 explains the *Contracting Curve Density* Algorithm and shows *our newly developed extension to multiocular images*. This appearance based matching algorithm is especially interesting as no explicit edge extraction is performed. Instead a statistical bipartition of the input images is optimised along with the object pose.

Chap. 6 illustrates the mechanisms behind *Closest Feature Methods*. Sec. 6.1 describes an object recognition method that projects feature to the image and finds the *closest matching positions* of these features. A 2D-3D pose estimation is then used update the current pose in accordance with the feature matching results.

Sec. 6.2 introduces a novel pose estimation method based on *Gradient Sign Tables*. This methods can be used to compute the distance transform *without the requirement for distance images* or for *high speed pose refinement*.

Part III is concerned with the *evaluation* of the aforementioned methods. All chapters in this part first state the goal of the investigation, followed by the experimental setup and are concluded with the results as well as their interpretation.

In Chap. 7 we analyse the performance of the *camera calibration*. Chap. 8 compares the performance of the pose estimation and classification for *rigid objects*. Chap. 9 shows the evaluation of flexible objects, such as *cables and tubes*.

Part IV completes the thesis by providing an outlook to further applications, improvements, and open questions in Chap. 11. A summary of all insights gained in this thesis is given in Chap. 10.

The appendix in Part V provides a set of definitions in Appendix A, and the bibliography.

Part II

The Multicocular Object Recognition System

3 Methodical Framework

3.1 System Overview

This section provides an overview of the basic software architecture used in this thesis. We start by describing the different phases that can be found in the recognition methods in this thesis.

3.1.1 The Four Phases of Object Recognition and Localisation

All object recognition and localisation methods considered here can be divided into four phases:

1. Training phase of the camera parameters.
2. Application phase of the camera parameters.
3. Training phase of the object recognition system.
4. Application phase of the object recognition system.
 - a) System evaluation on synthetic images.
 - b) System evaluation on real images.
 - c) Operational system.

The terms *training phase* and *application phase* are used similar to the terminology related to classification methods, even if most of the algorithms here are not strictly classifiers. *Training phase* means the modelling phase of some aspect of the system, *application phase* means the use of this model to fulfil some sub task of the system. Whether or not these phases are separate or interleaved, explicitly performed or done on the fly, depends on the algorithms of the respective recognition system.

In phase 1 the connection between image coordinates and real world (metric) coordinates is established. Even human beings have such a step while growing up. It is this step that allows us to distinguish an air plane model from a real air plane, e.g. by comparing it to the size of a human being or the distance in which we see it. In a computer system this step is usually one form of camera calibration to model one specific sensor with application relevant detail.

Camera calibration in this sense is sometimes done implicitly during the training phase of the recognition algorithm. In this case any geometrical information (e.g. size, position) of the object can be extracted in multiples of pixels only, not metrically. This is due to the fact that no information about the camera is obtained, only about the appearance of the object. Current commercial quality control systems fall into this category. It is clear that changing the camera or its lens change the appearance of the object in the image, sometimes so much that all stored measures are invalid and have to be redone.

Phase 2 makes use of the camera calibration. In appearance based recognition systems (Def. A.5) having an explicit phase 1, phase 2 contains the phases 3 and 4. This is due to the dependence of the recognition data on pose and camera parameters.

Phase 3 extracts the relevant information about the objects to be recognised. This may consist of extracting the recognition relevant geometrical elements from the face list of the models.

In phase 4 the calibrated and trained recognition system performs its task. All information obtained during the training phases is used here to obtain object type, position and orientation from the recorded images. In order to efficiently develop and test even complicated recognition systems this phase consists of three sub phases, the evaluation of the system using synthetic images or pre-recorded real life images and the actual operation of the deployed recognition system.

The difference between these sub phases is the source of input images and what is done with the recognition results. During the evaluation the recognition results are recorded and

analysed to improve the system performance, in the deployed system these results directly influence the manufacturing process. Only phases 4a and 4b are relevant for this thesis and the results of this evaluation are listed in Part III.

In the recognition system proposed here, except for the template based methods, all phases are distinct and explicit. Phase 1, the camera calibration, is described in Sec. 3.2.

3.1.2 System Architecture

The architecture of the proposed multiocular object recognition system is depicted in Fig. 3.1. This architecture clearly belongs to the class of feed-forward image processing systems. At first the input images are acquired, either by loading from image files (data source I), directly from a camera (data source II) or from an OpenGL renderer (data source III). The output of either data source is an image tuple, which is passed on to the low level image processing layer. Here the relevant features are extracted and passed to the high level image processing layer where the actual object recognition (or feature preparation during training) takes place.

The high level image processing layer generates pose and class membership information which is analysed by the application logic. In the simplest case these data are visualised in a suitable way (data sink a). This is useful during algorithm development where no actual reaction is required. The deployed application usually feeds its output to either a machine or a process control (data sink b). During training the prepared features are sent to the recognition database (data sink c).

Some of the data paths I-a through III-c can be assigned to particular phases as Table 3.1 suggests. Data paths not listed make no sense and do not need to be implemented. Fig. 3.2 illustrates these data paths separately for the phases. Additionally the different data paths for appearance based methods and non-appearance based methods in phase 3 are depicted.

3.2 Camera Calibration

Camera calibration lies the foundation for all object recognition methods considered in this thesis. The aim of the calibration is to obtain the parameters of a certain camera

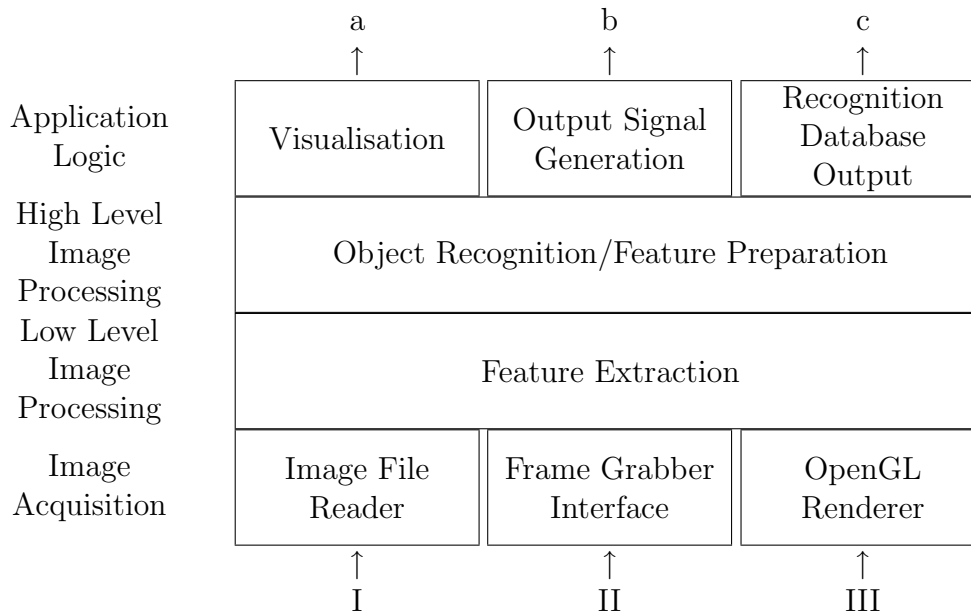


Figure 3.1: System architecture at software module level. Roman numbers (I-III) denote data sources, letters (a-c) denote data sinks. See Table 3.1 for data paths.

Table 3.1: Data paths in Fig. 3.1 and their assignment to the four phases. Missing paths are not meaningful.

Data Source	Data Sink	Application
I	a	Phase 4b
I	c	Phases 1 and 3, recorded images
II	a	Phase 4b, demonstration
II	b	Phase 4c
II	c	Phases 1 and 3, live images
III	a	Phase 4a, test on ideal images
III	c	Phase 3, ideal images

model, describing one physical camera in an optimal way according to this model. This also applies to a set of cameras.

Throughout this thesis the term *camera system* denotes a group of N_c digital cameras* mounted in a rigid frame with respect to to each other. Such groups consist of at least $N_c = 2$ cameras, even though calibration of single cameras can be done with the new

*Cameras with analogue image acquisition, analogue signal transfer and a frame grabber are mathematically and algorithmically identical to purely digital cameras.

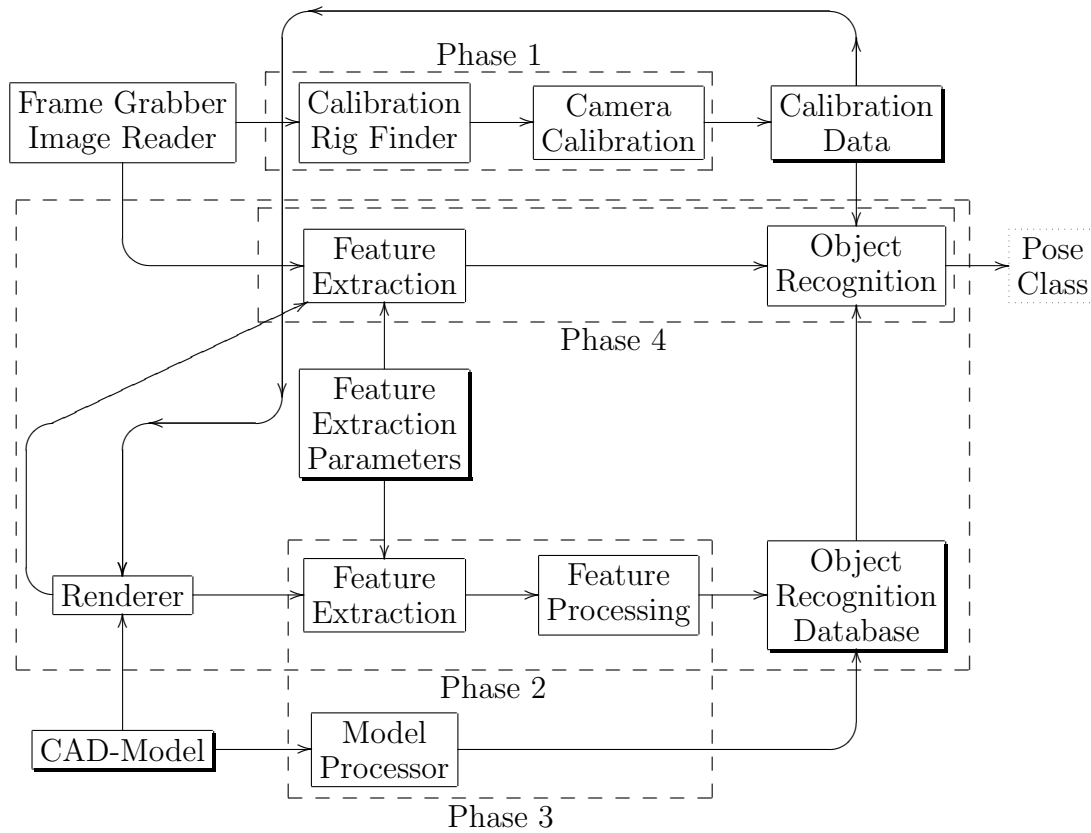


Figure 3.2: Data flow within and between the four phases of object recognition. Elevated boxes denote persistent data structures.

methods in this section. The imaging process which captures the outside world in form of grey or colour values in the acquired pixel matrix is modelled by a set of mathematical equations. One set of these equations is called a camera (system) model.

Camera models denote models of the imaging process of a single camera. A simple model, the *pinhole camera model* [28], assumes a *Camera Obscura* type (assumed to be described first by Chinese philosopher Mo-Ti, 5th century BC) of imaging, which models the imaging process so that light shines through a very small hole on the sensor chip. Only the magnification effect resulting from the distance between sensor, pinhole, and imaged object are accounted for. Tele lenses (long focal length) and expensive, so-called *distortion free* optics can be modelled adequately as a pinhole camera.

The parameters of the pinhole model are magnification factor, commonly called *focal length* or more accurately *camera constant*, the distances of the pixel centres, and the coordinates of the so-called *principal point*. This point is situated where a line parallel to the surface

normal of the sensor chip and passing through the pinhole intersects the infinitely thin, planar sensor chip. This differs from the usual definition involving the optical axis as a pinhole camera does not possess it. Eq. 3.1 and Fig. 3.3 illustrate this model.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f \cdot s_u \cdot \frac{X}{Z} + c_u \\ f \cdot s_v \cdot \frac{Y}{Z} + c_v \end{bmatrix}, \quad (3.1)$$

where f denotes the focal length in e.g. millimetre, s_u and s_v denote the pixel density in e.g. pixel/millimetre, c_u and c_v denote the coordinates of the principal point in pixel. The scene point $\vec{P} = [XYZ]^T$ is projected onto the sensor point $\vec{p} = [u, v]^T$.

The image points \vec{p} are defined in the *image coordinate system*. The origin of this coordinate system is at the top left pixel of the image. The positive u axis points to the right hand side of the image, the positive v axis points down.

The scene points \vec{P} are defined in the camera coordinate system, the origin of which is in the pinhole. The Z -axis points in viewing direction of the camera. The positive X axis is assumed to be parallel to the positive u axis, the positive Y axis is therefore parallel to the positive v axis in order to constitute a right handed coordinate system.

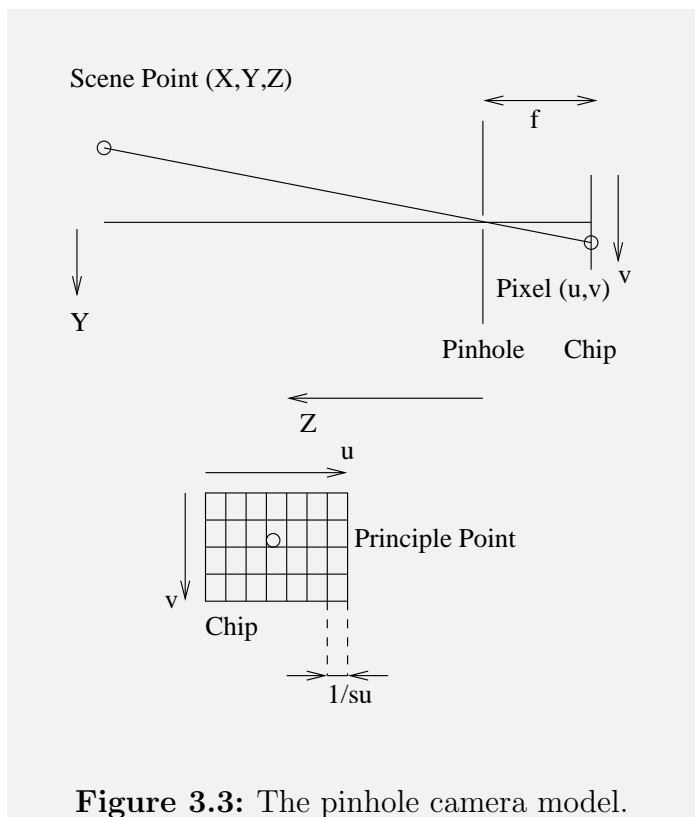


Figure 3.3: The pinhole camera model.

More sophisticated camera models consider that the imaging process of today's cameras[†] involves a lens or a group of lenses instead of the pinhole. Such lens based optics allow more light to illuminate the sensor compared to the pinhole, but the price for this advantage is the fact that such optical devices distort the acquired image. This is perceivable in that straight lines in the outside world become more or less curved on the chip (Fig. 3.4). The

[†]Lens cameras are assumed to be invented someday between the 14th and 16th century, for the use as painters aids much as the original Camera Obscuras were.

extend of this bending is coupled to the lens design. Whether or not this bending influences the image processing depends on the algorithm and its accuracy requirements.

It is clear that the pinhole model does not hold in presence of noticeable distortions, so a more complicated model of the projection is required. There are quite a lot of models that capture the distortion effects of a real optical system [11, 28, 77] better than the pinhole model. The model used in this thesis is explained in Sec. 3.2.2.1.

The parameters f , s , and c together with the distortion parameters of a model are called *intrinsic camera parameters*. These parameters cover geometrical properties of the camera only. Other parameters, e.g. blurring, are usually ignored as they are not important for object recognition tasks as long as the image is focused on the object. Methods that rely on such parameters (e.g. depth from defocus [18]) are not considered in this thesis.

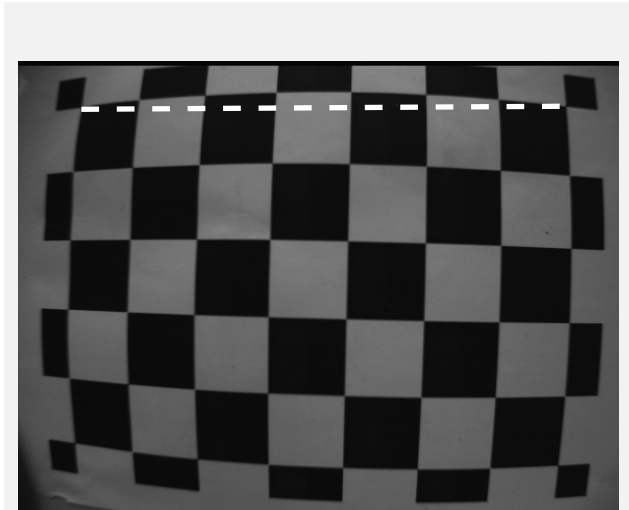


Figure 3.4: Imaging a regular checkerboard with lens distortion. The white dashed line was added to illustrate the amount of distortion.

The relationships between coordinate systems are modelled using coordinate transformations [21]. The parameters of the transform between camera coordinate systems are called *extrinsic camera parameters*. In this thesis two kinds of transforms are to be distinguished: the transforms from one of the N_c camera coordinate system to any of the other $N_c - 1$ camera coordinate systems, called *camera system internal transforms* and the transform from one camera coordinate system to a global reference coordinate system, the *camera system external transform*.

The calibration, corresponding to phase 1 in Sec. 3.1.1, is performed by estimating all aforementioned parameters, consisting of N_c sets of intrinsic parameters and $N_c - 1$ sets of extrinsic parameters. These estimation is performed using modelled scene points \vec{P} and their corresponding image points \vec{p} on the chip. The following five steps are performed in the calibration procedure used in this thesis:

1. Images depicting the calibration rig are recorded.

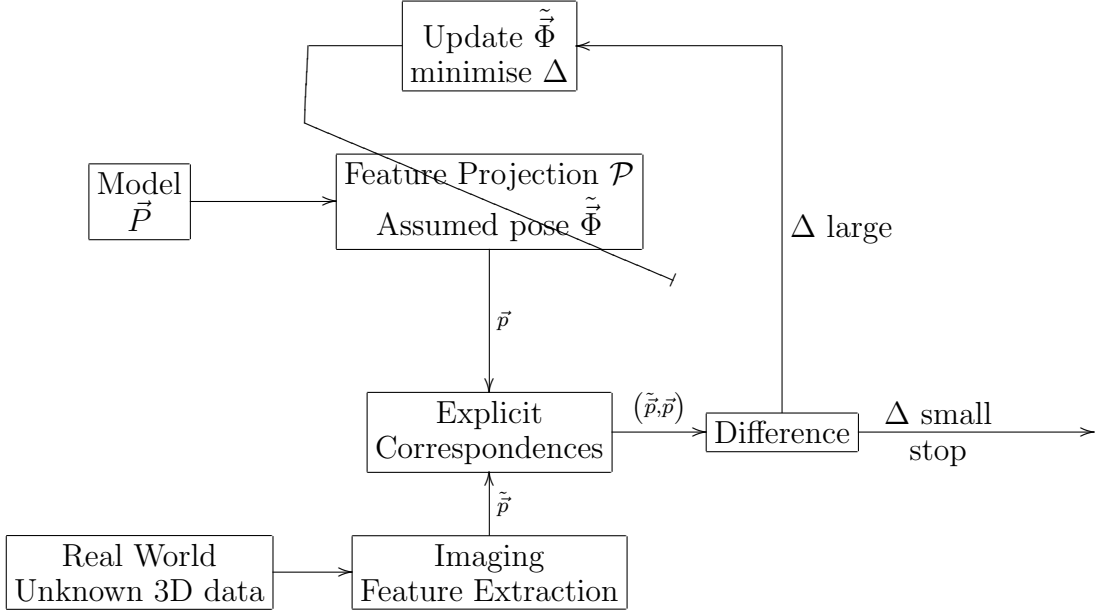


Figure 3.5: Bundle Adjustment: Principle of operation. The 2D-3D correspondences are established by the calibration rig detector.

2. Salient points in the image are detected and located with sub-pixel accuracy.
3. The extracted points are identified such that explicit correspondences between the 2D image points and the corresponding points of a 3D model of the calibration rig can be established. This includes the removal of false positives.
4. The unknown parameters of the camera system — and possibly additional metric and angular information — is obtained by optimisation.

The points \vec{P} have metric coordinates in the model (calibration rig) coordinate system. Their 2D counterparts \tilde{p} are the sub-pixel accurate estimated locations in the image. From, position and reference coordinate system of the points \vec{P} , the camera model and the estimation procedure are the differences between calibration schemes.

At least the points \tilde{p} are subject to noise (due to discretisation, sensor noise, etc.). In order to overcome these noise effects and increase the accuracy of the calibration result $\tilde{\Phi}$ *least mean squares optimisation methods* are used to minimise the difference between camera parameter related values \vec{p} and measurement related values \tilde{p} .

Although experiments with other approaches exist [41], usually the difference between 3D points \vec{P} projected as 2D points \vec{p} using a camera model \mathcal{P} (e.g. Eq. 3.1) and measured 2D

points $\tilde{\vec{p}}$ subject to a norm is minimised as denoted in Eq. 3.2. This minimisation principle is illustrated in Fig. 3.5 and denoted in the following equation:

$$\begin{aligned} \tilde{\vec{\Phi}} &= \operatorname{argmin}_{\vec{\Phi}} \sum_i \left\| \vec{p}_i - \tilde{\vec{p}}_i \right\| \\ \tilde{\vec{p}}_i &= \mathcal{P} \left(\vec{\Phi} \right) \end{aligned} \tag{3.2}$$

This equation minimisation can be seen as sub problem of Bundle Adjustment [76].

Due to the non-linearity of the camera models direct estimation of the camera parameters is impossible. Therefore indirect methods have to be employed, e.g. gradient descent, Newton minimisation etc. See [65] for a good introduction to these methods.

There are two major groups of calibration procedures: *self calibration* and *supervised calibration*. Self calibration methods use the observed scene to obtain the correspondences. Since the scene coordinates of the observed points $\tilde{\vec{P}}$ are unknown themselves, only correspondences of two dimensional points $\tilde{\vec{p}}$ are established between images. It is clear that — depending on the scene — such approaches are less robust and require more images to obtain the camera system parameters. The major advantage of self calibration is the fact that the calibration can be done continuously and therefore compensate changing camera parameters. Self calibration is used in autonomous vehicles such as underwater robots [70].

Supervised calibration on the other hand requires less images, is far more robust to illumination changes and the actual camera parameters. This is achieved by using a calibration object of precisely known geometry. The calibration object, also called *calibration rig*, consists of a set of points \vec{P} with known coordinates in the calibration rig coordinate system. The rig is designed such that its image can be easily distinguished from other objects in the image. This avoids false correspondences which would spoil the calibration result.

3.2.1 Calibration Rig Design and Detection

The purpose of the rig detection algorithm is to establish the aforementioned correspondences between the calibration rig points \vec{P} and their representations $\tilde{\vec{p}}$ in the N_c images. To be of broadest use, rig detection algorithms work purely within the images, they do not make any assumptions about the position, size and orientation of the rig in the image. Additionally they cannot make any assumption about either intrinsic or extrinsic parameters, as these are the values to be estimated. The choice of the calibration rig and

the design of the rig point detection algorithm, and moreover the correspondence building algorithm is therefore crucial to the performance of the calibration in terms of processing speed, reliability and accuracy.

3.2.1.1 Calibration Rig Design

Various calibration rig designs are available both commercially as well as for self-construction. A few of them will be analysed in this section.

Calibration rigs are to be designed such that there is a large number of known points \vec{P} on the rig, the rig is easily detectable by computer vision methods, and the image coordinates \tilde{p} of the rig points can be obtained as accurate as possible.

The key criterion for the design of calibration rigs is that each rig point \vec{P} must be identified unambiguously regardless how the rig appears in the images. One may see this as to assign a unique number to each rig point \vec{P} . Given the number, the actual coordinate of the point may be computed in the rig coordinate system. This unique number may be either visible on the rig or can be determined by the rig geometry (e.g. the rig is a regular grid of points and point 0 is at the lower left).

Monocular calibration is still possible in some cases if the correspondences cannot be established correctly. If, for example, the rig is a regular grid and the identification of points is wrong by one row of points the pose of the rig will have an offset of one row width. This case is indistinguishable from placing the rig at a shifted position and has therefore no consequences on the calibration result.

Multicocular calibration is no longer correctly possible if the rig is detected with incorrect correspondences in only one image of a single tuple. This is due to the fact that external calibration estimates the transformations between the cameras. The only data that is available for this are the image coordinates of the rig points. If the coordinates are wrong or assumingly corresponding points are actually different physical points the resulting calibration will be incorrect. In a calibration process according to Eq. 3.2, incorrectly assigned points will increase the error locally, which is reduced by the estimation procedure globally. This will change the camera system parameters $\tilde{\Phi}$.

The following example will illustrate this fact: About 20 images showing different rig positions and orientations are required for a proper calibration. If one out of the 20 image tuples is incorrectly assigned, the calibration is off by 5% of the difference between the

correct rig pose and the detected rig pose. Assume the rig is detected with a rotation of 180 degrees (i.e. the orientation was incorrectly detected in the opposite direction). The total error of the calibration will then be 9 degree. An acceptable error is usually less than a few tenths of a degree.

This illustrates the importance of a robust rig detection algorithm. The detector has to establish the correspondences between the rig points \vec{P} and its image \tilde{p} acquired by all N_c cameras of a tuple. This has to be done on all the N_c images independently and regardless of the actual appearance. Since new images are easily recorded it is a suitable strategy to discard images where the correspondences cannot be obtained reliably.

Planar and Non-Planar Calibration Rigs On a planar calibration rig all calibration relevant points are situated on a plane. Rigs of this type can be manufactured very easily. The simplest version requires a printed version of the calibration pattern, adhesive tape and a table top. More expensive (and portable) version consist of composite structures and an e.g. photolithographically printed pattern.

Non-planar rigs exist in various forms. Usually rigidly connected planar rigs are used. One variant connects these planes by a right angle, another in a layered design.

There are two differences between planar and non-planar calibration rigs. On one hand, planar rigs are easier to manufacture. On the other hand the calibration results of non planar calibration rigs are more accurate [52]. All the following rig designs are explained for the planar version.

Regular Dot Pattern The first pattern type to be considered consists of a white background covered with black circular dots placed in a quadratic grid (Fig. 3.6(a)). The known coordinates of the points are the dot centres relative to one specific dot, e.g. the top left one. Other designs use white reflective dots on black background.

The major advantage of such a pattern is that the centre of the circles can be determined with excellent sub-pixel accuracy if the circles are large enough in the image [64]. Heikkilä [43] even includes a correction step for the distortion induced error of the centre coordinates resulting in a standard deviation of 0.02 pixel for the dot centre.

The major drawback of such rigs is that the orientation of multiple cameras can only be recovered with additional markers on the rig. The marker either takes up space which cannot be used for dots e.g. if the marker is a ring around certain points or it is likely

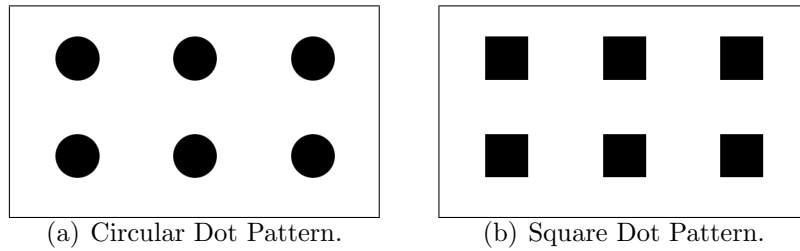


Figure 3.6: Calibration rigs using different regular dot patterns.

that the marker is not present the image (e.g. the marker is on the edge of the rig) in at least some images of a tuple. The latter is the more serious problem as this leads to compromises in rig placement: all markers in all images have to be visible.

A minor drawback is that circular dot pattern waste space on the rig: Only the black dots can be used for the point estimation, the white space in between is useless and therefore wasted. Usually the dots are circular and spaced along a grid line with $4r$ if r denotes the dot radius, thus at 50 % mark-to-space ratio. This leads to 80% white and therefore waste area on the rig.

Ring Code Dot Pattern The commercial system by Aicon [1] extends the dot pattern by introducing a rotation invariant, binary code placed in a ring around the dot. The binary code assigns a unique number to each point. Rigs are sold as carbon fibre planes with a set of ring code points on them. The spatial coordinates are stored in a text file accompanying the rig. The spatial coordinates are estimated to 0.001 mm accuracy during rig manufacturing.

The advantage of these rigs are the relative sturdiness of the carbon fibre planes, the accurate model coordinates and the robust point identification even for a partially visible rig.

The drawbacks are that the user of these rigs has to use the commercial software that cannot be adjusted to the specific needs of the application or improved in any way and the comparatively high price. Additionally the wasted space is much greater than in the case of the regular dot pattern.

Regular Squares Pattern Zhang [87] uses pattern of black squares on white background (Fig. 3.6(b)) in a regular grid on a planar surface, using the four corner of each square as the rig points.

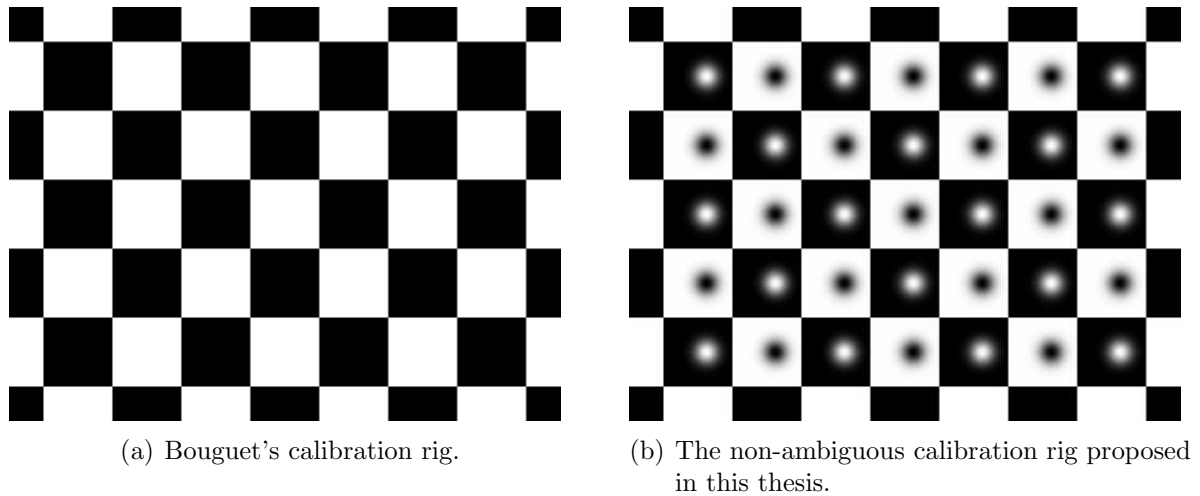


Figure 3.7: Calibration rig: chequerboard

The advantage of such a rig is that no space is wasted on the rig. The disadvantage is that orientation information has to be determined similar to the dot pattern, from specific points or from the edge.

Chequerboard Pattern Bouguet [11] suggests a chequerboard pattern. This pattern also does not waste rig space and the corners are found more reliably and efficiently compared to Zhang's rig: In man made objects rectangular corners are quite common, whereas the chequerboard corner is used in special cases only.

In its original form any orientation information has to be determined in similar ways to the circular dot pattern or the regular squares pattern. Fig. 3.7(a) depicts this kind of calibration rig.

We propose a new chequerboard design in this thesis, so information can be integrated into the rig (Fig. 3.7(b)). Each square is augmented with an eccentric marker in contrasting colour. The marker does not have sharply defined edges so that the corner detector does not recognise this structure as a chequerboard corner. This modification might be applied to Zhang's rig too.

Since the markers exist in all squares, even partially visible rigs contain sufficient information to reliably obtain the orientation of the rig. Only if a rig is partially visible such that complete rows or columns are missing the detection is ambiguous. This is due to the identical direction markers. If those are chosen such that adjacent squares form unique marker

pattern e.g. by varying form, size, and/or placement, even this drawback can be overcome. The *chequerboard rig* with identical markers will be used for calibration purposes.

In the following sections the parts of this chequerboard rig will be named:

square The black or white fields of the rig.

corner A point where four squares touch.

rig corner Outermost four corners.

3.2.1.2 Manual Chequerboard Detector

The MATLAB toolbox provides a manual calibration solution which prompts the user to click near the rig corners. While being completely out of scope for an automatic process it makes even the occasional calibration process very cumbersome and error prone: A good calibration needs about 20 images per camera. At four clicks per image (if no mistake was made), this amounts to 240 clicks with a precision of less than four pixels for a trinocular system.

This manual method counts the black/white crossings between the four corners to obtain the number of squares in the rig. Next the square corners are located. Then the user is prompted for an initial guess for the first distortion parameter of the camera model. To do so the user has to check the system's guess of the focal length which in turn is non-linearly connected to the distortion to be guessed. The guessed distortion is used to reproject the corner search seed points. The process ends on user's request. With a bit of training it is possible to learn a few rules of thumb how focal length and distortion are connected. These rules are lens and rig dependent. A trained user can process about one image per minute.

The actual sub-pixel accurate position is obtained using a modified Harris corner detector [42].

3.2.1.3 Polygon Fitting Based Automatic Chequerboard Detector

The OpenCV library [49] provides a chequerboard detection algorithm too. It is based on an initial scanning process with a subsequent sub-pixel accurate search of the corner. The initial scanning process operates on binary images, extracts the potential corner candidates

and tries to sort them with a polygonal approximation in a loop. If all the expected corners are determined the result is passed to the sub-pixel accurate location. This is implemented using a gradient minimum search with a relocation of a neighbourhood window until the centre keeps within a given threshold.

In practice this contour analysis turns out to be rather unstable. The OpenCV algorithm is not able to properly detect and sort the corners under outdoor or factory floor lighting conditions. In order to avoid singularities due to parallel image and calibration planes the calibration rig must be imaged at a certain obliquity. The corner analysis of the OpenCV implementation often fails if the angle or the camera-to-rig distance is too large, mainly due to an improper polygon approximation as we have noted in various experiments. Finally, the gradient based determination of the sub-pixel precise location of the chequerboard crossing is not satisfactory. From [59].

3.2.1.4 Hough Transform Based Automatic Chequerboard Detector

Two proprietary implementations based on cross-correlation template matching, line fitting and subsequent sub-pixel precise location estimation have been presented in [59]. They are the predecessors to the algorithm in the following section.

The first algorithm operates on the assumption of a fully visible calibration rig. Hence, it extracts a number of most prominent features, equal to the number of corners in the calibration rig, by means of cross correlation matching [4]. Subsequently, an outlier detection is performed based on Hough transform line detection and geometric constraint evaluation. Upon this, the lines are approximated by least-squares methods. Independent of the previously detected features the line intersections are used as coarse corner guesses and a maximum search followed by fitting a paraboloid to the correlation coefficients is performed.

The second method discards the assumption of a fully visible calibration rig. Instead of extracting a fixed number it selected features according to their reliability. The coarse feature sorting and outlier detection is performed as described above. After that, each possible rig corner is tested for reliability with appropriately scaled rig-corner templates. The final sorting is performed by accumulating features along line segments to least-squares approximated lines. At last the sub-pixel precise feature location is performed.

These two state-of-the-art algorithms are very robust with respect to occlusion, illumination, and noise. Due to the efficient implementation of the correlation process they are

fast enough for real time processing. The limit of usability is the assumption that the straight lines through the corners appear approximately straight in the image. Once the distortion effects are comparable to the corner spacing in the image this assumption cannot be upheld, it is only true for lenses with small distortion.

If one has to cope with wide angle lenses, the distortion increases. In the image the lens distortion make straight lines appear to be bent. This directly leads to blurring in the Hough accumulator of the first sorting stage compared to a lens with low distortion. As our experiments showed the blurred Hough accumulator either produces false-positive lines or thresholds will not be exceeded due to the fact that the peak and its environment is wider and less high. Obviously, this is directly connected to the fact that the Hough transform is a global algorithm. One way to cope with the distortion is to change from a global approach to a local approach as we propose in this thesis.

3.2.1.5 New Topology Based Automatic Chequerboard Detector

Due to the drawbacks of the aforementioned chequerboard detectors, we designed, tested, and evaluated a new rig finding algorithm in this thesis. We published an overview of the following in [59].

Overview Our new method is based on the aforementioned correlation coefficient between the image and a corner mask, which proved to be robust in various experiments, both indoor and outdoor. The local extrema are identified and located to sub-pixel accuracy by weighted mean or bivariate quadratic interpolation. Fig. 3.8 depicts an input image and its resulting correlation coefficient image. The image has been chosen to illustrate the capabilities of the algorithm even under extreme lighting conditions.

The major difference is the integration of these local extrema to a complete rig. Both the Hough transform and the new algorithm are bottom-up: Starting with atomic features, more complex entities are constructed, cumulating in the complete calibration rig.

The integration in this algorithm is done by topological methods. This approach is guided by one general principle: Prefer discarding the whole image to accepting a false positive identification of the corners. This strategy results from the fact that images are easy to acquire and errors are hard to cope with during subsequent processing stages.

The following steps are performed during the rig detection:

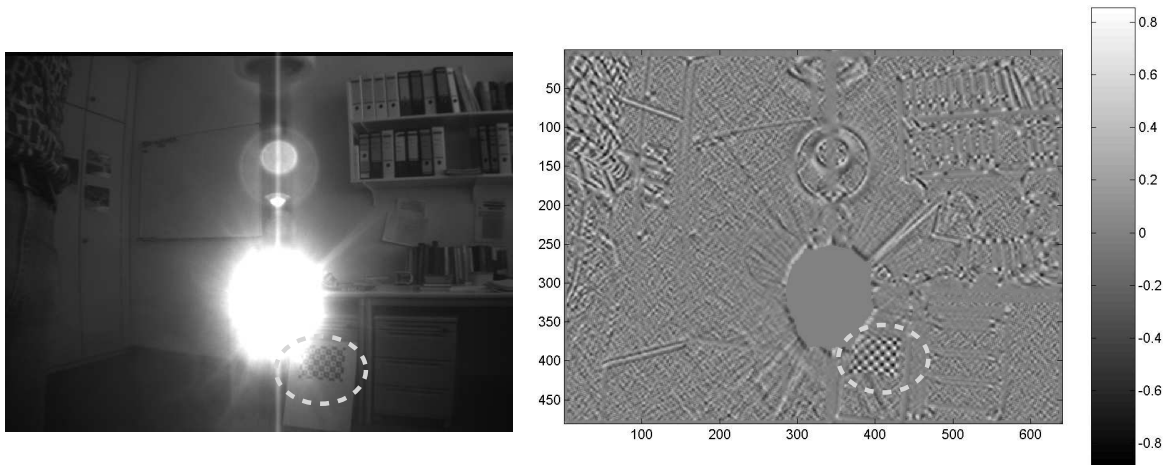


Figure 3.8: Example image of a calibration rig (left) and its correlation coefficient image (right). The rig is indicated by the dashed ellipse. Its top left corner is covered by the bright spot.

1. Compute image of correlation coefficient between input image and corner mask.
2. Detect corners at local extrema of the correlation coefficient image.
3. Construct a graph with vertexes at the local extrema and direction labelled (up, down, left, right) edges from maxima to minima and vice versa.
4. Filter for bi-directional edges, removing unidirectional edges and unconnected vertices.
5. Filter for edges with strong length differences, removing the longer edges and unconnected vertices.
6. Identify possible rig corners and enumerate the corners. Discard the image if an inconsistent enumeration is found.
7. (Only for rigs from Fig. 3.7(b)) Find marker directions and change enumeration accordingly.

Correlation Coefficient Image We want to find the corners by computing the correlation coefficient between image and corner template, implemented efficiently. Since the correla-

tion coefficient is a normalised value, both in terms of brightness and contrast, the actual grey values of the templates are not important. So they are chosen to be -1 for the black parts and +1 for the white parts.

The empirical correlation coefficient is defined as

$$c = \frac{\sum_i x_i y_i - \frac{1}{n} \sum_i x_i \sum_i y_i}{\sqrt{\left[\sum_i x_i^2 - \frac{1}{n} \left(\sum_i x_i \right)^2 \right] \left[\sum_i y_i^2 - \frac{1}{n} \left(\sum_i y_i \right)^2 \right]}} \quad (3.3)$$

where x_i is a grey value of the input image and $y_i \in \{-1, +1\}$ is a pixel of the template. It is assumed that the black and white areas in the template have an equal number of pixels thus $\sum_i y_i = 0$. With a total area of n pixels this simplifies Eq. 3.3 to:

$$c = \frac{\sum_i x_i y_i}{\sqrt{\left[\sum_i x_i^2 - \frac{1}{n} \left(\sum_i x_i \right)^2 \right] n}} \quad (3.4)$$

This leads to the following optimisation: The pixel sums $\sum_i x_i y_i$, $\sum_i x_i^2$ and $\sum_i x_i$ are computed using integral images. In an integral image each pixel contains the grey value sum of the top left sub image including the pixel itself. The sum of grey values in an rectangular area is computed by reading the values of the area's four corner pixels. Since the memory bandwidth is the prominent bottleneck in this function, we reduced the number of memory accesses per pixel. Obtaining numerator and the square of the denominator requires 13 memory read operations per correlation coefficient.

Additionally, the run time of the algorithm is independent of the size of the corner template. This allows for improvements in terms of detection robustness and sub-pixel accuracy: As soon as the integral images are computed, the recomputation of the correlation coefficient only takes very little time. One could try different mask sizes and select one or integrate the results of all of them.

Corner Detection Finding the corner candidates in the correlation coefficient image is performed by a non-maximum suppression followed by false-positive removal.

The non-maximum suppression is done by counting the number of pixels with a lower absolute value than the centre pixel in the eight-neighbourhood. If the count exceeds a threshold (currently 6) and the centre value exceeds another threshold (0.75), the pixel is assumed to be a corner candidate.

This non-maximum suppression provides a robust detection with a reasonable amount of false positives. The false positives are the neighbouring pixels of the true positive. Deciding which of the pixels is the true positive is done during the estimation of the sub-pixel accurate position. As soon as this position is available, it is used to determine the interpolated cross-correlation value. The candidate with the larger cross-correlation value is assumed to be the true positive.

Two algorithms for computing sub-pixel accurate positions are investigated: Weighted mean (WM) and bivariate quadratic interpolation (BVI). The sub-pixel position of WM is the mean of the eight-neighbourhood positions weighted by the corresponding correlation coefficients. The sub-pixel position of BVI is the location of the extremum of the bivariate quadratic function fitted to the correlation coefficient values, assuming it is sufficiently shaped and does not form a saddle point. The interpolated cross-correlation value is the function value at the extremum.

We investigate both these algorithms (WM and BVI) and the original manual rig finder in terms of accuracy and speed in Part III.

Graph Generation The following processing steps operate on a directed graph G defined as:

$$G = \{V, E\} \tag{3.5}$$

$$V = \left\{ \vec{v}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} \mid i = 1 \dots N_v \right\} \tag{3.6}$$

$$E = \left\{ \vec{e}_i = (s_i, t_i, d_i) \mid i = 1 \dots N_e; s_i, t_i \in 1 \dots N_v, d_i \in D = \{\text{left, right, up, down}\} \right\} \tag{3.7}$$

One observes that positive and negative correlation coefficient peaks interchange. Each positive peak has four negative neighbours directly connected along the black/white edges of the squares and vice versa. The first step is to identify this neighbourhood relation. Each corner candidate \vec{v}_i — estimated at sub-pixel accuracy above — is linked by edges

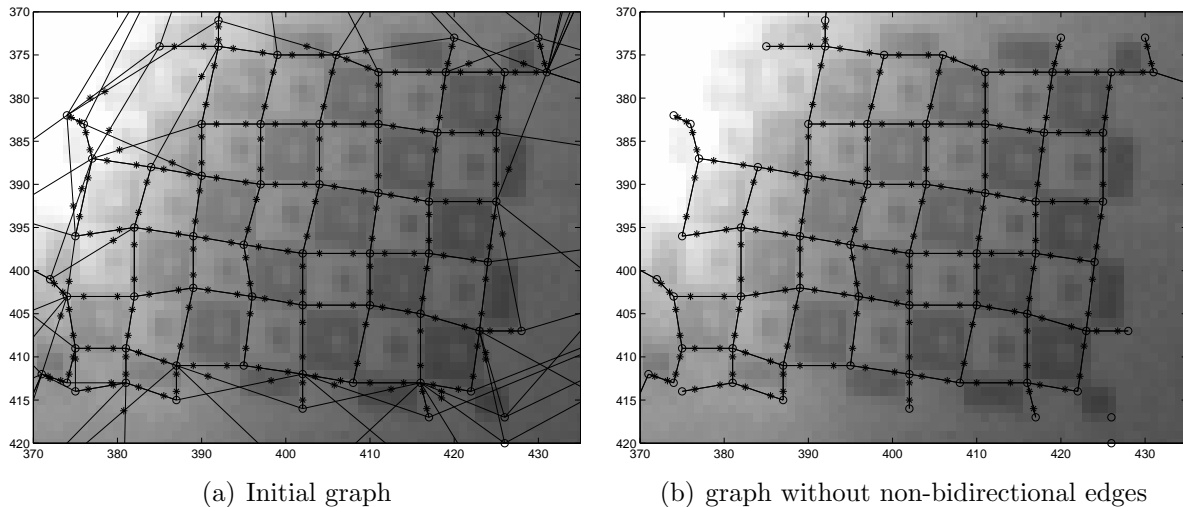


Figure 3.9: Results of the elimination of non-bidirectional edges. The images depict a zoom into the relevant part of Fig. 3.8, demonstrating the performance of the processing steps under difficult conditions. The asterisks denote the direction of the edges. They are placed at 80% distance towards the end of the edge.

\vec{e}_j — containing the numerical identifiers of the source (s_i) and target (t_i) vertex — to the four neighbours, labelled with the respective directions (left, right, up, down). This is implemented in the obvious way: $O(N^2)$ distances are computed from the N candidates and compared with respect to the candidates sign of the correlation coefficient. Fig. 3.9 shows this for a close up of the image from Fig. 3.8.

Non-Bidirectional Edge Elimination The first filter is used to eliminate non-bidirectional graph edges. Fig. 3.9(a) and 3.9(b) illustrate a situation where a false positive is eliminated this way. This procedure consists of deleting all graph edges $e = (s, t, d)$ subject to $(t, s, \text{opposite}(d)) \neq e$.

Edge Circle Filter The second filter checks for circles of length four. These circles in the graph map directly to the edges of the squares. Incomplete circles are present in e.g. Fig. 3.9(b), top row, leftmost corner and complete circles in Fig. 3.10(a).

The filter is implemented as a mark-and-sweep algorithm. The first run marks all corner candidates that are part of at least one circle (closed, non-reversing path in E) of length four. The second run eliminates all candidates that are not marked. The circle check is

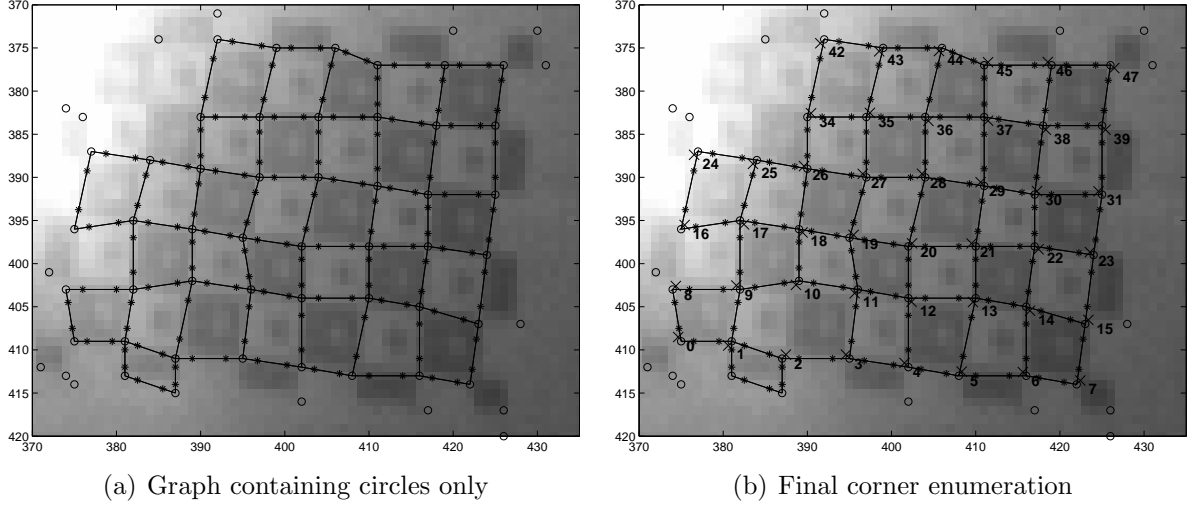


Figure 3.10: Results of circle filter and enumeration. The images depict a zoom into the relevant part of Fig. 3.8, demonstrating the performance of the processing steps under difficult conditions. The asterisks denote the direction of the edges. They are placed at 80% distance towards the end of the edge.

performed as follows: A vertex v which fulfils the conditions in Eq. 3.8 through 3.10 is marked. There is also a check which operates counter-clockwise instead of clockwise as done here:

$$e_1 = (v_0, v_1, d_1) \in E \mid \forall d_1 \in D \quad (3.8)$$

$$e_i = (v_{i-1}, v_i, d_i = \text{cw}(d_{i-1})) \in E \mid \forall i \in \{2, 3, 4\} \quad (3.9)$$

$$v_0 = v_4 \quad (3.10)$$

The functions $\text{cw}(d)$, $\text{ccw}(d)$ and $\text{opposite}(d)$ are defined as follows:

d	$\text{cw}(d)$	$\text{ccw}(d)$	$\text{opposite}(d)$
left	up	down	right
up	right	left	down
right	down	up	left
down	left	right	up

Edge Length Filter The third filter eliminates graph edges that have an exceptional length difference. The lengths are compared along one axis only, i.e. left-right and up-down. This avoids problems with extremely tilted rigs and false positives next to the rig (not shown here). The lengths $l_1 = \|v_s - v_{t_1}\|$ and $l_2 = \|v_s - v_{t_2}\|$ are obtained for each vertex with opposing edges $(s, t_1, d) \in E$ and $(s, t_2, \text{opposite}(d)) \in E$. If the condition

$$\begin{aligned} l_a &= \max(l_1, l_2) \\ l_i &= \min(l_1, l_2) \\ 1 - \frac{l_i}{l_a} &> \theta \end{aligned} \tag{3.11}$$

in Eq. 3.11 is fulfilled, the lengths are assumed to be so different that this cannot be a result of lens distortion or a slanted rig. The threshold θ (currently 0.4) is connected to the expected distortion. The larger the distortion, the smaller the threshold has to be.

Another run of the circle filter is performed to eliminate corners which have become unconnected due to length differences.

Corner Enumeration At this point the rig is assumed to be one component of the graph. The next step is to identify the component which describes the rig (Fig. 3.10(b)).

This algorithm iterates over all rig corner candidates. A rig corner candidate is identified as a vertex with exactly two links which form a 90 degree angle when undistorted, e.g. left and up. This can be checked efficiently by setting a bit for each direction and providing an array of flags which holds the information whether the bit combination interpreted as an integer number is a corner or not.

Starting from the rig corner candidate the links are followed in one direction. For each vertex visited this way all vertices connected in the perpendicular direction are counted. This process is done with both directions of the corner as the primary direction. The maximum number of visited vertices is noted.

Handling both directions of the corner is important to reach all corner candidates in the presence of missed corners or unseen corners when the rig is partially visible. Additionally it allows for checking if extraneous points are connected to the rig. If this happens, i.e. 10 links are found in one direction where 9 are expected, the last candidate is assumed to be a false positive.

The remaining corner candidates are identified using the horizontal (N_x) and vertical (N_y) square counts. The lower left corner number is set to 0, the upper right to $N_x N_y - 1$ etc. From these starting positions all other connected vertices are identified depending on their relative positions, e.g. going right increases the number by 1, going up by N_x . Additionally the position in the rig (row and column) is computed and compared to the position of the starting vertex. This avoids wrap-around effects and removes further false positives which were not found during previous processing steps.

If a corner identifier is assigned twice, both corners along with the respective edges are removed from the graph. This eliminates some rare errors where two corner candidates are generated such that they are not eliminated by other steps. If the enumeration does not yield the same result for all rig corner candidates used as starting points, the image is discarded.

During the traversal the numbers of vertices per component is counted. The largest component is assumed to be the rig. If the rig is rectangular the algorithm checks for the correct size in both axes, i.e. a rig turned by 90 degrees is discarded.

Marker Direction Detector The last step is to detect the direction marks on the rig and change the corner enumeration accordingly.

In order to detect the direction of each marker the grey values along two lines are extracted. These lines start in the middle of one square border and end in the middle of the opposite square border.

The grey levels on the horizontal line lead to a greater standard deviation than those on the vertical line if the rig is horizontal (as in Fig. 3.7). Additionally it is obvious that the weighted mean of these grey values yields a position that clearly detects them to be off-centre.

We threshold the standard deviation to detect missing markers if the processing options require them. In this case the image is discarded because expected markers cannot be found.

The direction of the rig is determined by computing the mean direction vector from the intersection points of the two lines to the centre of gravity along the line. The direction of the mean vector is quantised into four directions (left, right, up, down) and the corner identifiers obtained in the previous step are adjusted accordingly using the square counts.

3.2.2 Monocular Camera Calibration

In this section we briefly summarise the calibration procedure for both monocular and multiocular camera systems. The methods applied here are state of the art in photogrammetry and have been implemented using standard numerical algebra libraries.

3.2.2.1 Camera Model

In this thesis we use the camera model by Bouguet, which is inspired by Brown [13], Fryer [31] and Heikkilä [43]. The documentation of the MATLAB camera calibration toolbox [11] gives the details about this camera model.

The camera model characterises a calibrated camera by the following parameters.

$\vec{f} = [f_u, f_v]^T$ Focal lengths in pixel. There is a horizontal and a vertical length. The values \vec{f} is the product of the more common parameters focal length f in mm and pixel pitches s_u and s_v in pixel per mm. If the pixel pitch is unknown, there is no way to compute f , hence \vec{f} is computed.

$\vec{c} = [c_u, c_v]^T$ Principal point in pixel. This is the point on the chip where the optical axis intersects the chip. This is different to the definition of the pinhole camera above, but consistent with literature. The normal of the chip is assumed to be parallel to the optical axis. In practice this is not the case and the resulting errors are compensated as tangential distortions. Additionally the radial distortions are centred around this point.

α Skew coefficient. This value is the sine of the angle between pixel rows and pixel columns. For digital cameras the computation is disabled by default and if enabled yields values close to 1.

$\vec{k} = [k_1, k_2, k_3, k_4, k_5]^T$ The five distortion parameters. Two denote the tangential distortions, three denote radial distortions as forming a polynomial of degrees 2, 4 and 6.

The complete camera model

$$\vec{p} = \mathcal{P}(\kappa, \vec{P}) \quad (3.12)$$

transforms a point \vec{P} in the camera coordinate system to a point \vec{p} on the pixel grid. Due to the tangential distortions \mathcal{P} is not invertible in closed form. The vector $\kappa = [\vec{f}, \vec{c}, \alpha, \vec{k}]$ combines the camera parameters. One may see this as Eq. 3.1 extended by distortions.

3.2.2.2 Monocular Calibration Procedure

According to Bouguet’s approach the calibration procedure has three steps: initialisation, single camera calibration, and multiocular camera system calibration. Both the initialisation and the single camera calibration algorithm stem from Zhang [87] and Heikkilä [43].

The initialisation step performs a linear estimation of the focal lengths based on homographies. Additionally the initial poses \vec{R}_i, \vec{T}_i of the calibration rigs are obtained from similar operations.

The single camera calibration obtains the internal camera parameters and the poses of the calibration rigs by finding the local minimum of the objective function in Eq. 3.13 optimised by a Gauss-Newton algorithm:

$$\left[\tilde{\kappa}, \left\{ \tilde{\vec{R}}_i, \tilde{\vec{T}}_i \right\} \right] = \underset{\kappa, \{ \vec{R}_i, \vec{T}_i \}}{\operatorname{argmin}} \sum_i \sum_j \left(\tilde{p}_{ij} - \mathcal{P} \left(\vec{\kappa}, \vec{R}_i \vec{P}_j + \vec{T}_i \right) \right)^2. \quad (3.13)$$

The detected rig point j in image i is denoted by \tilde{p}_{ij} . It is compared with the projected rig point \vec{P}_j subject to the transform from rig coordinates to camera coordinates using \vec{R}_i, \vec{T}_i and the internal parameters.

3.2.2.3 Calibration and Pose Estimation

It is interesting to compare Eq. 3.13 to the pose estimation method in [2]. The 2D-3D pose estimation problem is one of the most important problems in monocular, three dimensional scene analysis. Quite a few approaches have been pursued in the past, most notably [2], its predecessor [63], and [28, 41].

These algorithms try to solve a sub-problem of calibration: Obtain the transformation from the object coordinate system to the camera system *using a known camera model*. During calibration one tries to obtain the same transformation *and the unknown camera model*.

The method in [2] solves the 2D-3D pose estimation problem exactly in the same way as if Eq. 3.13 would not modify the internal parameters κ . This fact leads to a completely dif-

ferent view of the camera calibration task. If pose estimation is sub-problem of calibration and object recognition, can calibration be seen as an object recognition task?

The answer is: Yes, camera calibration is a model-based object recognition and localisation task subject to specific constraints. The object to be found is cooperative: the calibration rig is designed to be found easily. Its presence has to be detected to discard faulty images. The rig pose has to be estimated along with other degrees of freedom: the internal parameters. The rig pose itself is usually of no interest, the internal parameters are the goal of the computation.

The rig itself is designed for best recognisability, which is not an option for most object recognition applications where the object cannot be changed. On the other hand, the accuracy requirements of a calibration are usually higher than those of object recognition in general.

An interesting field of investigation is opened by this insight: Which object recognition approaches can be extended to serve as a calibration procedure? The algorithms should be able to handle non-linear projections, sub-pixel accurate features and poor initialisations. Algorithms such as the Contracting Curve Density [38], Sec. 5.2; statistical object recognition [81]; and Iterative Closest Point methods [30] are probably best suited as they provide a large radius of convergence, even if they are top-down approaches which iteratively refine an initial pose. The method from Sec. 6.2 — newly developed in this thesis — is also applicable.

It is very likely that the integration of feature extraction and parameter estimation is computationally more expensive than the current scheme since iconic operations tend to be time consuming. It is also very likely that such an object recognition scheme will produce far more accurate results from fewer images. The reason for this is the fact that the current image processing concentrates on the square corners only, in other words the coordinates of less than one hundred points are used. The edges between the squares are completely neglected during processing although they account for at least one order of magnitude more coordinates.

Furthermore calibration in the form presented here is also sub-problem of Bundle Adjustment, the primary method of photogrammetry. As [76] states, methods that are known to photogrameters for more than a century are rediscovered by the computer vision community. We will revisit this connection in Sec. 3.2.4.

3.2.3 Multiocular Calibration

In order to define the so far independently calibrated cameras in one common coordinate system a further optimisation procedure is necessary. We therefore require a successful and correct monocular calibration of the cameras involved.

3.2.3.1 Coordinate Systems

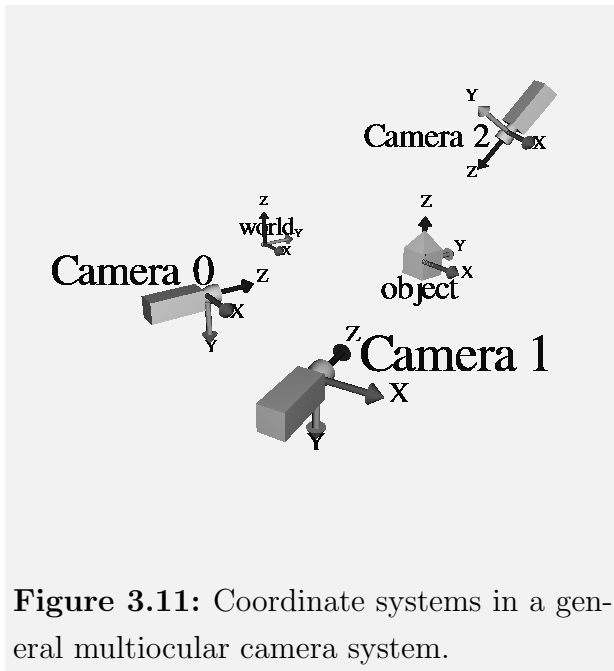


Figure 3.11: Coordinate systems in a general multiocular camera system.

Fig. 3.11 depicts the coordinate systems of a multiocular camera system with three cameras. One can see that the image of the object is upside-down in camera 2 compared to cameras 0 and 1. An external calibration has to account for this.

The coordinate systems of the individual camera are named *camera coordinate systems*. Assuming the cameras can be moved as a whole with respect to the object, the world, or both, the *sensor coordinate system* defines a reference frame between the cameras. Without loss of generality we assume that the sensor coordinate system is identical to the camera coordinate system of camera 0.

3.2.3.2 External Calibration by Bundle Adjustment

The camera system calibration according to Bouguet obtains the external camera parameters by finding the local minimum of a target function optimised by a Newton algorithm.

Additionally to the internal camera parameters, the positions and orientations of the camera coordinate systems with respect to each other will be estimated. These transforms use the coordinate system of camera 0 as their reference coordinate system. Doing so has the advantage that a transformation of points in the world coordinate system into the coordinate system of camera i can be performed by building a transformation chain: world — camera 0 — camera i . It is clear that such a convention forces the transformation of camera 0 to be the identity transformation.

Bouguet’s approach is to use a target function that optimises both the internal and external parameters in the same run. This works pretty well for cameras with normal fields of view (up to 30 to 45 degrees) and therefore small or moderate distortions.

For cameras with larger fields of view (e.g. 90 degrees) and thus larger distortions this combined approach leads to less accurate results. We observed in several calibration runs on two camera systems of this kind (10 cm and 24 cm base line) that the distortion parameters were reduced by the combined optimisation. The reason for this behaviour is rather obvious: As calibration requires images which show the rig in all cameras to establish the correspondences in different views of the same scene the calibration rig has to be far enough from the camera to be seen by all cameras.

The distortions as a function spanning the whole image cannot be determined properly if the noisy measurements cover only a small portion of the image. Fig. 3.12 illustrates this. It sketches the fit of a non-linear function to a small number of closely spaced points, leading in a poor signal-to-noise ratio.

Similar to the internal calibration this procedure is a Bundle Adjustment method. We modified it to optimise the external parameters only to overcome the aforementioned problems. The objective function of this minimisation is:

$$F = \sum_{ijc} (f_{ijc})^2 \quad (3.14)$$

$$f_{ijc} = \tilde{p}_i - \mathcal{P} \left(\tilde{\kappa}_c, {}^c_c H \cdot \vec{H}_j \cdot \vec{P}_i \right) \quad (3.15)$$

with \mathcal{P} being the projection function of the respective camera model as obtained by the internal calibration. The matrix \vec{H}_j denotes the transformation of the rig coordinate \vec{P}_i into the camera system coordinate system for image tuple j . The matrix ${}^c_c H$ denotes the transformation from the sensor coordinate system to the respective camera coordinate system. The variables to be optimised are ${}^c_c H$ and all \vec{H}_j . The point \tilde{p}_i was observed in camera c and identified as point i .

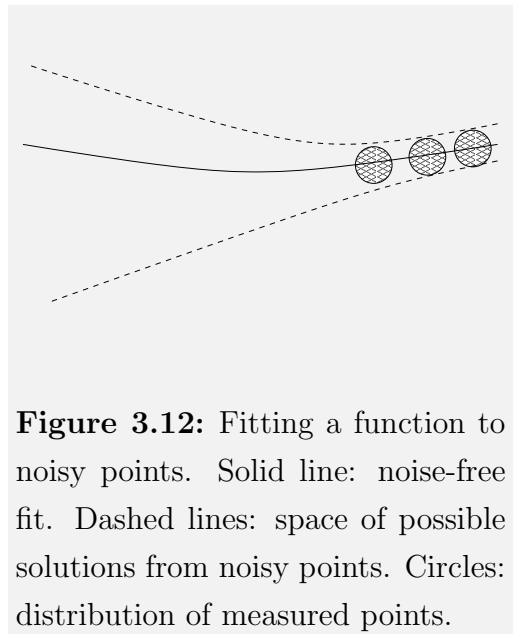


Figure 3.12: Fitting a function to noisy points. Solid line: noise-free fit. Dashed lines: space of possible solutions from noisy points. Circles: distribution of measured points.

A suitable initialisation can be linearly obtained by finding the transform between the spatial coordinates ${}^c\tilde{P}_i$ and ${}^0\tilde{P}_i$ of the calibration rig points as obtained in the individual camera calibrations.

The transforms 0_kH are split into rotation and translation. The rotation matrices are usually not orthonormal (esp. for strong distortions). But according to Haralick et al. [41] this is not a problem, because the singular value decomposition of the matrices, followed by setting the singular values to one, and reconstruction of the matrix yields a suitable initial orthonormal matrix.

3.2.3.3 External Calibration From Metric Rig Coordinates

A different approach to the calibration of the external camera parameters is to reuse the metric rig coordinates ${}^j\tilde{P}_i$ obtained during the internal calibration of the individual cameras. As the internal calibration is a bundle adjustment too, one can assume reasonable quality of the metric scene reconstruction.

Using the metric rig coordinates reduces the number of free parameters in the optimisation. Instead of optimising the few camera-to-camera transforms and the many camera-to-rig transforms only the camera-to-camera transforms are optimised. This may yield a better convergence behaviour as the dimensionality of the optimisation problem is drastically reduced.

However it is very likely that the optimum of this different objective function will be at a different position. We are therefore interested how well this method performs compared to Bundle Adjustment.

The most simple camera system that allows to derive the equations is one with three cameras, we therefore assume in this section $N_c = 3$. The resulting objective functions is to be found in Eq. 3.16:

$$F = \sum_i \left({}^0f_i^2 + {}^1f_i^2 + {}^2f_i^2 \right) \quad (3.16)$$

$${}^kf_i = {}^k\tilde{P}_i - \left({}^k\vec{R}^j\tilde{P}_i + {}^k\vec{T} \right) \quad (3.17)$$

Eq. 3.17 for $k = 2, j = 1$ has to be reformulated such that the transform ${}^2_1\vec{R}, {}^2_1\vec{T}$ is expressed

in terms of ${}^2_0\vec{R}, {}^2_0\vec{T}, {}^1_0\vec{R}, {}^1_0\vec{T}$ since those are the optimisation parameters. This is done using transformation chains [21], e.g. ${}^2_1\vec{R} = {}^2_0\vec{R} \cdot \left({}^1_0\vec{R}\right)^{-1}$.

The initialisation is done as stated in Sec. 3.2.3.2.

An extension of these equations to an arbitrary number of cameras is straight forward, as the objective functions for any N_c cameras will always consist of $N_c - 1$ functions Eq. 3.17 that can be used directly and $\frac{N_c(N_c-1)}{2} - (N_c - 1)$ functions where the transformation is to be derived using transformation chains.

For N_c cameras there will be a camera 0 and $N_c - 1$ other cameras. Obviously there are $N_c(N_c - 1)$ possible transformation between any given 2 cameras out of the N_c available. Half of the transformations describe the inverse of the other half. To describe all rigid connections between the camera one therefore needs $\frac{N_c(N_c-1)}{2}$ transforms. There are only $N_c - 1$ unique transformations as the other $\frac{N_c(N_c-1)}{2} - (N_c - 1)$ transformations can be built as a connection of any two unique transformations involving the camera 0 coordinate system.

This means that $N_c - 1$ transformations start at camera 0 and end at one of the other cameras. For any other camera there either is a transformation ending at camera 0 — which therefore is the inverse of one of the unique transformations — or there is a transformation chain no longer than 2 transformations, the intermediate coordinate system being that of camera 0. There exist many other transformation chains from one camera to the other, but only $\frac{N_c(N_c-1)}{2} - (N_c - 1)$ involve camera 0 in the middle.

3.2.4 Rig Placement Strategy

In order to obtain a precise calibration of the camera system, the position and orientation of the rig in each recorded image tuple has to be selected such that the resulting optimisation problem is well posed.

Seeing calibration as a form of bundle-adjustment, the rig placement strategy is identical to the camera placement strategy given a fixed set of object points that are (almost) always visible. The photogrammetric literature identifies this problem as *First Order Design*.

The only feasible way to deduce a placement strategy is simulation [66]. The simulation selects suitable initial camera positions and optimises a quality measure of the resulting calibration expressed as function of camera orientation and position. Whether the initial poses are selected according to a heuristic (as in [66]), randomly or on a multidimensional

grid is a matter of computing time. This bears similarities to [25], which might be applicable to the problem. Since the complexity of this subject is probably greater than that of this thesis, we have excluded a detailed investigation here.

Another interesting method may be used: *Third Order Design*, the improvement of an existing calibration by adding further images of rigs. This opens the opportunity to interactively improve the calibration: The user may be directed to move the rig to particularly promising positions and orientations.

In practice the following “rules of thumb” have been collected from the literature and from our own experience to yield a suitable set of calibration images:

- For each camera, the rig should fill the image. This captures the distortions.
- The rig should be shown at three different distances to the camera: At the beginning and the end of the working volume as well as in the middle.
- At the near end, the far end, and in the middle of the working volume, the calibration rig should be placed in the image corners.
- At each of those position the rig should be shown orthogonal to the optical axis and in four tilted orientations either axis-wise (up, down, left, right) or diagonal (up-left, up-right, down-left, down-right). The latter is slightly better according to Mason [66, Table 4.2]. The angle between optical axis and rig normal should be between 0 and 30 degrees.

This yields about 20–30 image tuples and should result in a calibration of reasonable accuracy. If the calibration stops with unusually large error bounds, more images are usually required. This is either achieved by fixing the rig at each position and recording a number of images to reduce the corner detection noise or capture intermediate positions.

Since the rig detector is fast enough to run on interactive frame rates, it is recommended to apply it to all incoming images and to store the corner coordinates. For small working volumes this results in about 200 image tuples in a matter of less than a few minutes according to the aforementioned rules and should ensure a correct calibration.

4 Appearance Based Methods

4.1 Multiocular Template Matching

Template matching is one of the oldest, and most established, appearance based object recognition methods. There are quite a few references concerning theory and practice of template matching, even hardware has been custom built for the purpose of template matching [34, 44]. We present here a novel multiocular definition of edge templates and an accompanying matching procedure where only the correlation function is taken from [35].

A template T is a possible appearance of the object to be recognised. It is assumed that there is a function $T(\vec{\Phi})$ that generates the template (usually by retrieving from a storage device) corresponding to the object pose $\vec{\Phi}$. In the simplest case a template can be a recorded image of the object, however templates are not limited to this.

Given an image I the template matching procedure in Eq. 4.1 identifies which template matches best according to a certain correlation function f .

$$\underset{\vec{\Phi}}{\text{best}} f(I, T(\vec{\Phi})) \quad (4.1)$$

Usually the function $T(\vec{\Phi})$ retrieves templates from a finite storage. The set of templates in this storage is $\mathcal{T} = [T_i]$. The template set is computed during phase 3 of Sec. 3.1.1 and used during phase 4.

It is clear that the number of templates can be quite large, depending on how different the object may appear in the image. The choice of template representation and correlation function is therefore crucial to run-time and memory consumption, as well as recognition robustness and the ability to distinguish objects. Obviously the first constraints contrast the latter, so compromises must be made.

4.1.1 Image Templates and Cross Correlation Matching

The classical template matching method [5] defines a template T_i to be an image of the same type (grey level, colour, ...) as I . A template is assumed to be a rectangular matrix of u_i by v_i pixels, n_i pixel in area. This means that each template may be of a different size.

Templates are either recorded images of the object (possibly using special hardware like turntables to realise the different appearances) or more recently rendered using methods of generative computer graphics.

Classically the object pose consists of the planar rotation and the position of the template's top left corner in the image, i.e. $\vec{\Phi} = [\Phi_\alpha, \Phi_u, \Phi_v]^T$. The template set contains rotated versions of the objects using a suitable quantisation of Φ_α .

The correlation function f is the empirical cross correlation coefficient depending on I , T_i , and $\vec{\Phi}$ as denoted in Eq. 4.2.

$$f(I, T_i) = \frac{\sum_{uv} I_{uv} T_{uv} - \frac{1}{n_i} \sum_{uv} I_{uv} \sum_{uv} T_{uv}}{\sqrt{\left[\sum_{uv} I_{uv}^2 - \frac{1}{n_i} \bar{I}^2 \right] \left[\sum_{uv} T_{uv}^2 - \frac{1}{n_i} \bar{T}^2 \right]}} \quad (4.2)$$

$$I_{uv} = I(\Phi_u + u, \Phi_v + v)$$

$$T_{uv} = T_i(u, v)$$

$$\bar{I} = \sum_{uv} I_{uv}$$

$$\bar{T} = \sum_{uv} T_{uv}$$

The properties of the method are:

- Memory consumption and run-time is fairly high.
- The object's texture, not only its geometry, is considered. This may lead to false-positive recognitions due to local maxima in f .
- Slight differences in appearance lead to strong differences in the values of f . This requires dense sampling of f and/or leads to reduced robustness of the recognition.

- Using the correlation coefficients above makes the matching theoretically invariant to global brightness and contrast. In practice the matching is very tolerant to strong changes if they do not change the relative grey values e.g. by shadows on the object.

4.1.2 Edge Templates and Chamfer Matching

A different template matching method is based on edge templates. This means that the appearance of intensity discontinuities of an object is used as the primary recognition feature. To achieve this, a grey or colour image of an object is obtained as described in Sec. 4.1.1 and an edge operator is applied to this image. The output of this edge operator is stored as T_i .

For the remainder of this section we assume that the edge orientation is quantised as N_o steps per 360 degrees. The notation $E(u, v, o)$ refers to a so-called demultiplexed edge image, where each layer o is of the same size as the input image, but contains only edge pixels of orientation o .

The template T_i is assumed to consist of n_i edge pixels. The definition of an edge template is therefore:

$$T_i = \{(u_{ij}, v_{ij}, o_{ij} \in [0, N_o - 1])\} \text{ with } j \in [0, n_i - 1] \quad (4.3)$$

The empirical correlation coefficient computed from T_i and the edge image of I cannot be used here, as it is very sensitive to the slightest shifts of the template. If the template is one pixel away from the correct position, f drops to very small values since the number of overlapping edge pixels is rather low. A smoother function is desirable to promote robustness and run-time efficiency. This is accomplished by applying the so called distance transform to the input image. Fig. 4.1 and Def. A.8 depict edge images and their distance transforms.

The distance transform $D = \text{DT}(E)$ of a feature image E is defined as an image of the same size as E . Each pixel (u_d, v_d) of D contains the distance d to the nearest feature position (u_e, v_e) according to a certain distance metric. The transform is computed independently for each orientation layer of E .

Since the Euclidean distance is too slow to compute in practice, it is approximated using fix-point numbers to the base of two. This assigns horizontal and vertical differences of 2 and a

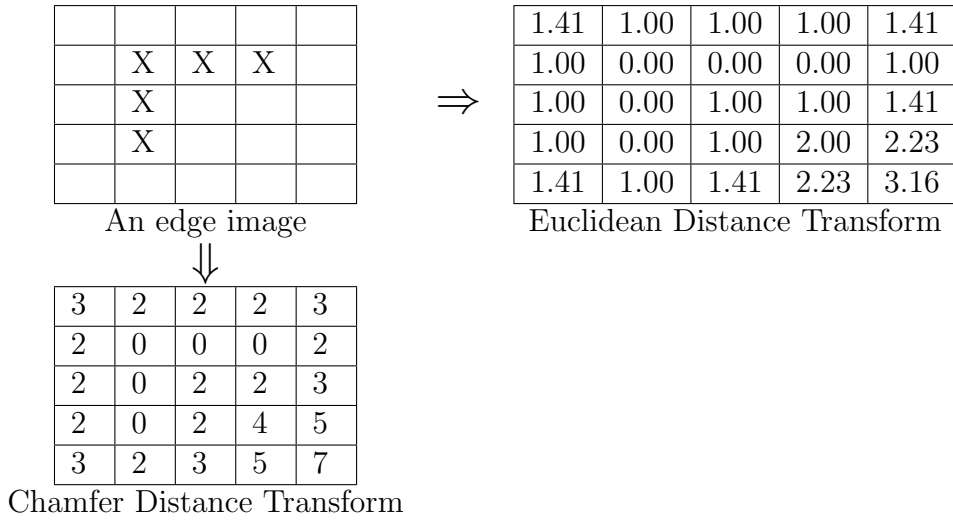


Figure 4.1: An edge image and its Euclidean and Chamfer distance transform.

diagonal difference of 3. Fig. 4.1 illustrates this. Since the minimum relation is transitive, a distance transform can be computed efficiently using a recursive algorithm [35].

A suitable correlation function f between image I and template T_i is defined [35] as:

$$f(D, T_i) = \frac{1}{n_i} \sum_j D(u_{ij}, v_{ij}, d_{ij}) \quad (4.4)$$

with $D = DT(E(I))$.

This correlation function is called the *Chamfer distance*. Its properties are:

- Memory consumption and run-time is fairly low.
- The object's edges are considered. This makes false positives less likely especially if the edges result from the objects silhouette only.
- Slight differences in appearance lead to slight differences in the values of f .

Due to these excellent properties the Chamfer distance is used throughout this thesis.

4.1.3 Multiocular Evaluation Strategy

Another important point to be considered is the evaluation strategy of f from Eq. 4.4 to find its local minima, which are assumed to correspond to real world objects. Since classical

template matching is a 2D method, its evaluation strategy cannot be used efficiently for multiocular object localisation. We will show that in the following.

The classical evaluation strategy [35] consists of a hierarchical evaluation of f depending on the *image based degrees of freedom*, $IDoF$ (pixel position of the template centre, scale, rotation). It starts by evaluating f at a coarse resolution of the IDoF and refines at promising positions.

Different views of the object are handled in [35] by a hierarchical clustering based on similar appearances. This is appropriate for objects with many internal degrees of freedom such as pedestrians as in [35], but yields a complicated template generation and matching procedure. In another, unpublished in-house implementation, this was replaced by a combined coarse-to-fine-search in pose space (3 rotations and distance to camera) and image space. We have extended this to a coarse-to-fine-search in the application relevant space.

The simplest extension to multiple images performs the matching in each image independently and establishes correspondences between the matching results of different images. This works fairly well if only one matching template is found in each image (e.g. object at the top of Fig. 4.2). If there are more matches (e.g. the two objects at the bottom of Fig. 4.2) these correspondences may be ambiguous. It is not clear which two of the four combinations are the correct ones. Only external constraints, e.g. admissible distance to the camera and therefore admissible disparity between left image position and right im-

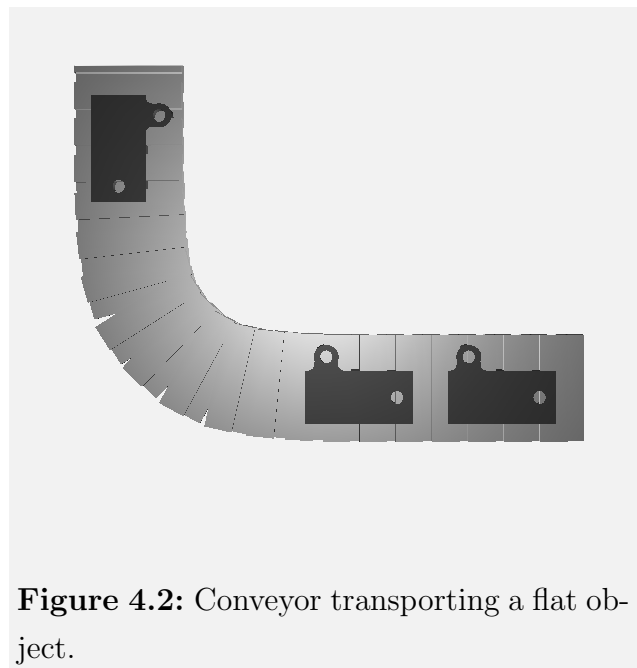


Figure 4.2: Conveyor transporting a flat object.

age position, can be used to disambiguate the results. The algorithm is therefore identical to the problem of feature-based stereo with the object templates as features.

Fig. 4.2 illustrates another disadvantage of this simple extension of 2D matching: Far more templates than necessary have to be matched resulting in poor run-time performance. In this example the object may only appear on the conveyor. The *application specific degree of freedom*, $ADoF$ of the object to be localised here is exactly one: The position of the

object along the conveyor's transportation direction. The unmodified 2D matching method requires three IDoF: translation and planar rotation. This leads to a run-time complexity of $O(n^3)$ template matches versus $O(n)$ for ADoF with n templates per degree of freedom. An improvement would be to match only those appearances of the object that may actually appear: The poses which place the object on the conveyor. To do so, we simulate the appearance of the object by varying the ADoF in world coordinates and projecting it to each camera. This *multiocular template* actually consists of N_c individual templates which are evaluated together:

$$T_i = ({}^0T_i, {}^1T_i, \dots, {}^{N_c-1}T_i) \quad (4.5)$$

where cT_i denotes the template from one camera c out of N_c cameras total. From now on all templates are assumed to be of that type, but for simplicity they are indexed as if they were monocular (Eq. 4.3). This also maps well to the actual implementation: If the N_c images are placed in a larger meta-image (e.g. below each other at consecutive addresses), the multiocular extension adds more feature positions to a template that now refers to the meta-image. The actual template matching algorithm is identical for monocular and multiocular images.

The matching is done by hierarchically evaluating f (Eq. 4.4) depending on the DoF of the application (ADoF). Evaluating f depending on ADoF is far more robust than evaluating it using image space templates: Since the ADoF are defined with respect to the world or sensor coordinate system, the corresponding templates describe an *actual* object pose, not a *hypothetical* one which has to be verified. Therefore, problematic cases in 2D matching such as repeated objects are avoided by not testing for those object occurrences.

The ADoF dependent templates are in fact three dimensional templates. They implicitly perform a triangulation of corresponding points during matching. If the image tuple depicts the same pose as a template, all points on the object can be recovered. The method therefore performs localisation and recognition at the same time. A template only matches well if it recognises the object at the correct position.

4.1.4 Lattice of Poses

In practical applications the ranges of the DoF are finite. We will denote this range of values for the DoF *pose space* and impose a quantisation for each of the N_d dimensions

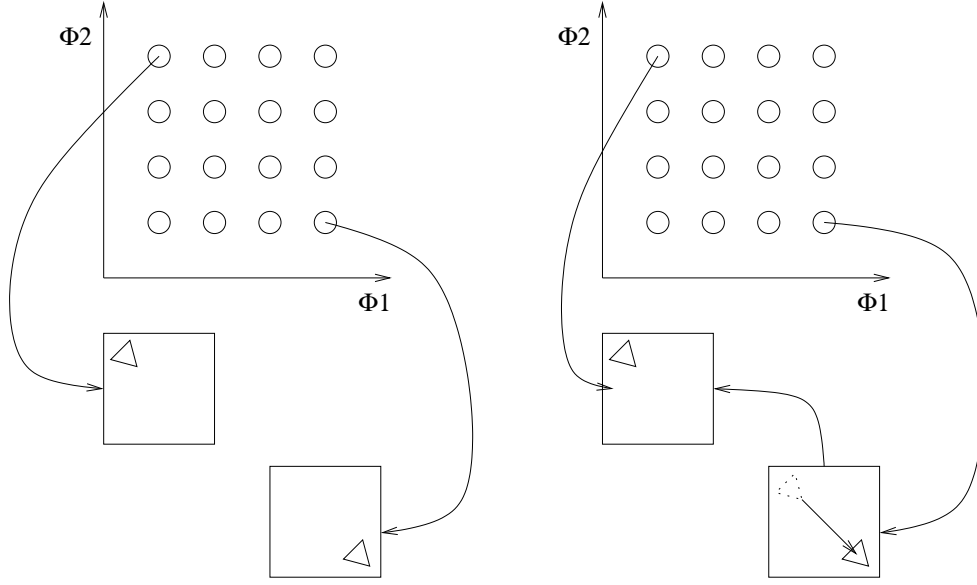


Figure 4.3: Lattice (top) and templates (triangles). Left: direct pose-template association. Right: similarity pose-template association. Details: see text.

(degree of freedom). Such a quantised space can be represented most conveniently as a multidimensional grid (math.: lattice), i.e. equidistantly spaced values in arbitrary many dimensions.

We assume that one dimension $d \in [1, N_d]$ of this lattice corresponding to the ADoF d is limited to the interval $[\underline{\Lambda}_d, \overline{\Lambda}_d]$ with a step size of σ_d . The handling of object symmetries is done by limiting the rotational DoF accordingly. In Fig. 4.3 each lattice has an $N_d = 2$ with 16 poses.

Now that a range of values for each component of the pose is defined, Eq. 4.4 can be evaluated on the lattice. Given a point on the lattice the template associated with this point must be retrieved in order to compare this template with the image. The straightforward method of associating one lattice point \vec{l} with exactly one template is called $T_{\vec{l}}$ *direct pose-template association*, *dPTA*. The advantage is the straightforward implementation of pose-template and template-pose association as seen in Algorithm 4.1. The corresponding retrieval function that is used during the matching phase is seen in Algorithm 4.2. A possible implementation using pointers is depicted in left part of Fig. 4.3. Each lattice point (circles) points to a different template data structure.

The drawback here is the memory consumption: As current 32 bit operating systems allow

at most 3 GB of memory per process*, one either has to conceive a memory management method or reduce the number of templates. 3 GB may seem like a lot of memory, but if the templates for all cameras together require about 3 kB this yields roughly one million templates. Given six DoF this makes just ten steps per DoF. One template usually contains about 1000 edge pixels, this limits the system to three cameras. An additional point to keep in mind is that the memory transfer rate becomes the limiting factor in terms of processing performance.

The memory consumption can also be reduced by a more sophisticated implementation of the pose-template association, the *similarity pose-template association*, *sPTA*. The idea is to avoid storing similar templates that differ only by the positions in the image, as depicted in the right part of Fig. 4.3.

Instead, some templates are stored completely (top left lattice point in Fig. 4.3) and other poses (bottom right lattice point in Fig. 4.3) refer to the stored template and the resulting offset in the image (Algorithm 4.3). Thus templates that differ very little from an already stored template require very little extra memory.

A template is stored as a link if the result of Eq. 4.4 is below a threshold θ_e after compensation of the position of the templates in the image. The corresponding retrieval function that is used during the matching phase is seen in Algorithm 4.4.

Algorithm 4.1 Template Generation for Direct Pose-Template Association

Variable	Type	Description
N_d	Input	Number of lattice dimensions (ADof)
$\underline{\Lambda}_d, \bar{\Lambda}_d$	Input	Limits of lattice dimension d
σ_d	Input	Step size of lattice dimension d
\mathcal{T}	Output	Template set
\vec{l}	State	Index of current pose
$T_{\vec{l}}$	State	Template of current pose p

- 1: **for** $l_d \in [\underline{\Lambda}_d : \sigma_d : \bar{\Lambda}_d] \Big|_{d=1}^{N_d}$ **do**
 - 2: Render template $T_{\vec{l}} = \left\{ \left(u_{\vec{l},i}, v_{\vec{l},i}, o_{\vec{l},i} \right) \right\}$
 - 3: **end for**
 - 4: $\mathcal{T} \leftarrow \{T_{\vec{l}}\}$
-

*Microsoft Windows NT: 300–500 MB, Windows XP: < 2 GB, Linux: < 3 GB

Algorithm 4.2 Template Retrieval for Direct Pose-Template Association

Variable	Type	Description
\vec{l}	Input	Index of template to be retrieved
\mathcal{T}	Input	Template set
$T_{\vec{l}}$	Output	Retrieved template

1: Read $T_{\vec{l}}$ from \mathcal{T}

Algorithm 4.3 Template Generation for Similarity Pose-Template Association

Variable	Type	Description
N_d	Input	Number of lattice dimensions (ADof)
$\underline{\Lambda}_d, \bar{\Lambda}_d$	Input	Limits of lattice dimension d
σ_d	Input	Step size of lattice dimension d
θ_e	Parameter	Maximal approximation error [pix]
\mathcal{T}	Output	Template set
\vec{l}	State	Index of current pose
$T_{\vec{l}}$	State	Template of current pose p
k	State	Index of most similar real template
e	State	Approximation error of current template [pix]

1: $\mathcal{T} \leftarrow \emptyset$
2: **for** $l_d \in [\underline{\Lambda}_d : \sigma_d : \bar{\Lambda}_d] \Big|_{d=1}^{N_d}$ **do**
3: Render template $T_{\vec{l}} = \left\{ \left(u_{\vec{l},i}, v_{\vec{l},i}, o_{\vec{l},i} \right) \right\}$ with projected origin $(x_{\vec{l}}, y_{\vec{l}})$
4: Find neighbouring templates $\left\{ T_{\vec{j}} \right\}$ in pose space with projected origins $\left\{ \left(x_{\vec{j}}, y_{\vec{j}} \right) \right\}$
5: $\vec{k} = \underset{\vec{j}}{\operatorname{argmin}} f(\operatorname{DT}(T_{\vec{m}}), T_{\vec{l}})$, with $T_{\vec{m}} = \left\{ \left(u_{\vec{j},i} - x_{\vec{j}} + x_{\vec{l}}, v_{\vec{j},i} - y_{\vec{j}} + y_{\vec{l}}, o_{\vec{j},i} \right) \right\}$
6: $e = f(\operatorname{DT}(T_{\vec{m}}), T_{\vec{l}})$ with $T_{\vec{m}} = \left\{ \left(u_{\vec{k},i} - x_{\vec{k}} + x_{\vec{l}}, v_{\vec{k},i} - y_{\vec{k}} + y_{\vec{l}}, o_{\vec{k},i} \right) \right\}$
7: **if** $e < \theta_e$ **then**
8: $T_p \leftarrow \left(\vec{k}, x_{\vec{l}}, y_{\vec{l}} \right)$ {Linked template}
9: **else**
10: $T_p \leftarrow T_{\vec{l}}$ {Reference template}
11: **end if**
12: **end for**
13: $\mathcal{T} \leftarrow \{T_p\}$

Algorithm 4.4 Template Retrieval for Similarity Pose-Template Association

Variable	Type	Description
\vec{l}	Input	Index of template to be retrieved
\mathcal{T}	Input	Template set
$T_{\vec{l}}$	Output	Retrieved template

```

1: if  $\vec{l}$  is a linked template then
2:   Read  $(\vec{k}, x_{\vec{l}}, y_{\vec{l}})$  from  $\mathcal{T}$ 
3:   Read  $T_{\vec{k}}$  from  $\mathcal{T}$ 
4:    $T_{\vec{l}} = \left\{ \left( u_{\vec{k},i} - x_{\vec{k}} + x_{\vec{l}}, v_{\vec{k},i} - y_{\vec{k}} + y_{\vec{l}}, o_{\vec{k},i} \right) \right\}$ 
5: else
6:   Read  $T_{\vec{l}}$  from  $\mathcal{T}$ 
7: end if

```

4.1.5 Hierarchical Multiocular Matching Procedure

The matching can be efficiently performed according to the following procedure. Without loss of generality we assume an N_d dimensional lattice with the limits $\underline{\vec{\Lambda}}$ and $\overline{\vec{\Lambda}}$ as during template generation. If there is more than one lattice per object type and/or more than one object type the other lattices can be evaluated in the same manner.

The Chamfer distance is a very smooth function. It has local minima only if other structures besides the object are visible in the edge image. Since we only know that these local minima are valid objects if the Chamfer distance is small enough, they have to be evaluated. Neighbourhoods of lattice points that are not sufficient similar to the object do not need to be inspected.

The criterion for a successful match is two-fold: A certain percentage of template pixels has to overlap with the image, i.e. the distance of so many template pixels must be below a threshold. The lattice is checked first with a small overlap, large admissible distance and coarse steps. This non-linear function is taken from the aforementioned unpublished implementation, where it proved to be a very robust method.

The steps are defined in multiples of the lattice resolution, which is an improvement over the aforementioned unpublished implementation, as it enables the use of ADoF instead of IDoF. A refinement of these matching poses is done by increasing overlap, reducing admissible distance and step size. Algorithm 4.5 details the matching procedure.

The total complexity in units of template evaluations is $O\left(\prod_{d=1}^{N_d} O_d\right)$. The DoF-wise com-

Algorithm 4.5 Hierarchical Multiocular Template Matching Procedure

Variable	Type	Description
N_l	Input	Number of hierarchy levels
N_d	Input	Number of lattice dimensions
$\underline{\Lambda}_d, \sigma_d, \bar{\Lambda}_d$	Input	Limits and step size of lattice dimension d
$D(u, v, o)$	Input	Demultiplexed distance image
\mathcal{T}	Input	Template set
θ_l	Parameter	Maximal Chamfer distance.
μ_l	Parameter	Minimal percentage of matching pixel.
$s_{ld} \in \mathcal{N}$	Parameter	Grid scaling factor
\mathcal{P}_{N_l-1}	Output	Set of matching poses
l	State	Current hierarchy level
\mathcal{P}_l	State	Set of promising poses in level l
\vec{p}	State	Index of pose candidate to be refined.
$n_{\vec{p}}$	State	$n_{\vec{p}} = T_{\vec{p}} $
$u_{\vec{p},i}, v_{\vec{p},i}, o_{\vec{p},i}$	State	Coordinates of feature i in $T_{\vec{p}}$

```

1:  $\vec{p}_0 \leftarrow \{\}$ 
2: for  $p_d \in [\underline{\Lambda}_d : s_{0d}\sigma_d : \bar{\Lambda}_d] \Big|_{d=1}^{N_d}$  do
3:   Retrieve  $T_{\vec{p}}$  from  $\mathcal{T}$  (Algorithm 4.2 or 4.4)
4:    $\zeta \leftarrow \{i | D(u_{\vec{p},i}, v_{\vec{p},i}, o_{\vec{p},i}) < \theta_0\}$ 
5:   if  $\frac{|\zeta|}{n_{\vec{p}}} > \mu_0$  then
6:      $f \leftarrow \frac{1}{n_{\vec{p}}} \sum_{i \in \zeta} D(u_{\vec{p},j}, v_{\vec{p},j}, d_{\vec{p},j})$ 
7:      $\mathcal{P}_0 \leftarrow \mathcal{P}_0 \cup \{(\vec{p}, f)\}$ 
8:   end if
9: end for
10: for  $l = 1$  to  $N_l - 1$  do
11:    $\mathcal{P}_l \leftarrow \{\}$ 
12:   for  $\vec{p} \in \mathcal{P}_{l-1}$  do
13:     for  $t_d \in [p_d - \frac{1}{2}s_{l-1,d}\sigma_d : s_{ld}\sigma_d : p_d + \frac{1}{2}s_{l-1,d}\sigma_d] \Big|_{d=1}^{N_d}$  do
14:       Retrieve  $T_{\vec{t}}$  from  $\mathcal{T}$  (Algorithm 4.2 or 4.4)
15:        $\zeta \leftarrow \{i | D(u_{\vec{t},i}, v_{\vec{t},i}, o_{\vec{t},i}) < \theta_l\}$ 
16:       if  $\frac{|\zeta|}{n_{\vec{p}}} > \mu_l$  then
17:          $f \leftarrow \frac{1}{n_{\vec{p}}} \sum_{i \in \zeta} D(u_{\vec{t},i}, v_{\vec{t},i}, d_{\vec{t},i})$ 
18:          $\mathcal{P}_l \leftarrow \mathcal{P}_l \cup \{(\vec{t}, f)\}$ 
19:       end if
20:     end for
21:   end for
22: end for

```

plexities O_d are computed as follows. n_d is the number of grid points in dimension d of the lattice. The best case complexity is $O_d = O\left(\frac{n_d}{s_0}\right)$. This is the case if no object could be found at the top level. The worst case complexity is $O_d = O(n_d)$. In this case all templates have to be evaluated. This will only happen with poor choice of the thresholds.

The average case complexity largely depends on the average scene and the number of real-world objects in it. One has to at least acquire one sample per object. Therefore the step size should be about the object size in that degree of freedom. On the other hand this requires a large admissible Chamfer distance, allowing for a variety of objects to be the cause for further computations at finer levels. Therefore the best number of resolution levels N_l , thresholds (θ_l, μ_l) and step size s_l is a matter of experimentation.

4.2 Feature Pose Maps

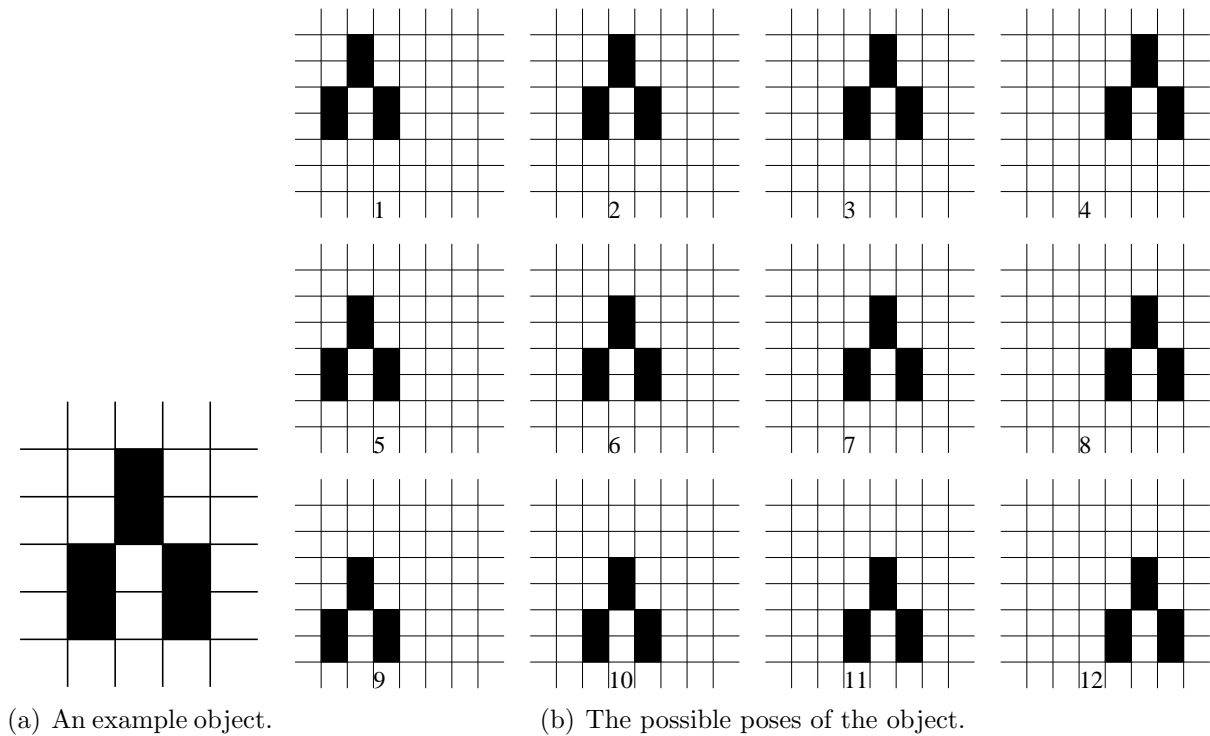
4.2.1 Method Description

Another 2.5 dimensional monocular object recognition and localisation method operates on so-called Feature Pose Maps [10, 82]. A Feature Pose Map (FPM) is a table in the size of the image, containing one layer per feature type. Each cell in this three dimensional table contains a list of the poses that created a feature of this type at this pixel position (Fig. 4.4).

When an image is to be analysed the features are computed and a vote is counted for each pose in a cell where there is a feature (Fig. 4.5). The poses with a sufficient number of votes are considered the recognition results. The integration of new poses and/or new objects is possible by extending the FPM or the pose lists. Algorithm 4.6 details the matching procedure.

FPMs can be generated in the same way as ordinary templates can. In fact the respective data structures of edge templates as defined in this thesis are transferable from FPM to template and back without loss. This might not be the case for other matching algorithms.

Templates generated with direct pose template association (dPTA, Sec. 4.1.4) can be matched using the aforementioned simple algorithm, those generated with similarity pose template association require a more elaborate matching procedure that operates on overlapping sub images. For the sake of simplicity we consider dPTA templates only.



		1	2	3	4	
		1,5	2,6	3,7	4,8	
1	2,5,9	1,3,6,10	2,4,7,11	3,8,12	4	
1,5	2,6,9	1,3,5,7,10	2,4,6,8,11	3,7,11,12	4,8	
5,9	6,10	5,7,9,11	6,8,10,12	7,11	8,12	
9	10	9,11	10,12	11	12	

(c) The Feature Pose Map.

Figure 4.4: Construction of a Feature Pose Map. The Feature Pose Map (c) contains in each cell the identifiers of those poses in (b) that have this pixel set.

FPM matching is a form of generalised Hough transform [24]. It is therefore a bottom-up method, integrating iconic low-level features to objects. The corresponding top-down method is of course template matching, as it assumes objects and tries to verify them using low-level features.

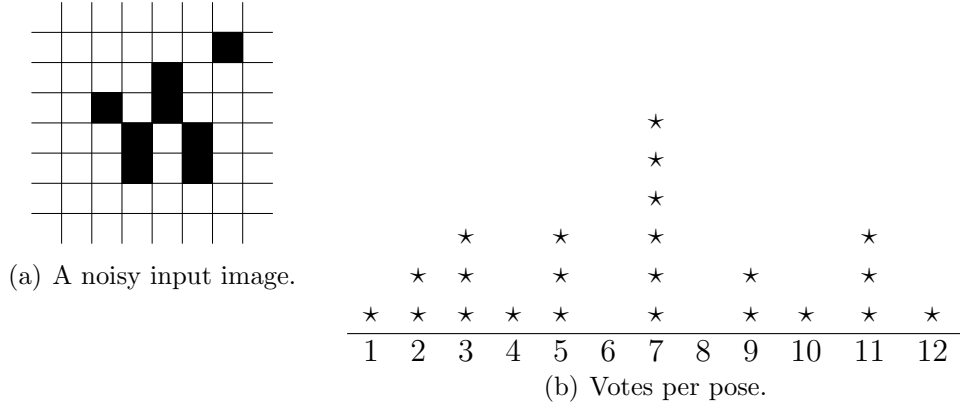


Figure 4.5: FPM from Fig. 4.4 matching a noisy image. The histogram (b) is generated by counting one vote for all entries in those cells of Fig. 4.4(c) that correspond to a black pixel in (a). Ground truth is pose # 7.

Algorithm 4.6 Feature Pose Map Matching Procedure

Variable	Type	Description
N_p	Input	Number of poses in hyper grid
n_p	Input	Number of features in pose p
N_t	Input	Number of feature types
$F = \{(u_i, v_i, t_i \in [1, N_t])\}$	Input	Features as extracted from image
$\{p\} = M(u, v, t)$	Input	Feature pose map.
θ	Parameter	Percentage of votes for a successful match.
$P = \{p\}$	Output	Set of matching poses.
V_p	State	Votes of pose p
$f = (f_u, f_v, f_t \in [1, N_t])$	State	Current extracted feature to vote for

```

1: for all  $p \in [1, N_p]$  do
2:    $V_p \leftarrow 0$ 
3: end for
4: for all  $f \in F$  do
5:   for all  $p \in M(f_u, f_v, t_t)$  do
6:      $V_p \leftarrow V_p + 1$ 
7:   end for
8: end for
9:  $P = \{p \in [1, N_p] \mid \frac{V_p}{n_p} > \theta\}$ 

```

4.2.2 Distance Voting

Generalised Hough Transform considers the position of a feature to be noise-free. In real-world applications the feature positions are noisy, because camera images are subject to quantisation and digitalisation noise in the sensor chip. The feature position noise will therefore cause noise in the accumulator, leading to less well defined peaks.

The methods [10, 82] suggest a user defined quantisation of the pose and feature spaces to avoid oversampling or undersampling the feature space. The underlying problem is that — just like in correlating edge images — the correlation function of the unscaled feature space is very sensitive to the slightest disturbances.

Therefore we have developed a new voting process such that the number of votes is inversely proportional to the distance to the nearest feature. More votes would be accumulated for exact matches while slight disturbances result only in slight changes of the number of votes. The accumulator content should therefore be more robust to feature position noise and the size of the pose sampling grid.

This new voting scheme leads to an increase in run-time, but not memory consumption. If the feature positions to cast votes are limited to strips of a small maximal distance value, the slow-down is negligible.

4.2.3 Multiocular Extension

The extension to multiocular images proceeds analogous to template matching: Each camera of a tuple has its own feature-pose map, but the pose votes are cast into a single accumulator. The modification to Algorithm 4.6 are straightforward and will therefore not be listed here.

This extension has the favourable properties of a linear increase in both memory and run-time consumption with a growing number of cameras.

5 Segmentation Methods

5.1 Multiocular Active Contours

In this section we describe another object recognition algorithm, based on a well-known image segmentation method, Active Contours or *snakes* [12, 45, 68, 83]. Traditionally, snakes are two-dimensional parametric curves, that deform during the image processing procedure to approximate the segmentation boundary. The deformation is directed by the image content such that successive iterations approximate the object boundary better than the preceding ones.

There are quite a few publications concerning modifications and extension of the original snake approach. So far the majority of algorithms uses snakes only in the dimension of the input data, i.e. two-dimensional curves for images and three-dimensional surfaces for volumetric, e.g. tomography, data.

Only a minority of publications describe three-dimensional curves, that are projected into two-dimensional images. These algorithms are all limited to stereo images, but have not been investigated for the general case of multiocular images.

We use snakes for object recognition also by projecting the curve, parametrised by the pose, to the images of a tuple and improve the match between model curve and image content. Object classification can only be done by e.g. checking the object pose against a set of values that distinguish the different classes (e.g. flexible tubes with different diameters).

Most of the work in this section was published in our paper [23], which itself is an excerpt from a master thesis [22] that was supervised by the author of this doctoral thesis.

5.1.1 Active Contours

Active contours or snakes have been used extensively for segmentation [12, 45, 68, 83] and tracking [9, 53] purposes. In the Kass approaches [53], the basic snake is a parametric

function \vec{p} , representing a contour curve or model:

$$\vec{p} = \vec{v}(s), \forall s \in [0, l] \quad (5.1)$$

where \vec{p} is a contour point at parameter s . An energy function F is minimised over the contour $\vec{v}(s)$:

$$F = \int_0^l E_{\text{snake}}(\vec{v}(s)) ds \quad (5.2)$$

E_{snake} can be separated into four terms:

$$E_{\text{snake}}(\vec{v}(s)) = \alpha E_{\text{cont}}(\vec{v}(s)) + \beta E_{\text{curv}}(\vec{v}(s)) + \gamma E_{\text{ext}}(\vec{v}(s)) + \delta E_{\text{con}}(\vec{v}(s)) \quad (5.3)$$

The internal energy $E_{\text{int}} = \alpha E_{\text{cont}}(\vec{v}(s)) + \beta E_{\text{curv}}(\vec{v}(s))$ regularises the problem and favours a smooth contour.

The external energy E_{ext} , is based on the image at $\vec{v}(s)$ and links the contour with the image. In this thesis, the gradient magnitude of the image I is used: $E_{\text{ext}} = \left| \nabla I(\vec{v}(s)) \right|$.

The term E_{con} is used by the original snake [53] to introduce constraints, for example linking of an active contour point to other contours or springs. User interaction can also be cast into E_{con} . Later on, balloon snakes have emerged [19], where E_{con} is used to “inflate” the active contour, to counteract the shrinking introduced by the original E_{int} .

The weight factors α , β , γ and δ can be chosen based on the application.

Disadvantages of the parametric model includes its dependence on parametrisation, which can lead to numerical instabilities and self intersection problems when applied to some complex segmentation tasks. Implicit active contours models avoid these problems and can handle topological changes automatically [17]. However, they are not used in this thesis due to their higher computational complexity.

A contour does not need to be a single curve. Modifications to delineate ribbon structures, like roads in aerial images [32], or blood vessels in angiographic images [45], have been proposed in the literature.

5.1.2 Constraints, Optimisation and Prior Information

In our new approach we perform the optimisation of Eq. 5.3 by a greedy algorithm. In a bounded neighbourhood around the current parameter value all points on a lattice are

generated, the energy is evaluated, and the point of lowest energy is selected as the new parameter set. This optimisation scheme has advantages over other optimisation algorithms, namely the direct inclusion of hard constraints, simplicity and speed.

Soft constraints (e.g. desired object dimensions) are imposed by further energy terms in Eq. 5.3. These soft constraints reshape the objective function so that the minimum moves towards the desired position in parameter space. This is very similar to the technique of Lagrange multipliers, which is extensively used in the classic optimisation literature.

Hard constraints are enforced by limiting the search neighbourhood such that the constraints are held. Prior or model information can be integrated in the same way.

5.1.3 Extension to Multicocular Cameras

Assuming that a segmentation boundary is observable in multiple images there exist three ways to use it: Fusing multiple independently obtained 2D snakes, reconstructing the spatial position of the segmentation boundary and fitting the 3D snake to it, or projecting the 3D snake into the images and computing E_{int} as the sum of the individual energies of the images in the tuple.

From the fusion of multiple independently obtained 2D snakes arises the correspondence problem mentioned in Sec. 1.2, e.g. if parts of the segmentation boundary are not detected in one image. This requires specific efforts to establish the correspondences which in turn may be used to obtain the spatial coordinates in the known way.

The reconstruction of spatial coordinates of the segmentation boundary and the successive fitting of the 3D snake to it bears the usual problems of stereo vision. The methods to follow this approach have been published elsewhere and are not subject of this thesis anyway.

In this new approach we concentrate on the projection of the 3D snake to the image and their evaluation in the images of a tuple. To do so we extend Eq. 5.2 to multiple images in the same way as in template matching:

$$F = \int_0^l \left[E_{\text{int}}(\vec{v}(s)) + \sum_{c=0}^{N_c} E_{\text{ext}} \left(\vec{P}_c ({}^c_0 H \vec{v}(s)) \right) \right] ds \quad (5.4)$$

The internal energy terms are similar to those of the 2D snakes, just in 3D and measured

in meter, not pixel. The external energy terms are identical to the 2D case as they operate on the projected spatial curve.

5.2 The Contracting Curve Density Algorithm

The Contracting Curve Density (CCD) algorithm [37, 38, 39], and its real-time variant [40], fit a parametric curve $\vec{p}(s, \vec{\Phi})$ to an image \vec{I} . Parameter $s \in [0, 1]$ increases monotonically along the curve, and $\vec{\Phi}$ denotes the curve parameters to be optimised. Instead of finding the most likely value of $\vec{\Phi}$ only, the CCD algorithm also obtains a measure of uncertainty. The density of $\vec{\Phi}$ is assumed to be Gaussian and will be denoted by its mean vector $\vec{\mu}_{\vec{\Phi}}$ and covariance matrix $\vec{\Sigma}_{\vec{\Phi}}$.

Inputs to the CCD algorithm are the image \vec{I} and the Gaussian approximation of the prior parameter density $(\vec{\mu}_{\vec{\Phi}}^0, \vec{\Sigma}_{\vec{\Phi}}^0)$. Outputs are the estimated density parameters $\vec{\mu}_{\vec{\Phi}}^{\tilde{}}$ and $\vec{\Sigma}_{\vec{\Phi}}^{\tilde{}}$.

Furthermore a parametric curve $\vec{p}(s, \vec{\Phi})$ is required that models the position of the relevant object features in the image. Requirements for \vec{p} are:

- The curve is smooth, continuous and differentiable twice in $\vec{\Phi}$.
- The curve does not have to be smooth, continuous and differentiable in s .
- The normal of the curve must exist.
- The curve’s normal is smooth, continuous and differentiable twice in $\vec{\Phi}$.

It is clear that the set of objects that can be matched by the CCD algorithm is more limited than that of e.g. template matching. The geometry of these objects should be simple enough to allow an efficient curve model. They are, however, not limited to 2D objects. Spheres and cylinders have been used in [37, 38, 39, 40]. The curves even include the lens distortion of the camera.

In the following we introduce the algorithm and present two different extensions to multiple cameras, which have been designed, implemented, tested, and evaluated for this thesis.

5.2.1 Monocular Matching Procedure

In this section we introduce the basics of the CCD algorithm, condensed such that subsequent extension and applications become clear, but without the elaboration of [38]. We repeat here the fundamental operations of the Real-Time CCD algorithm [40] since it is better understandable and more suited to our applications than the original variant.

The first basic idea of the CCD algorithm is that at the beginning of the matching the uncertainty of the curve parameters is high. During the matching, this uncertainty reduces up to a point where the curve is known with sub-pixel accuracy. The second basic idea is that a matching algorithm should aim to find the most probable value of $\vec{\Phi}$ given an image. The CCD algorithm therefore is a maximum-a-posteriori probability (MAP) algorithm.

The a-posteriori probability is approximated as follows using Bayes' equation:

$$p(\vec{\Phi}|\vec{I}) = p(\vec{I}|\vec{\Phi})p(\vec{\Phi}|\vec{\mu}_{\vec{\Phi}}^0, \vec{\Sigma}_{\vec{\Phi}}^0) \quad (5.5)$$

$$p(\vec{\Phi}|\vec{I}) = p(\vec{I}|S(\vec{\mu}_{\vec{\Phi}}, \vec{\Sigma}_{\vec{\Phi}})) \cdot p(\vec{\mu}_{\vec{\Phi}}|\vec{\mu}_{\vec{\Phi}}^0, \vec{\Sigma}_{\vec{\Phi}}^0) \quad (5.6)$$

$S(\vec{\mu}_{\vec{\Phi}}, \vec{\Sigma}_{\vec{\Phi}})$ denotes the grey value statistics close to the curve and is used to approximate $p(\vec{I}|\vec{\Phi})$ by $p(\vec{I}|S(\mu_{\vec{\Phi}}, \Sigma_{\vec{\Phi}}))$.

The maximisation of Eq. 5.6 is performed by iterating the following two steps until the changes of $\vec{\mu}_{\vec{\Phi}}$ and $\vec{\Sigma}_{\vec{\Phi}}$ fall below a threshold or a fixed number of iterations is completed. The data in Fig. 5.1 illustrates the steps. The procedure starts with a user-supplied starting density $(\vec{\mu}_{\vec{\Phi}}^0, \vec{\Sigma}_{\vec{\Phi}}^0)$.

1. Compute pixel value statistics $S(\vec{\mu}_{\vec{\Phi}}, \vec{\Sigma}_{\vec{\Phi}})$, close to both sides of the curve. For grey scale image this amounts to a mean and a standard deviation of the image content on either side of the curve.
2. Refine the curve density parameters $(\vec{\mu}_{\vec{\Phi}}, \vec{\Sigma}_{\vec{\Phi}})$ towards the maximum of Eq. 5.6 by one step of the Newton-Raphson optimisation procedure. This step moves the segmentation boundary such that the image content conforms better to the pixel statistics, i.e. towards an edge.

A numerically favourable form of Eq. 5.6 is obtained by computing the log-likelihood:

$$\chi = -2 \ln \left[p(\vec{I}|S(\vec{\mu}_{\vec{\Phi}}, \vec{\Sigma}_{\vec{\Phi}})) \cdot p(\vec{\mu}_{\vec{\Phi}}|\vec{\mu}_{\vec{\Phi}}^0, \vec{\Sigma}_{\vec{\Phi}}^0) \right] \quad (5.7)$$

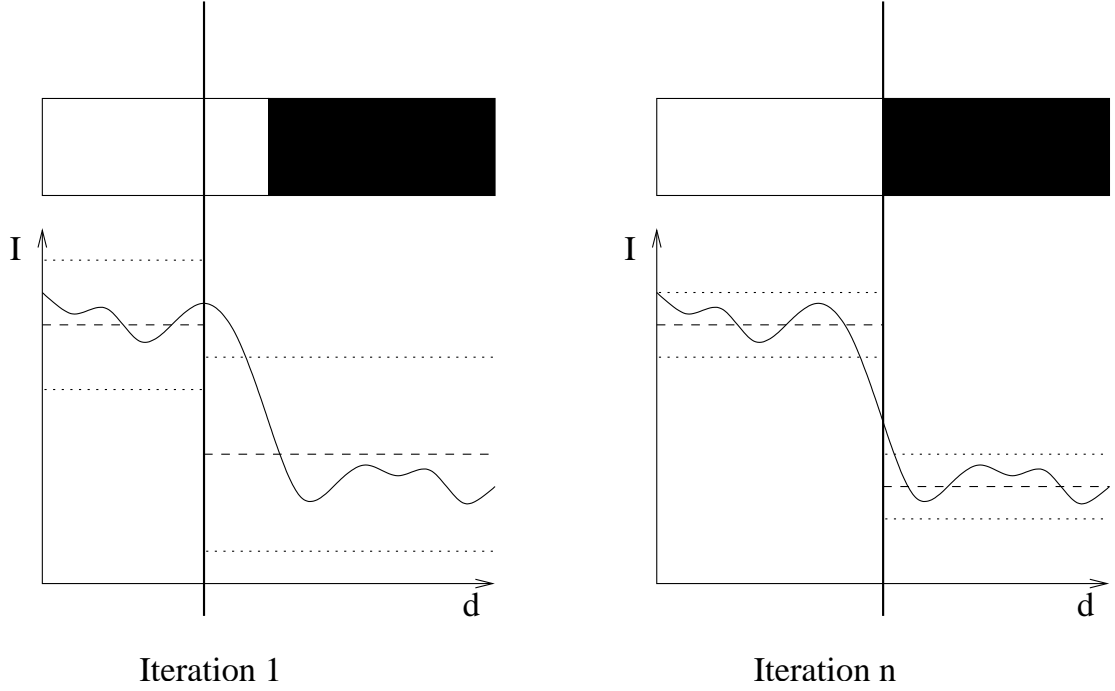


Figure 5.1: The principle of the CCD algorithm. Fitting the segmentation boundary (bold line) to the image content (solid line) by estimating the mean (dashed line) and standard deviation (dotted line) on either side of the assumed boundary. The boundary is moved such that the image content has the highest probability according to the means and standard deviations.

In image processing/optimisation terms this can be seen as follows: The sum of Gaussian probability densities $p\left(\vec{I}|S\left(\vec{\mu}_{\vec{\Phi}}, \vec{\Sigma}_{\vec{\Phi}}\right)\right)$ is an edge detector along the curve normal, i.e. if the curve is at the edge, the function value is maximal. In contrast to classical edge detectors (e.g. Sobel, Prewitt) the kernel size is adaptive and the function is spatially differentiable. These properties are the main reasons for the robustness and accuracy of the CCD algorithm. At the beginning of the matching $\vec{\Sigma}_{\vec{\Phi}}$ is quite large, providing a wide and smooth function. During the estimation $\vec{\Sigma}_{\vec{\Phi}}$ is gradually reduced to approximate the actual uncertainty of the parameters and therefore reducing the smoothing effect. The Gaussian probability density $p\left(\vec{\mu}_{\vec{\Phi}}|\vec{\mu}_{\vec{\Phi}}^0, \vec{\Sigma}_{\vec{\Phi}}^0\right)$ serves as a regularisation since the edge detection results are not a smooth function.

In order to improve the performance not all pixels in the vicinity of the curve are used. For efficiency reasons a set $\{s\} \subset [0, 1]$ with a finite number of members is used. Only at the points $\{\vec{p}_s\} = \left\{\vec{p}\left(s, \vec{\Phi}\right)\right\}$ data for the objective function (Eq. 5.7) is extracted. The curve normal \vec{n}_s is supposed to be perpendicular to the curve at \vec{p}_s .

Along \vec{n}_s the weighted mean and standard deviation of the pixel values on either side are obtained separately, forming $S(\vec{\mu}_{\Phi}, \vec{\Sigma}_{\Phi})$.

Please note that our curve definition $\vec{p}_s = \vec{p}(s, t, \Phi)$ is slightly different to [37, 38, 39, 40] as we have to handle multiple curves (e.g. the left and the right edge of a tube). Each curve is identified by a different value of $t \in \mathbb{Z}$, e.g. -1 for left and $+1$ for right edge.

Details — including the complete set derivatives, which are missing in Hanek’s publications — can be found in [57].

5.2.2 Multiocular Extension of the CCD Algorithm

The simplest way to extend the CCD algorithm to multiple calibrated cameras is to maximise the joint probability

$$p(\vec{\Phi}|\vec{I}) = \prod_{c=1}^{N_c} p(\vec{I}_c|\vec{\Phi}) \cdot p(\vec{\Phi}|\vec{\mu}_{\Phi}^0, \vec{\Sigma}_{\Phi}^0) . \quad (5.8)$$

The assumption here is that images are independent random variables. This is an approximation of the true relationship between the images: On pixel level those images are independent due to independence of the pixel noise, on an object level they are not. Moving the object in space causes a change of the contents of one or more images. The actual relationship is hard to compute and is therefore approximated in the aforementioned way.

This extension provides implicit triangulation capabilities as the other methods before. We will denote this version as MOCCD (Multiocular CCD algorithm).

The major drawback of this approach is the non-linearity of the model due to the projection. Depending on the curve model, Eq. 5.7 even may be no longer a convex function. In this case, the Newton-Raphson step may require further measures — such as step size control — in order to converge correctly. In any case, the Hessian of Eq. 5.7 will be a dense matrix, leading to an $O(n^3)$ complexity of the Newton-Raphson step for n curve parameters.

However, if the model is linear, the Hessian of Eq. 5.7 degenerates to a diagonal matrix. This leads to an $O(n)$ complexity of the Newton-Raphson step. Suitable curve models define 2D entities and depend on translation, scaling and shearing only. Although not linear, planar rotation can be used too.

In order to express the original model in 3D coordinates and match it in 2D we project the curve to the images, match them independently in 2D, and perform a 2D-3D pose estimation. The multiocular CCD algorithm with 2D matching (MOCCD2D) is summarised in the following steps:

1. Project 3D model to 2D representations: ${}^c\mu = {}^c\vec{p} \left({}^c_0H, \vec{\Phi}^k \right)$
2. Match 2D representations by planar operations: ${}^c\tilde{\mu} = \text{CCD1} ({}^c\mu)$
3. Find 3D model with LMS projection error: $\vec{\Phi}^{k+1} = \underset{\vec{\Phi}}{\text{argmin}} \left({}^c\tilde{\mu} - {}^c\vec{p} \left({}^c_0H, \vec{\Phi} \right) \right)^2$

The function CCD1 denotes a single Newton-Raphson step. We operate these three steps repeatedly until $\vec{\Phi}$ doesn't change any more or a fixed number of iterations is performed. Clearly, this approach is independent of both the 3D and the 2D representation of the object model as long as the 3D-2D projections exist.

6 Closest Point Methods

6.1 Object Recognition using Characteristic Local Features

In this section we describe a new multiocular object recognition method that performs a greedy search for the best matching point of a projected feature. In [54] we have shown the properties of this approach for localising airplanes on the taxiway. In this thesis we will show how to extend the method to multiple calibrated cameras.

The method improves a set of pose parameters given an image and an object model by rendering a 2D representation of pose and model. This representation does not have to be photo-realistic or an image at all, it is to be seen as the viewpoint dependent parameters of a 2D feature extractor. After projection the optimal position of the feature will be determined in the vicinity of the projected position by means of a greedy search. After all features have found their optimal positions, a 2D-3D pose estimation [2, 41, 63] using the obtained optimal 2D positions and the modelled 3D positions is performed. These steps are repeated until convergence is reached.

6.1.1 Characteristic Local Features

The central concept for this method is the Characteristic Local Feature (CLF). We assume:

- that the pose to be optimised is close to the pose depicted in the input data
- the input data exhibits salient points
- these points can be modelled in 3D
- the appearance of these points in the image can be computed from the pose and further data.

If these assumptions are upheld, the projected CLFs are similar to the image. One can verify this by a slightly changing the pose of a cube and overlaying the camera images.

The corner positions vary relative to each other, the image content close to the corners, however, does not change much.

Details on how to generate CLFs from CAD models can be found in [26, 74].

6.1.2 Monocular Matching Procedure

A model consists of several CLFs. Each CLF stores the following elements in a model and accesses them read-only during matching:

- A set of rules deciding the applicability of the CLF, given the pose and possibly other data, e.g. temperature if matching far-infrared (thermal) images.
- At least one 3D position in the model coordinate system.
- A projection function to generate the 2D representation.
- A matching function that compares the 2D representation and the input image.
- Further parameters for the projection function and the matching function, e.g. reference values.

Furthermore, a CLF stores the following elements in the object hypothesis (together with pose and model reference) for read-write access during matching:

- At least one 2D position (the image coordinates to be optimised).
- A reference to the read-only part of the CLF.
- Further viewpoint dependent parameters for the matching function, e.g. expected textures.

Algorithm 6.1 details the matching procedure.

The separation of the CLFs and the 2D/3D points is necessary as one CLF may be influenced by more than one point, e.g. a line is one CLF, but requires two points if rotation and scaling is to be covered.

The details of the procedures used in Algorithm 6.1 are left out for brevity, their semantics are listed in Table 6.1.

Algorithm 6.1 Monocular Matching using Characteristic Local Features

Variable	Type	Description
I	Input	Image to be matched
N_f	Input	Number of CLFs
N_c	Input	Number of cameras
$\vec{\Phi}^0$	Input	Start pose
$M = \{C_i\}$	Input	Model as a set of CLFs
N_m	Parameter	Number of matching iterations
$\theta_{\vec{\Phi}}$	Parameter	Pose similarity criterion
Δ_p	Parameter	2D search window around projected position
$\vec{\Phi}$	Output	Optimised pose
c_i	State	Projected data of the CLF
P	State	Set of model point indices to project to 2D
\vec{p}_i	State	2D point coordinates
\vec{P}_i	State	3D point coordinates
A	State	Set of CLF indices to use during the matching

```

1:  $m \leftarrow 0$ 
2: repeat
3:    $P \leftarrow \{\}$ 
4:    $A \leftarrow \{\}$ 
5:   for all  $i \in [1, N_f]$  do
6:     if applicable( $C_i, \vec{\Phi}^m, \dots$ ) then
7:        $c_i \leftarrow \text{projectData}(C_i)$ 
8:        $P \leftarrow P \cup \text{points}(C_i)$ 
9:        $A \leftarrow A \cup \{i\}$ 
10:    end if
11:  end for
12:  for all  $i \in P$  do
13:     $p_i \leftarrow \text{projectPoint}(P_i)$ 
14:  end for
15:  for all  $j \in P$  do
16:     $p_j \leftarrow \underset{\delta_p \in \Delta_p}{\text{argmin}} \sum_{i \in A} \text{similarity}(I, C_i, c_i, p_0, \dots, p_j + \delta_p, \dots)$ 
17:  end for
18:   $\vec{\Phi}^{m+1} \leftarrow \text{poseEstimation}(P, M)$ 
19:   $m \leftarrow m + 1$ 
20: until  $\|\vec{\Phi}^m - \vec{\Phi}^{m-1}\| < \theta_{\vec{\Phi}} \wedge m > N_m$ 
21:  $\vec{\Phi} \leftarrow \vec{\Phi}^m$ 

```

Table 6.1: Semantics of the procedures used in Algorithm 6.1.

Procedure	Description
$b = \text{applicable}(C_i, \vec{\Phi}, \dots)$	Check (boolean), whether CLF C_i is applicable for the current pose $\vec{\Phi}$. Further information may influence this decision.
$c_i = \text{projectData}(C_i)$	Compute the 2D representation c_i of the CLF C_i .
$P = \text{points}(C_i)$	Obtain the indices of the model points used.
$p_i = \text{projectPoint}(P_i)$	Project 3D point P_i into the image as 2D point p_i .
$s = \text{similarity}(I, C_i, c_i, p_0, \dots, p_j, \dots)$	Compute the similarity $s \in [0, 1]$ of a projected CLF C_i, c_i to the image I , if the 2D positions of the CLFs were at p_0, \dots, p_j, \dots

6.1.3 Multiocular Matching Procedure

The extension to multiple calibrated cameras is performed by independently projecting each CLF to all cameras, optimising the 2D positions, and computing the LMS pose of the object.

This extension influences three actions in Algorithm 6.1: The applicability check, projection and the pose estimation. The extension of the applicability check and the projection are trivial. They have to take into account the different interior and exterior camera parameters, but work as if there was just a single camera.

The only part that operates on all cameras is the pose estimation. Extending it to multiple cameras is straightforward, as [2, 63] are Bundle Adjustment methods, which were described in Sec. 3.2.

Extending [41] to multiple cameras has not been attempted directly on point data, but [75] show a similar approach, just that 3D surfaces are matched, not 3D points. Due to its simplicity and stability we use a Bundle Adjustment based method [2].

6.2 Object Recognition using Gradient Sign Tables

The object recognition methods presented before introduced new variants of established recognition schemes which improve upon the state of the art in various ways. They exhibit,

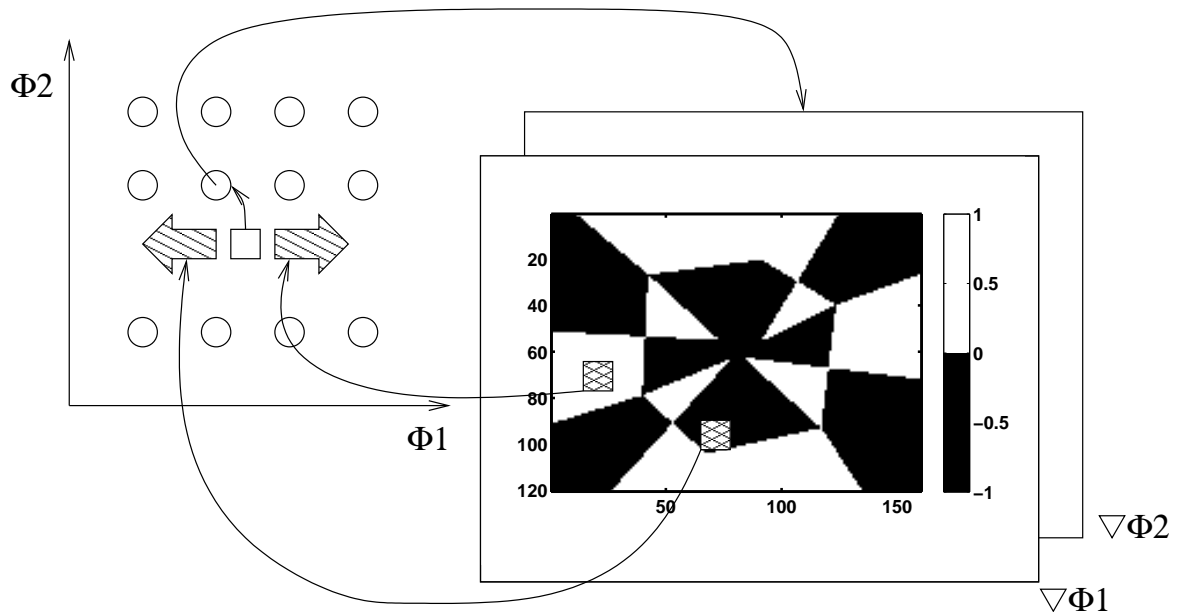


Figure 6.1: Gradient Sign Table matching: Principle of operation. For an initial pose $\vec{\Phi}$ (square in pose space. left diagram), the nearest quantised pose $[\vec{\Phi}]$ (circle) is computed. The quantised pose is attributed with one Gradient Sign Table (GST, right diagram) per degree of freedom. For all detected features (hatched squares) the respective sign is obtained and the scaled average value of these signs is used as the update for the pose.

however, a rather conventional approach to object recognition. In this section we will attack the problem of pose estimation and object classification from a completely different angle.

The objective here is to design an object recognition method which is as simple as possible, both in regard to the algorithm and the data structures. It would be advantageous if the new method operates using simple, quickly obtainable features such as edges only, but is not limited to them.

We will start with an overview of the matching procedure in order to illustrate the underlying idea followed by the mathematical background. Refer to Fig. 6.1 for a sketch of the operation of the matching procedure.

Gradient Sign Table (GST) matching operates on localised features (e.g. grey level edges, colour edges, corners) that are extracted from the input image. The matching procedure is a pose refinement hence an initial pose is necessary. The following steps are repeated until convergence or for a fixed number of iterations. Given the pose the most appropriate GST is selected from the database, e.g. by finding the GST that was generated from a pose close to the pose to be refined. The contents of the GST indicate that if a feature is

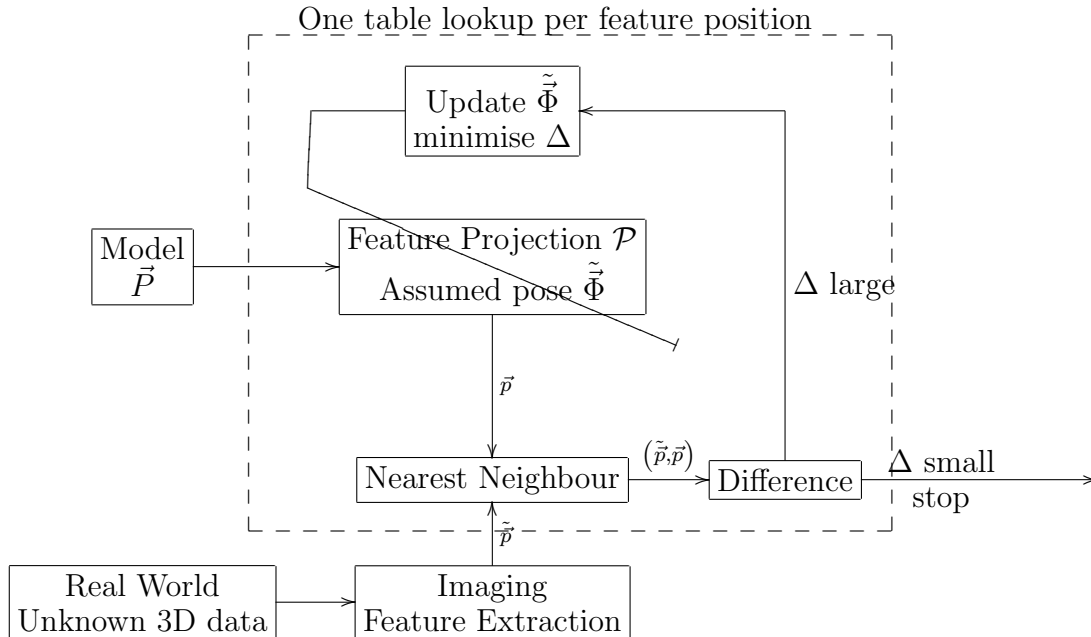


Figure 6.2: Gradient Sign Table matching: Principle of operation. Similar to Bundle Adjustment (Fig. 3.5), but without explicit correspondences and with precomputed gradients.

found at a certain pixel position in the input image whether an increase or a decrease of the value of current pose would improve the correspondence between image and model.

This scheme allows us to get rid of the most expensive image processing operation that e.g. template matching requires: distance image computation. A single distance image is not computationally expensive, but since each edge direction channel requires one, the amount of time a program spends in this routine is about 50% of the total running time. In industrial applications this is not crucial, but some applications (e.g. controlling a robot using computer vision) require high frame rates.

We get rid of the distance transform of each input image by distance transforming the model. This idea is not really new [61], but our simplifications of the matching function are. In fact, the underlying idea could open the door to a new set of optimisation methods that use a precomputed view of the objective function, stored in an efficient manner.

The underlying idea of our new approach (Fig. 6.2) is to minimise the difference between image and model using a gradient descent optimisation with precomputed gradient values that are reduced to their sign only. This is the simplest pose refinement algorithm as it requires one table look-up per feature position only.

6.2.1 Gradient Sign Tables

Recall Eq. 3.2, page 30, the optimisation procedure of Bundle Adjustment (using Euclidean distance) repeated in Eq. 6.1:

$$\begin{aligned}\tilde{\Phi} &= \underset{\vec{\Phi}}{\operatorname{argmin}} \sum_{i=1}^{N_f} F_i \\ F_i &= \left(\vec{p}_i - \tilde{\vec{p}}_i \right)^2 \\ \vec{p}_i &= \mathcal{P} \left(\Phi, \vec{P}_i \right)\end{aligned}\tag{6.1}$$

We will assume $\tilde{\vec{p}}_i$ to be one of the N_f observed image features and $\vec{\Phi}$ the pose of the model. The model points are denoted \vec{P}_i . Eq. 6.1 therefore performs pose estimation of a given object. Pose estimation is usually performed using explicit one-to-one correspondences between $\tilde{\vec{p}}_i$ and \vec{P}_i as we did in Sec. 3.2.

In Sec. 6.1 we searched for the closest best matching feature of a projected image point in order to establish the correspondences for Araujo’s 2D-3D pose estimation. The latter is a sub-problem of bundle-adjustment similar to Eq. 6.1. Fitzgibbon’s LMICP method [30] later used a similar approach using edges alone. This shows that using the nearest feature during pose estimation yield sufficiently robust and accurate results. We will therefore retain this method of correspondence generation in this novel method.

Classically the minimisation of Eq. 6.1 (Fig. 3.5) is performed by the Gauss-Newton method starting at a given pose $\vec{\Phi}_0$. If we change the optimisation procedure to gradient descent the update rule becomes:

$$\begin{aligned}\vec{\Phi}^{(k+1)} &= \vec{\Phi}^{(k)} - \sum_i^{N_f} \nabla F_i \\ \nabla F_i &= \alpha_d \left. \frac{\partial F_i}{\partial \Phi_d} \right|_{\vec{\Phi}^{(k)}}, \quad d \in [1, N_d]\end{aligned}\tag{6.2}$$

with ∇F_i being the scaled gradient of F_i and α the step size.

The heart of the novel pose estimation presented here is how to compute the partial derivatives $\frac{\partial F_i}{\partial \Phi_d}$ efficiently. This involves two steps: finding the model point closest to $\tilde{\vec{p}}$ and computing the derivative. We will replace these steps by a single table lookup as depicted in Fig. 6.2.

For the moment we ignore that the probability of a point to belong to the object decreases with increasing distance to projected pose. This should be respected by different handling of points with different distances to the curve. We will integrate this individual handling into the compression as described below.

Let's assume the pose space is quantised like in Sec. 4.1 and for a pose $\vec{\Phi}$ the quantised pose is $\lfloor \vec{\Phi} \rfloor$. In Fig. 6.1 this is indicated by the arrow from the square depicting $\vec{\Phi}$ to the circle depicting $\lfloor \vec{\Phi} \rfloor$. The gradient of F_i then becomes

$$\nabla F_i = \alpha_d \left. \frac{\partial F_i}{\partial \Phi_d} \right|_{\lfloor \vec{\Phi} \rfloor^{(k)}}. \quad (6.3)$$

It is clear that for constant α_d the gradient ∇F_i becomes constant for a given feature position \tilde{p}_i . Assuming pixel accurate feature positions, we could store the images of partial derivatives instead of the template (cf. Algorithm 4.1).

This would lead to an image-sized matrix per degree of freedom and pose, requiring much more memory than the template. The matching procedure, however, would be reduced to a simple table look-up for every observed feature pixel.

In order to reduce the memory requirements of these gradient tables the proper choice of α_d is crucial. Since the poses are quantised to steps of σ_d , it would be efficient to update the current pose $\vec{\Phi}^{(k)}$ by a value of σ_d such that the quantised pose $\lfloor \vec{\Phi} \rfloor^{(k)}$ switches to the next grid location. If the update is less than σ_d , we unnecessarily repeat the same calculation all over again in the next iteration.

The consequence is that we set $\alpha_d = \frac{\sigma_d}{N_f \left| \left. \frac{\partial F_i}{\partial \Phi_d} \right|_{\lfloor \vec{\Phi} \rfloor^{(k)}} \right|}$. This gives a gradient of

$$\nabla F_i = \frac{\sigma_d}{N_f} \operatorname{sgn} \left. \frac{\partial F_i}{\partial \Phi_d} \right|_{\lfloor \vec{\Phi} \rfloor^{(k)}}. \quad (6.4)$$

We therefore have to store only a single bit per pixel, degree of freedom, and pose: The sign of the gradient for this pixel and degree of freedom. The resulting image-sized binary matrices are computed off-line and will be called *Gradient Sign Tables*, or *GST*.

6.2.2 Compressing Gradient Sign Tables

In order to reduce the memory consumption further, a compression scheme is necessary. Looking at a typical GST in Fig. 6.3 we see that the matrix is populated in a regular

manner. There are large areas of either only ones or only zeros, while other areas consist of non-random zero-one pattern. This is an indication that the matrices can be compressed further.

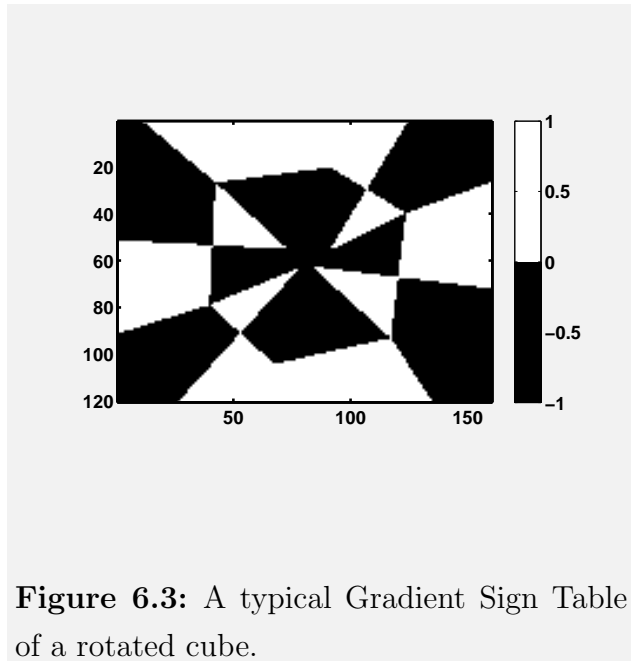


Figure 6.3: A typical Gradient Sign Table of a rotated cube.

There is a wide choice of lossless compression algorithms (e.g. [14, 47]), which can also be combined with spatial access methods (e.g. [29, 33]). For simplicity and high access speed we chose run-length encoding (RLE, Def. A.10). Additionally we computed the bounding box of the non-zero area and stored only its contents.

The interesting point here is that we can also use the template data and transform them into GST. This is accomplished by comparing the distance of pixels at identical image coordinates for different poses.

If the e.g. left neighbouring pose $\vec{\Phi} - 1$ of the current pose $\vec{\Phi}$ has a smaller distance at a certain point, we store a value of -1 in the GST of $\vec{\Phi}$.

At this point we will integrate the fact that only pixels near the projected features can be assumed to be object pixels. This assumption holds if the starting pose is sufficiently close to the real pose of the object. As explained in Sec. 4.1.5 we should not try to compensate a pose error that is larger than half of the object size to not to violate the Sampling Theorem. Otherwise — e.g. in case of non-oriented edges for features as done in this thesis — the optimisation will converge to a local minimum by e.g. aligning the left side of the model with the right side of the object’s image. This happens if the start pose is chosen too far to the right of the actual pose.

In order to define which pixels are to be processed we define a fixed-width corridor around the projected model curve. Pixels inside the corridor are assumed to be object pixels that are important for pose estimation, pixels outside the corridor are assumed to be either caused by background objects or objects that are found from other starting positions. The width of the corridor can be computed if a probability distribution of the pose is known. The union of all feature positions generated by all poses above a pre-defined probability

— with the currently generated pose as the Expectation Value — form the corridor.

If the distribution is assumed to be Gaussian and the object features form a curve with known normal at each pixel, the standard deviation of the pose can be projected onto the normal using a linearisation of the projection function at the current pose. A pre-defined probability can then be turned into a maximal distance using the corresponding quantile of the Gaussian distribution. This method is employed in the CCD algorithm [38] and can be easily adapted.

Algorithm 6.2 summarises the generation of compressed GSTs.

Algorithm 6.2 Generation of Gradient Sign Tables

Variable	Type	Description
N_d	Input	Number of lattice dimensions
$\underline{\Lambda}_d, \bar{\Lambda}_d$	Input	Limits of lattice dimension d
σ_d	Input	Step size of lattice dimension d
$G_{\vec{l},d}$	State	Gradient sign table for a pose
\vec{l}	State	Index of current pose
$C_{\vec{l},d}$	Output	Compressed gradient sign table for a pose

```

1: for  $l_d \in [\underline{\Lambda}_d : \sigma_d : \bar{\Lambda}_d] \Big|_{d=1}^{N_d}$  do
2:   for  $d \in [1, N_d]$  do
3:     Compute distance transformed template  $D_d^+$  at pose  $[\Phi_1 \dots \Phi_d + \sigma_d \dots \Phi_{N_d}]$ 
4:     Compute distance transformed template  $D_d^-$  at pose  $[\Phi_1 \dots \Phi_d - \sigma_d \dots \Phi_{N_d}]$ 
5:     Compute distance transformed template  $D_d$  at pose  $\vec{\Phi}$ 
6:     for each image pixel  $\tilde{p}_i$  with  $D_d(\tilde{p}_i) < d_{\max}$  do
7:       Update Bounding Box
8:        $g = 0$ 
9:       if  $D_d^-(\tilde{p}_i) < D_d(\tilde{p}_i)$  then
10:         $g = -1$ 
11:       else
12:        if  $D_d^+(\tilde{p}_i) < D_d(\tilde{p}_i)$  then
13:          $g = +1$ 
14:        end if
15:       end if
16:       Store  $g$  in RLE compressed GST  $C_{\vec{l},d}$ 
17:     end for
18:   end for
19: end for
    
```

One simple way to ignore these pixels outside corridor is to assign them a gradient sign of zero. This causes a trinary base $(-1, 0, +1)$, which is not convenient on a binary computer. We therefore require 3 symbols which are stored in 2 bits. This leaves 6 bits in a byte for the length per RLE strip.

6.2.3 Matching using Gradient Sign Tables

Eq. 6.2 and 6.4 form a single iteration of the matching procedure. In practice we want to step from lattice point to lattice point without rounding. Thus we make a majority decision for each degree of freedom individually whether to change positive, negative, or not at all.

The matching procedure therefore counts the features in the corridor and if a suitable number is found it checks whether the majority of these features were at positions labelled with $-1, 0$, or $+1$. The labels are read from the compressed GST stored at the current lattice position. If no DoF collected a sufficient number of votes to justify a move to a different lattice point, the iteration is stopped. Some precautions are taken as this algorithm tends to oscillate between two lattice points at the end of the optimisation.

In order to evaluate the quality of a local minimum one classically computes [76] the Hessian matrix (e.g. using numerical derivatives of the gradient from Eq. 6.4). If the condition number (ratio of smallest to largest eigenvalue) of the Hessian is close to 1, the dimension-wise curvatures of parameters of the objective function are uniform.

Therefore, this local minimum can be reached from any direction with approximately the same number of iterations and is therefore a good indication for a valid object. Smaller numbers indicate certain directions in pose space are less curved and therefore more prone to noise-related local minima. We define a “good” match to be a pose that can be reached from any direction in a stable manner and has to have therefore a high condition number.

This approach seems impractical. Therefore we decided to use a simpler method. We used the position of a pose in the optimisation sequence as a quality measure. Thus the pose the iteration stopped was given a quality of 1, the pose the iteration started a 0. Local maxima in the average sequence position were detected and the values were used to compute weighted average pose over a fixed neighbourhood if a suitably large quality value was found.

If a more classical measure of quality — the average distance from observed features to

the closest model feature — is required the gradient sign tables for $\frac{\partial F_i}{\partial(\frac{\vec{z}}{\vec{p}_i - \vec{p}_i})}$ can be used for numerical integration. We will designate these GST as distance-image coordinate GST, in contrast to the distance-pose GST explained before.

From Eq. 6.2 follows that an initial pose is required. Object detection, which cannot use any prior knowledge, is done by performing the pose estimation above from random or equidistantly spaced starting points. Real objects are more likely to attract poses from different positions in the vicinity. We therefore accumulate the quality values of the poses and place detection results at poses that were found from a number of different starting positions.

The GST matching is easily extended to multiple calibrated cameras by computing a GST for each camera separately. The total gradient is then computed by voting the individual gradients from each input image. The complexity therefore grows linear with a growing number of cameras.

We can safely omit the matching algorithm here due to its simplicity.

Part III

Experiments and Applications

In the following we will evaluate the methods from Part II using example applications from industrial environments. This chapter is structured into three sections: camera calibration, rigid objects, and flexible objects (tubes and cables).

Camera calibration — the foundation of the multiocular methods in this thesis — is evaluated without a specific application in mind. It can be and is used for all other applications. It establishes a self consistent reference system, in which all units of measurement are defined by the calibration result.

Using two rigid objects — oil caps — the pose estimation methods Template Matching, Feature Pose Maps, and Gradient Sign Table matching are compared regarding pose estimation accuracy and classification performance. As the same object is used in other publications [79] we can compare the matching results of the methods from this thesis with them. We will not include the Characteristic Local Feature matching in the investigation as we regard the newly developed Gradient Sign Table matching to supersede the older method.

Using a set of different cables and tubes with diameters between 6 mm and 12 mm we solved the “dangling rope problem”: the localisation of the loose end of a cable, tube, or rope given the position of the fixed end. We can obtain the diameter *and* the position of the flexible objects using the multiocular Contracting-Curve-Density Algorithm, the multiocular Active Contours, and the Gradient Sign Tables, but not with any of the monocular methods. We therefore use a fixed diameter to obtain comparable results.

The purpose of this thesis is to compare the different methods, not to e.g. find a feature that is robust under all circumstances. For simplicity we use Sobel edges (except for the Contracting Curve Density algorithm), followed by a non-maximum suppression and thresholding using a fixed threshold. The cameras are used either in auto-exposure mode or with a manual shutter setting.

7 Evaluating the Camera Calibration

7.1 Goal of Investigation

Two different influences on accuracy and repeatability of the calibration result are important: corner position noise and calibration rig placement. The actual accuracy of a calibration is very hard to verify since obtaining a suitable ground truth involves a calibration itself. Even if this calibration is done by a different method, all that is accomplished doing so is the comparison of the two methods. We therefore concentrate on the repeatability of calibration as a measure of accuracy.

The single source of noise influencing the otherwise deterministic calibration process is the noise of the sensor chip, assuming illumination remains unchanged. The correlation used to find corners is offset and gain independent, yet the noise of a camera image increases with decreasing brightness as the exposure control increases the gain between pixel voltage and A/D converter. We therefore obtain the image noise induced noise of the calibration data as a measure of repeatability. The smaller the standard deviation of parameter the more repeatable is it.

The number of degrees of freedom for the rig placement make it impossible to try out all combinations. We therefore try to verify the basic rules established in Sec. 3.2.4. To do so we build image sets of bad positions and include more and more good combinations until we reach the recommended combination of rig poses.

Additionally we are interested in the different external calibration results for the bundle adjustment and for the spatial optimisation.

7.2 Experimental Setup

In order to obtain sufficient statistics, we acquire one hundred images per rig position. The rig is fixed such that it can't move during image acquisition. Even though the rig is exposed to sun light it is a valid assumption that the lighting does not change too much

during the acquisition period of about twenty seconds. We do not use artificial lighting as this causes interferences between the power grid frequency and imaging frequency of the camera.

The images are referred to as I_{ij} with i denoting the pose and j the number of the image in the sequence. We then obtain a set of calibration parameters Φ_j given a set of i_1, i_2, \dots of rig poses. The standard deviation $\sigma\Phi$ is calculated from these one hundred calibrations and therefore yield a measure of repeatability. We only compute the individual variances not the covariances as the dependencies of the respective degrees of freedom are not particularly interesting.

All images have been labelled according to their pose parameters. The distance, for instance, has been divided into four groups (image filling, close, medium and far). Each group label is intuitively chosen, e.g. distance labels are zero through 3 with growing distance. The noise measurements are depicted as one bar per pose group. The bars indicate the square root of the mean square error (RMSE). All experiments are carried out using a 70 degree FoV VGA Digiclops.

The noise of the translation components is measured in meters. The noise of the rotational components cannot be expressed in a unit of measurement directly. Throughout the experiments the rotation is computed as a Rodrigues vector. Since the length of the vector gives the angle of rotation, the noise of this length essentially denotes the noise in the angle of rotation. This implies that the axis of rotation remains constant, which is of course not the case. However, the axis remains largely constant, which allows the use of the length noise as an approximation of the noise of the angle of rotation.

7.3 Pose Repeatability

In this investigation the size of the maximal noise event (largest Euklidean distance between individual pose/corner position and mean pose/corner position) is depicted by a black cross. Please note that three times the RMSE accounts for 95 % probability if the respective distribution is Gaussian, and four times account for 99 % percent probability. Both limits are indicated by black plus signs.

Table 7.1 through 7.5 summarise the noise values obtained from the experiments. The rig poses have been obtained as if the images had been obtained from individual cameras, i.e. monocular pose estimation was performed.

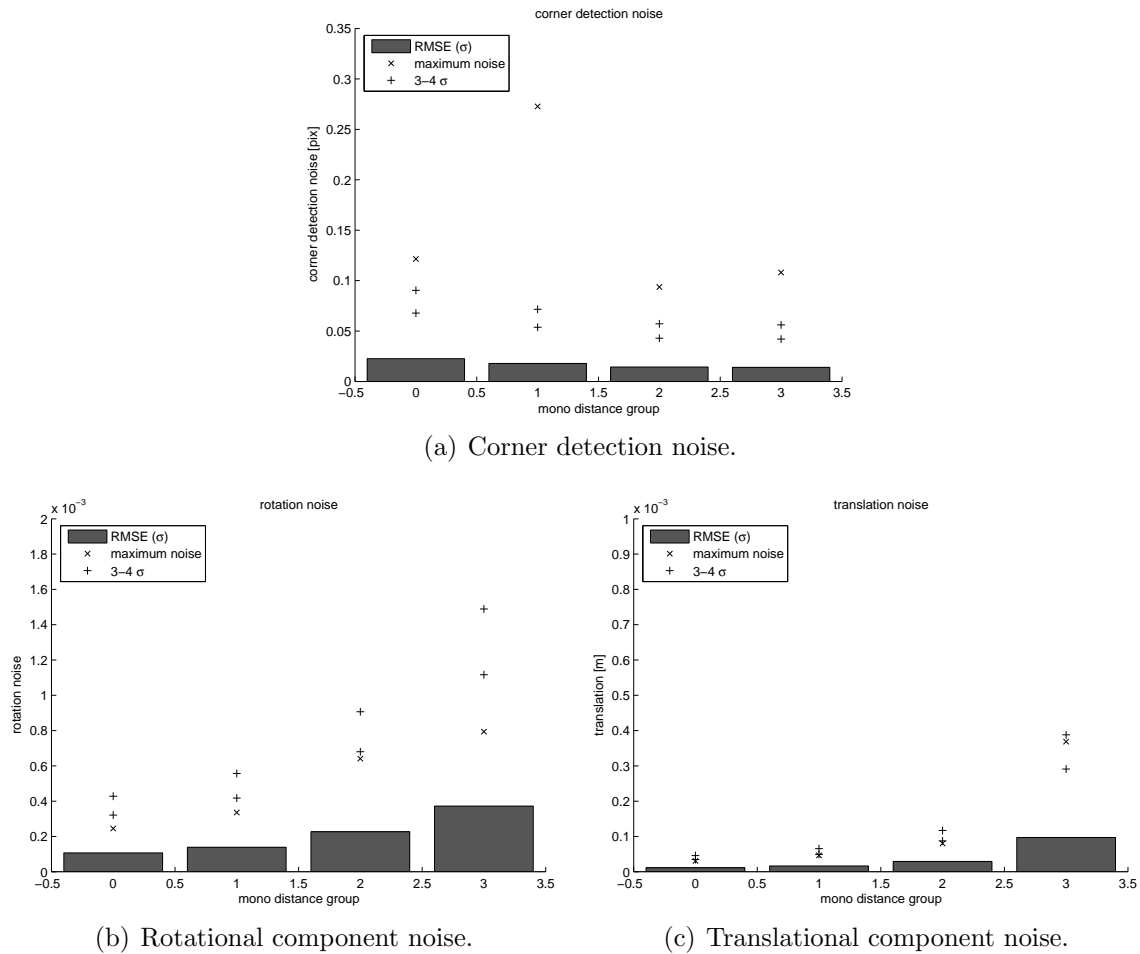


Figure 7.1: Noise of the 70 degree VGA digiclops camera. Rig poses are obtained from individual images (3 x monocular pose estimation). Group codes are: 0=20 cm, 1=40 cm, 2=80 cm, 3=160 cm

The first experiment illustrates the influence of the corner detection noise on the positional noise. Fig. 7.1 depicts both the corner noise and the noise of the pose parameters rotation and translation over the various distance groups. It is clearly visible that both the rotation and translation noise increases with growing distance to the camera. This is to be expected as the distance in an image of the rig is inversely proportional to the spatial distance.

The performance of the parabolic approximation of the corner correlation peak (in a 5×5 pixel wide window) yields an maximum noise of 0.3 pixels. Assuming a Gaussian distribution of the noise 95 % of all pixels are closer than 0.05 pixels to the mean position given the RMSE of 0.018 pixels. The largest maximal noise events happen in images that con-

tain saturated areas close to a corner. We therefore recommend to set the lighting during calibration such that overexposure of the cameras is avoided. Influences of the rig position in the image, orientation or distance on the corner noise cannot be observed.

From the values of the maximal noise events in the corner detection, rotation, and translation plots can be seen that single corner noise events do not always lead to similar events in the estimated poses. In Fig. 7.1 can be seen that the maximal corner noise event is always larger than 4σ , although in both the rotational and translational plots the maximal noise events are smaller than 4σ , usually even smaller than 3σ .

The rotation noise of about 0.015 degrees (cf. Sec. 7.2 for limitations of such an approximation) is nearly constant in all parameter groups. Except for the distance group none of them shows a clear trend.

The translational noise exhibits excellent behaviour. The largest noise event moves the pose less than one millimetre from the mean pose. The mean translation noise caused by the image noise amounts to a negligible 0.14 millimetres at the largest distance. Two slight trends can be observed: Moving the rig off-centre decreases the translational noise, while tilting the rig first decreases then increases the noise.

The first effect may have three causes: The aforementioned overexposure, the planarity of the rig or the fact that pixels at the edge of the image cover a smaller volume than those at the centre. Since the effect is observable to a similar extent in the multiocular pose estimation experiments, we conclude that planarity problems can be ruled out.

The rotation groups show that the noise increases slightly if the rig is strongly tilted. The reason for this effect, which is visible in the multiocular pose estimation too, is to be found in the fact that the corner positions are functions of the cosine of the rotation. The corner noise (which is affected by the slope of the function) is therefore a function of the sine of the rotation. For near planar rig-camera orientations the orientation is close to 180 degrees yielding an almost linear scaling of the corner noise. A greater rotation increases the noise stronger as it operates on the non-linear part of the arcsine function. So the same amount of corner noise produces a greater rotational noise. This is reflected by the rule of thumb [66] to present the calibration rig between zero and 30 degrees, where the sine function is approximately linear.

Table 7.1: Rig distance related noise of the 70 degree VGA digiclops camera. Rig poses are obtained from individual images (3 x monocular pose estimation). CN: Corner noise. PR: Pose noise, Rotation. PT: Pose noise, Translation. MSE: Mean Square Error. MN: Maximal Noise value. Group codes are 0=20cm, 1=40cm, 2=80cm, 3=160cm

distance	Group Code			
	0	1	2	3
CN MSE [pix]	0.022878	0.017145	0.014784	0.016159
CN MN [pix]	0.121505	0.272766	0.110986	0.190439
PR MSE x 1000	0.090268	0.127054	0.265410	0.580246
PR MN x 1000	0.245372	0.354260	0.842944	2.709669
PT MSE [mm]	0.010135	0.015105	0.033259	0.141063
PT MN [mm]	0.030951	0.053713	0.127694	0.976750

Table 7.2: Horizontal rig position related noise of the 70 degree VGA digiclops camera. Rig poses are obtained from individual images (3 x monocular pose estimation). CN: Corner noise. PR: Pose noise, Rotation. PT: Pose noise, Translation. MSE: Mean Square Error. MN: Maximal Noise value. Group codes are -1=left, 0=centre, 1=right

x-position	Group Code		
	-1	0	1
CN MSE [pix]	0.015359	0.016192	0.018086
CN MN [pix]	0.095578	0.272766	0.114174
PR MSE x 1000	0.309176	0.378643	0.377712
PR MN x 1000	0.797720	2.709669	1.240599
PT MSE [mm]	0.035578	0.092413	0.054311
PT MN [mm]	0.137780	0.976750	0.206290

7.4 Calibration Repeatability

In this section we want to explore the influence of the rig pose selection on the quality of the calibration. As a ground truth is not available we investigate the influence of the corner detection noise on the camera parameter noise. If the camera parameter noise of one set of rig poses is smaller (assuming a constant corner detection noise) than the noise of another set of poses, the first pose is more stable and will provide us with a more accountable calibration. The pose sets have been selected to verify the rules of thumb for calibration rig poses as listed in Sec. 3.2.4.

Please note that calibrations that did not converge were removed from the statistics. Non-converging calibrations invariably yield infinite results in some or all camera parameters.

Table 7.3: Vertical rig position related noise of the 70 degree VGA digiclops camera. Rig poses are obtained from individual images (3 x monocular pose estimation). CN: Corner noise. PR: Pose noise, Rotation. PT: Pose noise, Translation. MSE: Mean Square Error. MN: Maximal Noise value. Larger group values indicate more rotation. Group codes are -1=top, 0=centre, 1=bottom

y-position	Group Code		
	-1	0	1
CN MSE [pix]	0.017034	0.016192	0.016518
CN MN [pix]	0.114174	0.272766	0.100532
PR MSE x 1000	0.334569	0.378643	0.355415
PR MN x 1000	1.240599	2.709669	0.797720
PT MSE [mm]	0.052380	0.092413	0.038364
PT MN [mm]	0.206290	0.976750	0.127084

Table 7.4: Rig rotation (about vertical axis) related noise of the 70 degree VGA digiclops camera. Rig poses are obtained from individual images (3 x monocular pose estimation). CN: Corner noise. PR: Pose noise, Rotation. PT: Pose noise, Translation. MSE: Mean Square Error. MN: Maximal Noise value. Larger group values indicate more rotation. Group 0 is planar to camera 0.

x-rotation	Group Code				
	-2	-1	0	1	2
CN MSE [pix]	0.013554	0.011851	0.015991	0.018071	0.023590
CN MN [pix]	0.110767	0.088976	0.189937	0.190439	0.272766
PR MSE x 1000	0.430153	0.275917	0.385048	0.271239	0.292915
PR MN x 1000	1.900671	0.814374	2.709669	0.922019	1.142920
PT MSE [mm]	0.106327	0.049670	0.067220	0.047985	0.195913
PT MN [mm]	0.450244	0.180564	0.368562	0.146918	0.976750

Alternatively the double floating point computations produce the ‘not a number’ result if the Jacobian cannot be inverted during the Newton optimisation.

At first we want to investigate the importance of the image filling rig poses and the different distances. We therefore selected all images as a reference (label ‘all’), the images of distance group 1 only (label ‘dist1’) and combinations of different distances (labels ‘dist01’, ‘dist12’, ‘dist012’, ‘dist013’). A calibration using distance group 0 only is not possible because the optimisation procedure does not converge. Fig. 7.2 and 7.3 display the noise and difference values of these pose groups sorted by camera parameters. The difference values are the absolute values of the differences between the reference calibration and the mean

Table 7.5: Rig rotation (about horizontal axis) related noise of the 70 degree VGA digiclops camera. Rig poses are obtained from individual images (3 x monocular pose estimation). CN: Corner noise. PR: Pose noise, Rotation. PT: Pose noise, Translation. MSE: Mean Square Error. MN: Maximal Noise value. Larger group values indicate more rotation. Group 0 is planar to camera 0.

y-rotation	Group Code				
	-2	-1	0	1	2
CN MSE [pix]	0.015612	0.015693	0.016761	0.013453	0.015938
CN MN [pix]	0.115363	0.189937	0.272766	0.136006	0.112705
PR MSE x 1000	0.253625	0.362478	0.402541	0.282753	0.175790
PR MN x 1000	0.754782	1.196057	2.709669	0.720619	0.451605
PT MSE [mm]	0.101485	0.073741	0.082740	0.050267	0.094232
PT MN [mm]	0.330821	0.282876	0.976750	0.180085	0.368562

of all the calibrations using the respective pose groups. In Fig. 7.2(b) we see that the images of distance groups 0 and 2 are the most influential determining the focal length. All calibrations that include either pose group bring the mean calibration close towards the reference calibration. The calibration containing both groups is almost identical to the calibration containing all images. The noise levels indicate the same as the calibrations containing distance groups 0 and 2 are almost identical to the reference calibrations. Distance group 3 increases the noise level and all calibrations that include this pose differ from the reference calibration more than those that do not include it.

The reason for this effect is that distance group 3 is too far away from the camera for this size of the calibration rig. A shorter distance or bigger squares may improve the results.

Both the projection centre and the distortion parameters are most sensitive to distance group 0. All calibrations containing distance group 0 exhibit smaller differences to the reference calibration and similar noise levels as the reference calibration. The fact that the calibrations containing distance group 0 exhibit a smaller noise level stems from the fact that almost all calibrations containing distance group 0 converge to either of two local minima. Calibrations without distance group 0 exhibit only one minimum.

This was discovered by plotting the projection centre coordinates of all the one hundred calibrations. The coordinates form two clusters for calibrations including distance group 0 and one cluster between them for all other calibrations. If two images of the sequence (e.g. index 12 and 13) were used to calibrate the camera, the number of images that end up in the local minima is reduced.

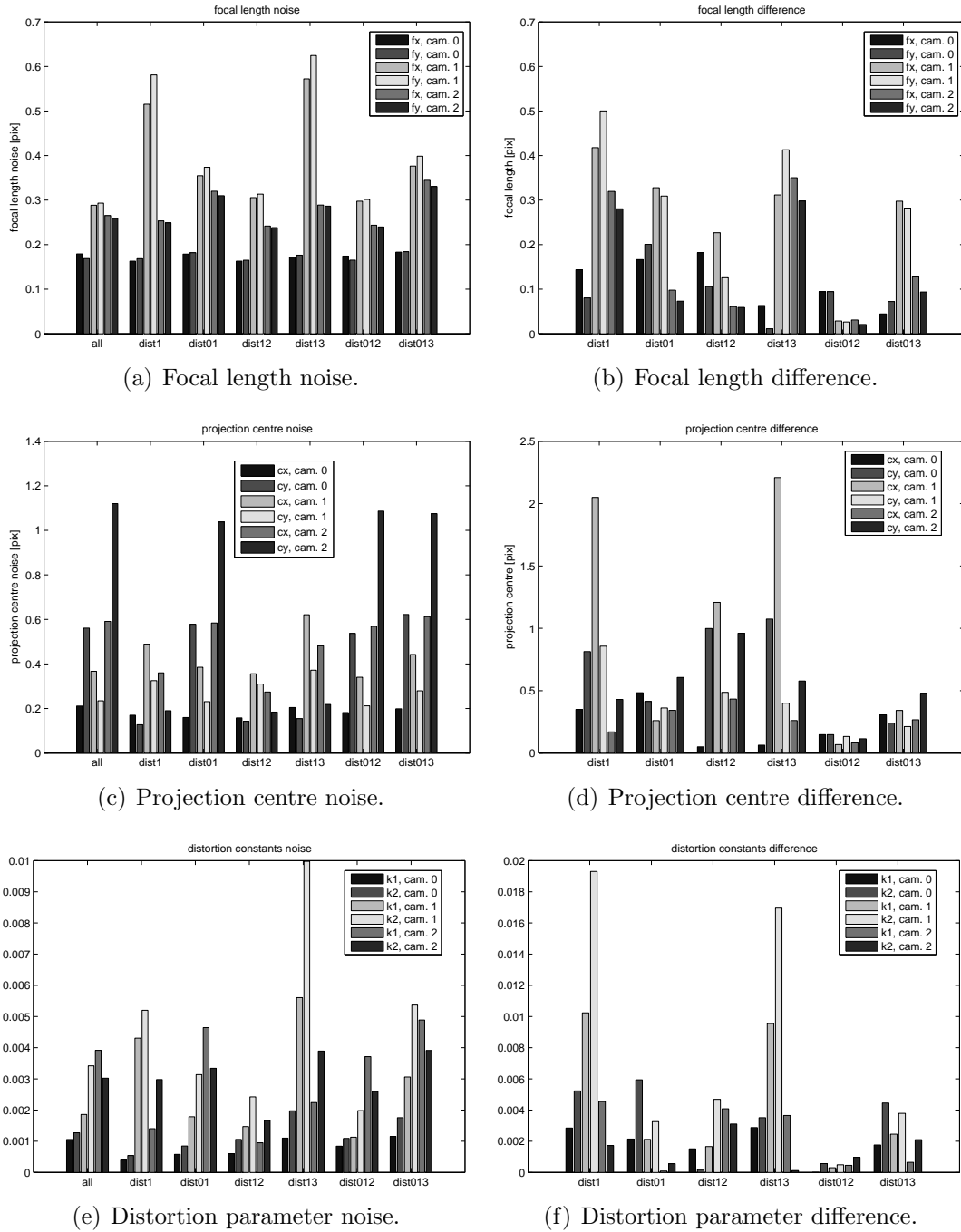


Figure 7.2: Internal calibration noise of the 70 degree VGA digiclops camera. The influence of different distance pose groups on the different camera parameters is depicted.

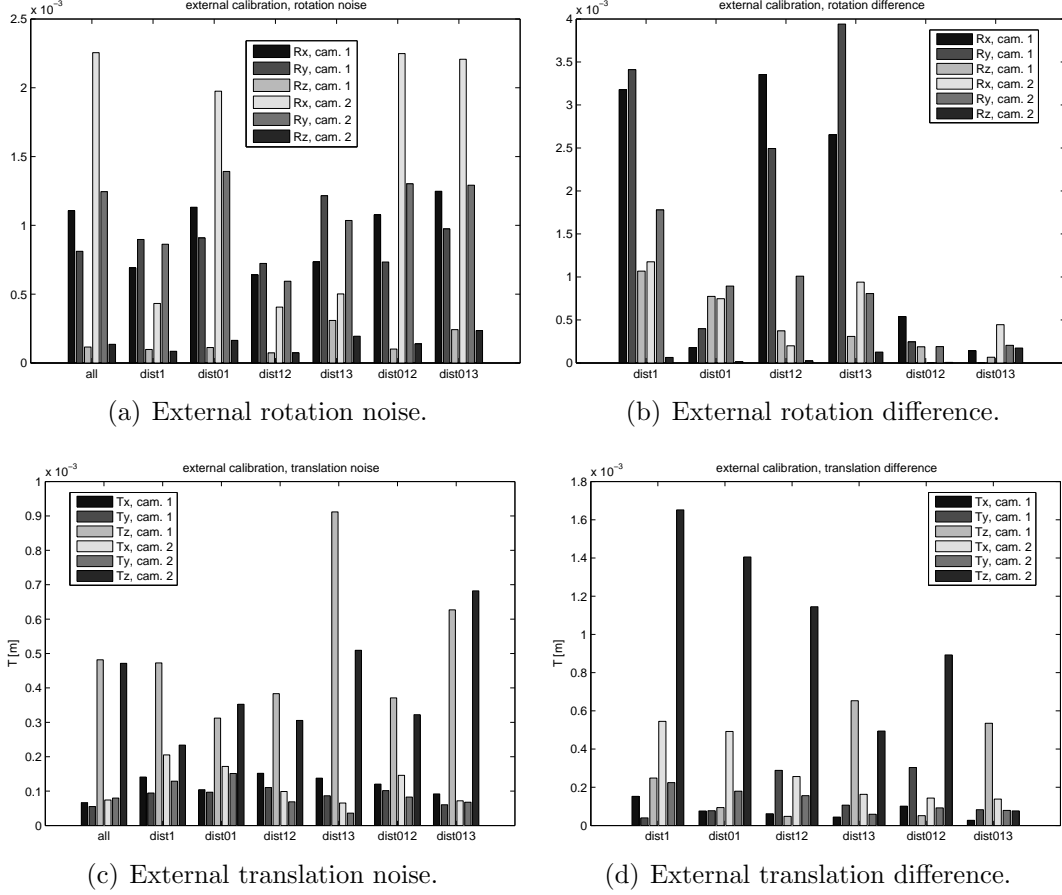


Figure 7.3: External calibration noise of the 70 degree VGA digiclops camera. The influence of different distance pose groups on the different camera parameters is depicted.

We therefore recommend to record multiple images per pose during the calibration whenever possible. It seems that the corner detection noise serves as a regularisation during the optimisation.

The external calibration (Fig. 7.3) suffers from this. All calibrations including distance group 0 exhibit a large noise especially in the x and y rotation components. The difference of these calibrations to the reference calibration is of course larger as both groups centre around two different solutions. These solutions differ in the projection centre coordinates which in turn affect the x and y components stronger because their mean value is much smaller than the larger z component.

The translation component with the largest noise is the z component. This is caused by the fact that the distance of planar objects cannot be estimated as accurately as the lateral

position. Small changes in the feature positions such as caused by the corner detection noise lead to much larger changes in the distance.

Fig. 7.4 and 7.5 depict the noise and difference values of the rotation pose groups. The ‘all’ group denotes the reference calibration, the ‘centerPos’ group contains only poses in the centre of the image (all rotations), the ‘slightRot’ group only small rotation, and the ‘strongRot’ group only large rotations. The two last groups also contain unrotated rigs. Using these statistics we investigate the influence of the rotation angles and the presence of the rigs in the corners.

As seen in Fig. 7.4, the focal length noise strongly depends on the amount of rotation as the ‘strongRot’ group exhibits almost the same noise level as the ‘all’ group. This behaviour is mirrored in the difference to the reference calibration data, where ‘strongRot’ group is closer to the reference calibration than the ‘slightRot’. Omitting the off-centre images leads to a magnitude more noise and 2.5–10 times the difference as the ‘centerPos’ group indicates.

The noise level of the projection centre is almost unchanged by the rotation angle, but strongly influenced by the missing off-centre images. The difference to the reference calibration instead favours smaller rotation angles. This is caused by the fact that the derivatives of the objective function to the centre values is exactly one. In other words the difference of the points in the image and the projected spatial points directly affects the projection centre. The spatial points in turn are proportional to the cosine of the rotation angle. If the angle is small the factor of proportionality is larger than if the angle was larger. It therefore influences the value of the objective function stronger if the angle is smaller. This means that for smaller angles the true value is captured better than for larger angles.

The ‘centerPos’ group produces a large noise in the focal length and projection centre parameters because the extracted coordinates are almost linear dependent, the system of equations to be solved is close to singularity. The slope of the function is therefore much higher here, so that the same corner noise level leads to a higher parameter noise level.

The distortion noise values are almost identical to the reference calibration. The distortion values of the ‘centerPos’ group are a little bit higher than those of the reference values. This is only naturally as the optimisation is close to a singularity. The distortion values therefore depend more strongly on the image filling rig positions than on other positions.

The external rotation noise is basically unaffected by both the missing off-centre images as well as the rotation angle. The rotation difference to the reference calibration is strongest

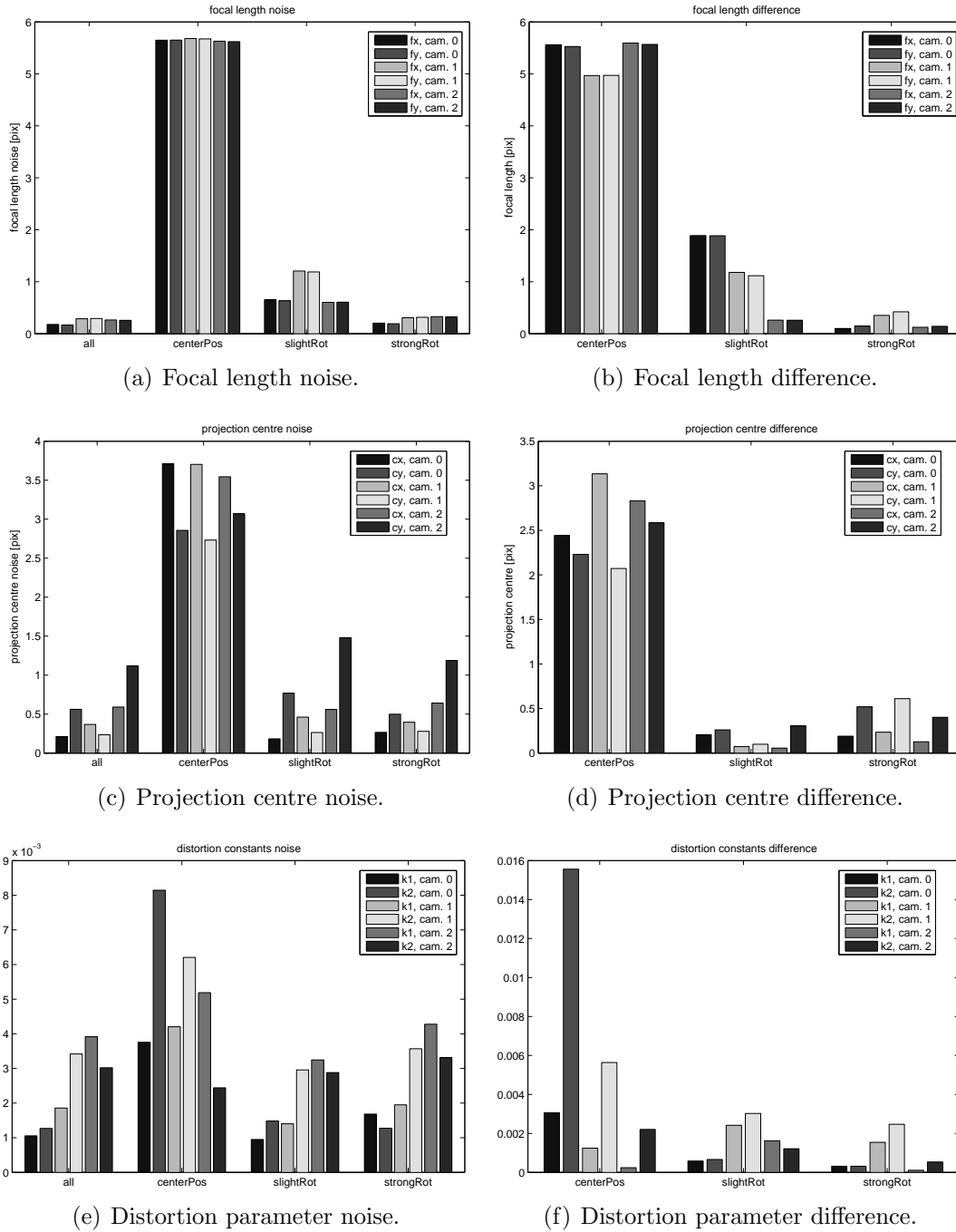


Figure 7.4: Internal calibration noise of the 70 degree VGA digiclops camera. The influence of different rotation pose groups on the different camera parameters is depicted.

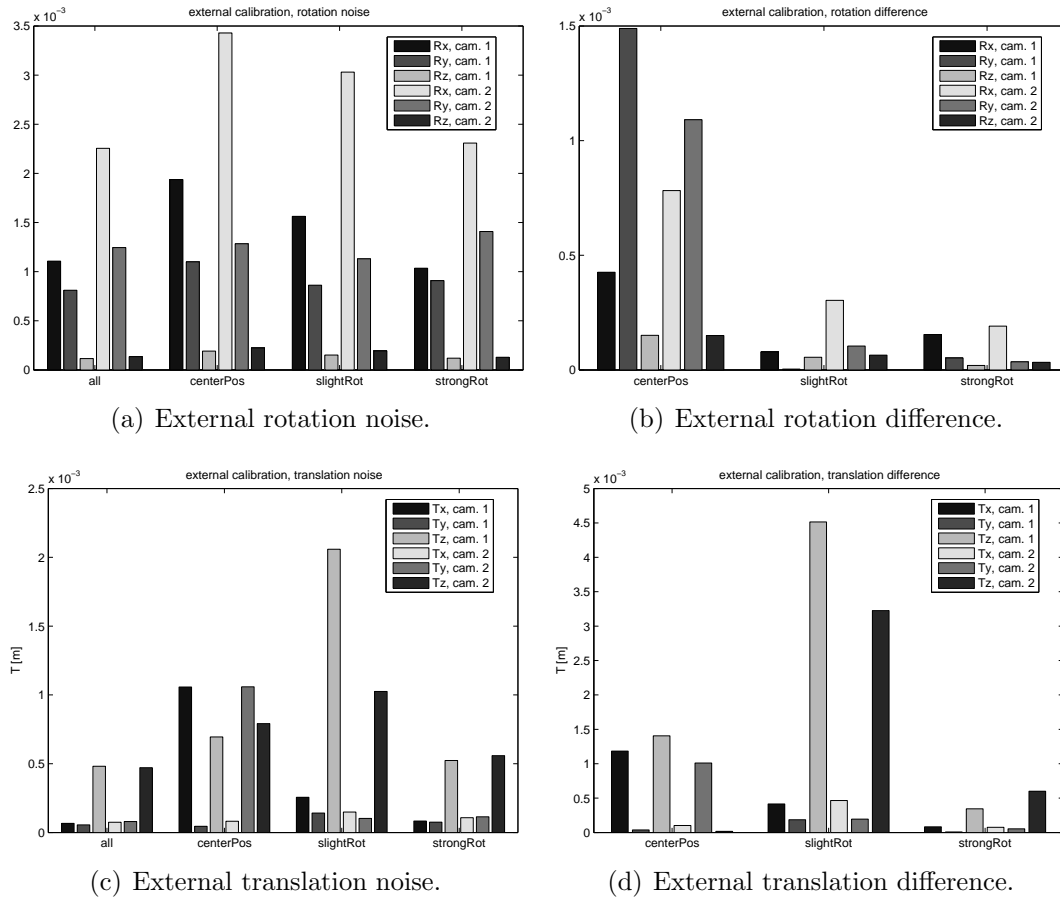


Figure 7.5: External calibration noise of the 70 degree VGA digiclops camera. The influence of different rotation pose groups on the different camera parameters is depicted.

for the ‘centerPos’ group. The difference of the rotation groups is almost non-existent. This leads to the conclusion that rotation of the rig does not have a strong influence on the external rotation. Instead the off-centre images have a big influence on these parameters.

The translation noise and difference values indicate that the relative camera positions, especially the z positions depend on the rotation angles more than on the off-centre images. The larger difference of the ‘slightRot’ group compared to the ‘centerPos’ group shows that.

So far we have investigated the influences of the rig positions on the calibration parameters and found supporting evidence for the rules of thumb. Additionally we found an additional rule to overcome local minima.

7.5 Choice of External Calibration Algorithm

In this section we investigate how well each of the external calibration algorithms performs. The quality of three external parameter sets are obtained and compared: The initial parameters using linear algebraic methods, the parameters after optimising spatially according to Eq. 3.17, and the parameters after bundle adjustment according to Eq. 3.14.

We use the respective external parameters to project the rig points into the image and obtain the error using Eq. 3.14. The internal parameters are kept constant and identical during the evaluation of the two methods.

Fig. 7.6 depicts the actual reprojections. The left column depicts a zoom into camera 1, the right column into camera 2. Initialisation, spatial optimisation and bundle adjustment are depicted from top to bottom.

Fig. 7.7 depicts the error distribution over all three cameras and the three image tuples used for calibration. Each marker denotes the function value according to Eq. 3.14 of a single rig point. The stroked circle indicates the square root of the mean square error (MSE) i.e. the square root of the mean of the square length of each vector from origin to marker. The dashed circle indicates the maximal radius (MR).

The MSE after the initialisation is 2.7 pixels at an MR of 6.1. The spatial optimisation reduces the MSE to 0.3 pixels with an MR of 1.1 pixel. The bundle adjustment further reduces the MSE to 0.15 pixel and the MR to 0.6 pixel.

We see that the linear initialisation produces an extremely non-gaussian distribution. This is to be expected since the parameters are not at an optimum of the objective function and therefore still contain systematic errors. The spatial optimisation yields a far better approximation of the external parameters at comparatively small computational costs. The bundle adjustment as expected removes the remaining systematic errors and yields MSE's comparatively to the accuracy of the corner detector.

In this somewhat extreme example (three image tuples are the absolute minimum for a calibration) we have seen that the linear initialisation is sufficient to get both the spatial optimisation and the bundle adjustment started. The spatial optimisation is an effective method to reduce the error at a low problem dimension. Its result is quite close to the optimum and can serve as a starting point for the final optimisation using the computationally more expensive bundle adjustment method.

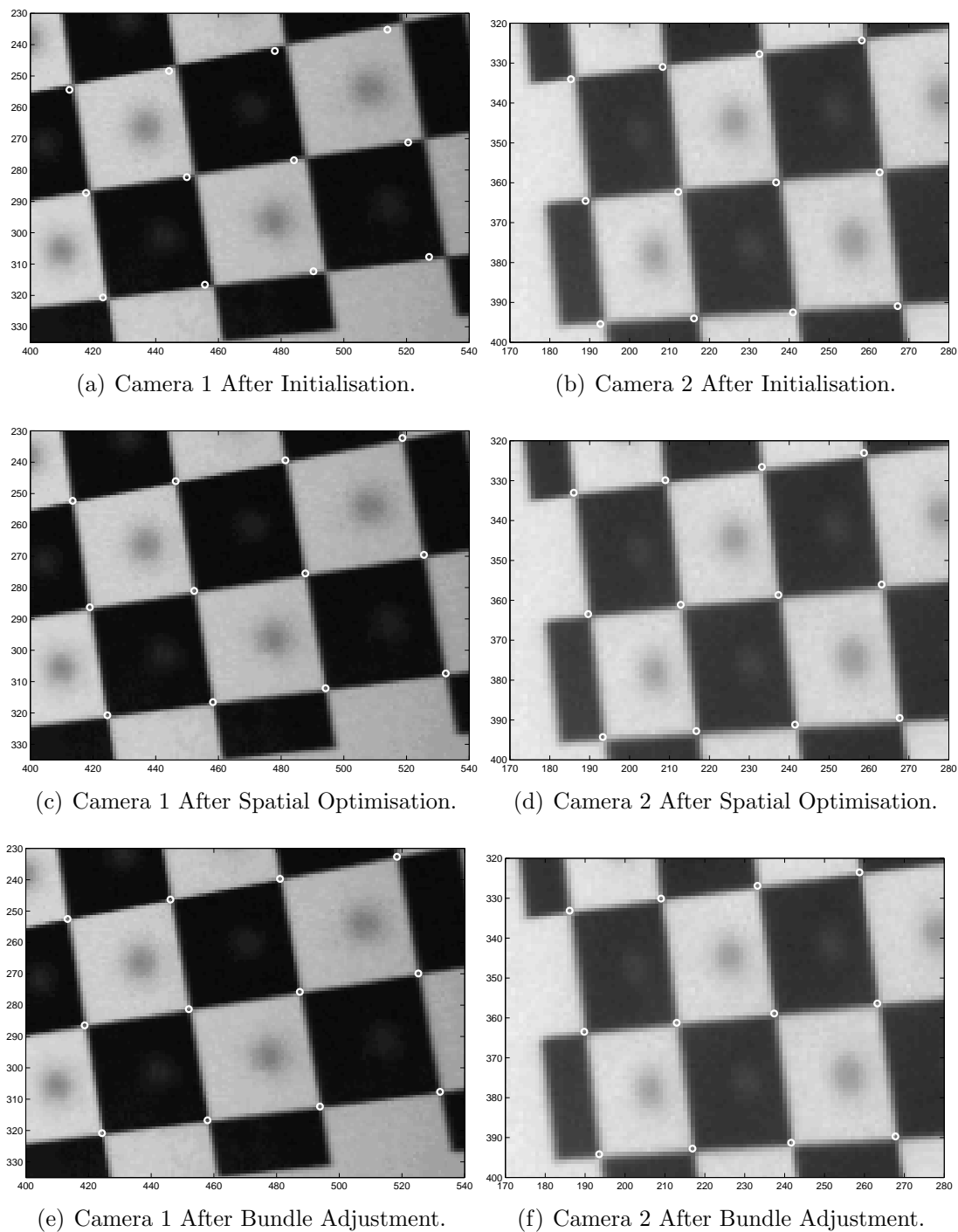


Figure 7.6: Back projection of the rig corners using obtained external calibration.

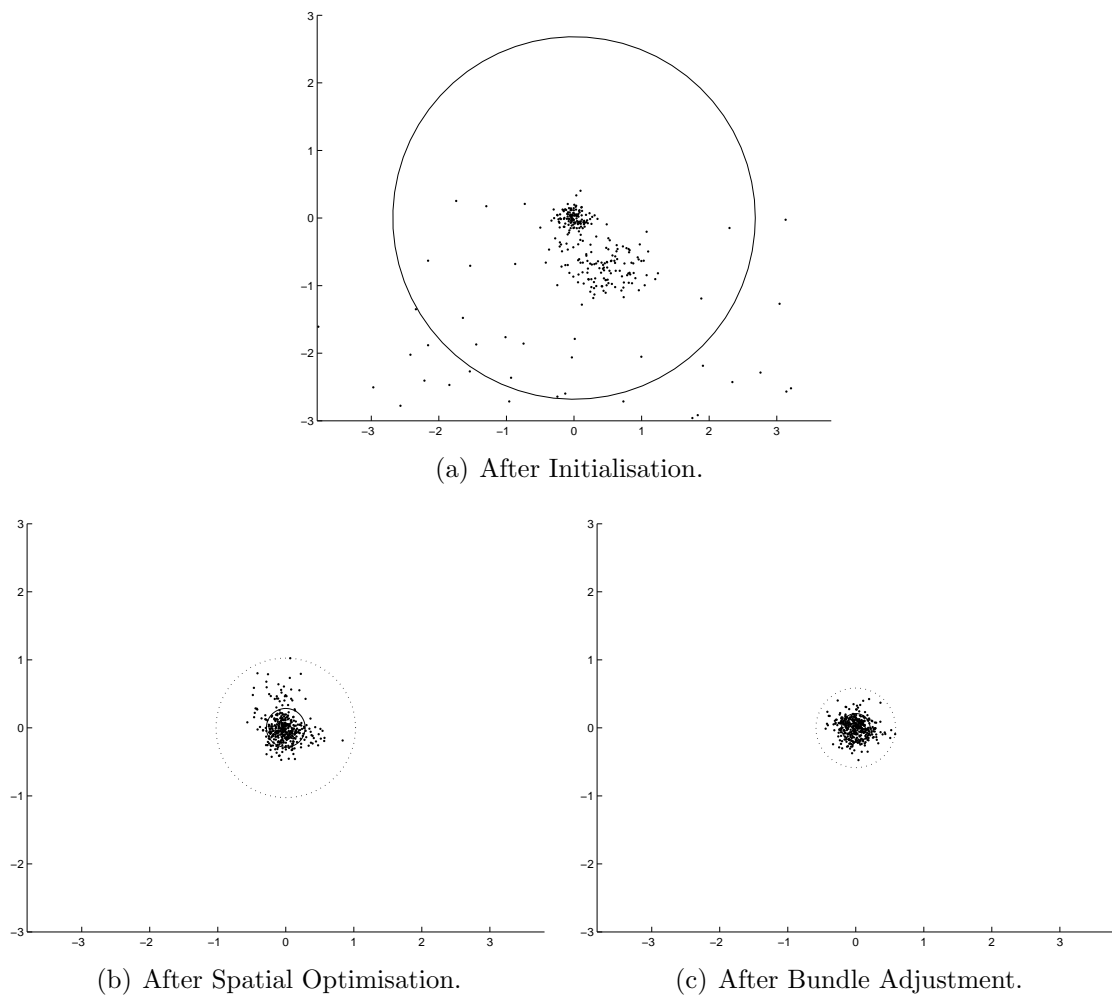


Figure 7.7: Distribution of the projection errors after external calibrations. Dot marker indicate the difference between reprojected point and corresponding corner detection. Stroked circle depicts square root of mean square error (MSE), dashed circle depicts maximal error.

7.6 Summary

The camera calibration and the pose estimation of the calibration rig are characterised by the following performance numbers:

Feature	Performance
Corner detection accuracy (1σ)	0.018 pixel
Corner detection accuracy (worst case)	0.3 pixel
Rotation error	0.015°
Translation error	0.14 mm at 160 cm distance
Focal length uncertainty	0.1%
Projection centre accuracy	1 pixel

The following non-quantitative insights have been found among other:

- Overexposure leads to inaccurate corner detection, but does not hinder detection.
- The rules of thumb regarding rig placement could be confirmed.
- Images taken beyond a certain distance reduce the accuracy of the focal length estimation. For a camera of 70° field of view and VGA resolution this distance is between 120 cm and 160 cm. This fact is not mentioned in the literature.
- All internal camera parameters are subject to local minima during the calibration. Including additional images at same location alleviates the effect. This is also a new insight not found in the literature.
- The off-centre images are important for the external rotation parameters. This insight is not found in the literature.
- In order to speed-up the calibration an optimisation using a metric error function could be used before the Bundle Adjustment. It can reduce the reprojection error by about $\frac{2}{3}$ at a fraction of the computation effort.

8 Oil Cap Inspection

8.1 Goal of Investigation

The oil cap of an engine is a real-world example for the necessity of visual inspection during production. It is rather easy to mount the cap incorrectly such that the oil circulation is not sealed. This might not be noticed during the subsequent test run, as the loss of oil is slow. In the long run, the non-sealed oil circulation leads to higher maintenance cost (oil replacement) or possible engine damages and pollution due to lost oil.

The inspection task is to obtain the pose of the oil cap with respect to the engine. We assume that the transform from engine to camera is known for each image in a real-world application and therefore the determined object pose can be compared directly with a pre-defined reference pose.

Additionally we performed investigations of the methods to determine their classification capabilities among different types of oil caps. A practical use for the classification is the check if the correct type of oil cap is used, if there are e.g. different caps for diesel and gasoline engines using similar sockets.

8.2 Experimental Setup

In our experimental investigations we obtain the ground truth by the use of a goniometer*. To determine the absolute pose without a big mechanical effort concerning the mounting of the oil cap is necessary. For doing so we would need to calibrate the goniometer axes to the camera coordinate system. This would require a rigid mounting between goniometer and camera as well as a rigid, precise, and replaceable fastening of oil cap and e.g. a chequerboard rig. Although not technically impossible, the mechanical effort is very high. Our approach is to use the high precision of our goniometer ($\pm 0.0005^\circ$ repeatability) to provide a ground truth for pose differences by comparing pose differences only.

*Provides two orthogonal axes of revolution around a common point.

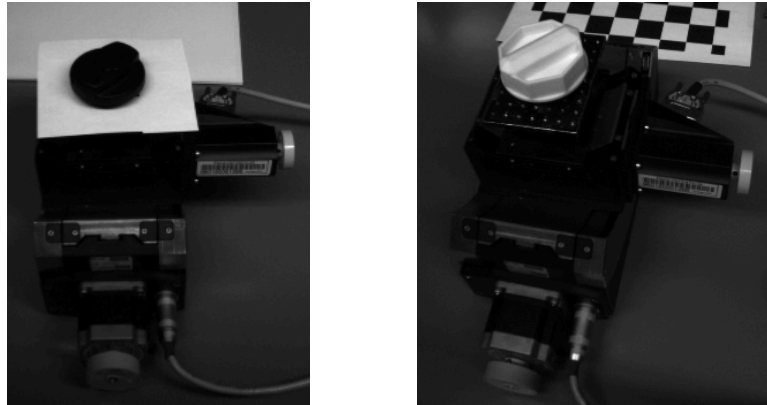


Figure 8.1: Experimental setup and input data. The elongated objects in the lower and the right parts are the motors of the goniometer. The oil cap is mounted on a plate that is fastened to the goniometer. The different grey values are compensated in the feature extraction such that only the shape of the object is used to distinguish them.

For the image recording we placed the camera on a tripod, modified the pose of the object using the electrical motors of the goniometer, and recorded 100 image tuples for each of the 35 poses. This will give us sufficient statistics to evaluate the influence of image noise on the pose estimation accuracy. Fig. 8.1 gives an overview of the setup and the input data.

Although the two oil caps differ in grey value, only their shape is important in this investigation. The black oil cap has a round circumference, while the white oil cap has a hexagonal circumference. The grey value invariance is realised by limiting the edge directions to a half-circle and mapping opposite gradient directions to the same edge direction code.

All images were labelled with their respective goniometer angles. For all unique pairs of goniometer settings of the per-axis set $[\pm 20^\circ, \pm 10^\circ, \pm 7.5^\circ, \pm 5^\circ, \pm 4^\circ, \pm 3^\circ, \pm 2^\circ, \pm 1^\circ, 0^\circ]$ the rotation matrices were computed using the kinematic model of the goniometer. These rotation matrices give the orientation of the rotated oil cap coordinate system with respect to the non-rotated oil cap coordinate system. The largest rotation angle between the corresponding axes (e.g. non-rotated Y axis to rotated Y axis) of the two selected goniometer settings was computed and constitutes the ground truth for the 100 images.

The poses of two images were obtained and the pose difference (rotational only) was computed in the same way as the pose difference of the goniometer settings. The difference between these ground truth pose difference and the estimated pose difference was stored and will be displayed as histograms.

In order to provide a fair comparison of the capabilities of the different methods under identical conditions in the same task we used the following approach: Since the matching algorithms can yield multiple detections even if only one object is present, we considered the pose of the best matching result only. The matching parameters of all methods were set to identical values (if applicable).

Parameters were set using a small set (< 10) of test images out of the full set. This corresponds to the set up phase after the deployment of an real world inspection system. The parameters were chosen such that the smallest number of results — one, whenever possible — was generated. We have therefore tuned the methods to yield one results per image. This setting might not provide the smallest pose error. It allows, however, to compare the methods under defined conditions even if they cannot be compared at absolutely identical parameter values as different methods have different parameters.

This procedure will eliminate all systematic pose errors from both the ground truth and the input images. Due to this systematic errors of the pose estimation (e.g. wrong model size, wrong calibration parameters, etc.) cannot be found. This is not a serious obstacle as these errors are easily found in practice.

In addition to the pose we recorded detection and uniqueness rate. The detection rate indicates the number of images (relative to the total number) where any number of results was generated. The uniqueness rate indicates the number of images where a single result was generated. Ideally, both rates should be 100% in our experiments, thus exactly one detection took place, as exactly one object was present. A detection rate lower than 100% indicates that the parameters are not tuned careful enough. A uniqueness rate less than 100% means that the method yielded multiple results. In our experiments only one object was present, so this indicates false positive matches.

The recording procedure for the distance estimation experiments was similar. The camera was placed on a photographic copy stand, looking straight down. The height of the camera was manually controlled using the ruler of the copy stand. The data analysis was performed on distance differences.

Furthermore we recorded the number of results per image along with the memory and run-

Table 8.1: Investigated methods and their abbreviations in the plots.

Abbreviation	Algorithm
direct 3	Trinocular Template Matching using Direct Pose Association, Sec. 4.1
direct-hier 3	Trinocular Template Matching using Direct Pose Association and hierarchical evaluation, Sec. 4.1
direct 1	Monocular Template Matching using Direct Pose Association, Sec. 4.1
sim 3	Trinocular Template Matching using Similarity Pose Association, Sec. 4.1
fpm 3	Trinocular Feature Pose Maps, Sec. 4.2
fpm-dil 3	Trinocular Feature Pose Maps on dilated edge images, Sec. 4.2
fpm-dst 3	Trinocular Feature Pose Maps using distance voting, Sec. 4.2.2
gst 3	Trinocular Gradient Sign Table matching using distance-image coordinate gradient, Sec. 6.2
gst-ref 3	Trinocular Gradient Sign Table matching using distance-pose gradient, Sec. 6.2

time requirements of the different methods. The methods in Table 8.1 were investigated. Their abbreviations for Fig. 8.2 through 8.15 are also given in the table.

The Gradient Sign Tables here encode the gradient of the distance to the nearest feature over the image coordinates. We therefore performed the distance transform on-the-fly without additional distance transformed images.

We reduced the number of methods so that all important aspects (trinocular vs. monocular, top-down vs. bottom-up, etc.) are covered and the number of methods remains small. We will display the error to the ground truth over a large set of poses using error histograms and their quantiles. We computed the location accuracy only for the real world oil cap in order to compare it to the other publications concerning this object.

All methods operate on the same edge images and generate their data from the same templates. The templates were generated with 8 edge directions from a re-engineered CAD model. The edge extraction was parametrised so that only the outline of the object was visible. The white warning label on the black cap was not reproduced in the model so we over-painted it on the real-world object too. This, too, leaves the outline of the real-world object only.

8.3 Pose Estimation Accuracy

8.3.1 Rotational Accuracy

The accuracy of the rotational pose components is crucial for the discrimination between a correctly and an incorrectly mounted oil cap. Additionally small memory consumption and short execution times are also important. A high detection rate is of course favourable. Additionally, a small number of detections per image is desirable in order to reduce the risk of false positive matches.

We generated the templates with 2 mm spatial resolution and 4° angular resolution. The ranges were selected from example images using manually selected 2D-3D correspondence points and subsequent 2D-3D pose estimation. In the following we state the observed effects. As many of these effects share the same reason, we will state this reason in Sec. 8.3.5.

In Fig. 8.2(a) and 8.2(b) the comparison between the trinocular and the monocular template matching is made. The monocular matching procedure is actually better at capturing the appearance differences due to object rotation. This is supported by the smaller values for the four quantiles and the mean error. Given the same parameters, the detection rate of the monocular version is strongly reduced while producing a larger number of similar good results (low uniqueness rate, Fig. 8.5(b)). This indicates that monococular matching procedure is less robust and fails more often in ambiguous situations. This is to be expected as no spatial information is considered in the match.

The mean error of the trinocular matching is less than 50% of the template resolution; that of the monocular matching less than 30%. The 99% quantile — which corresponds to 3σ assuming an underlying Gaussian distribution of the errors or the maximal error discounting outliers — is 1.5 times the template resolution for the trinocular matching and close to the angular resolution for the monocular matching. Possible reasons for the smaller errors of the monocular matching are discussed in Sec. 8.3.5

Fig. 8.2(a) and 8.2(c) compare the trinocular Template Matching methods using direct and similarity pose-template association. The histograms are nearly identical, with the similarity pose-template association possessing slightly smaller errors (cf. Fig. 8.4 for a direct comparison of the values). This indicates that the re-use of similar looking templates does not have a big negative influence on the pose accuracy.

Comparing the trinocular template matching with the classical Feature Pose Maps matching (Fig. 8.2(a) and 8.3(a)) shows that the error of the classical FPM is much larger ($2\times$

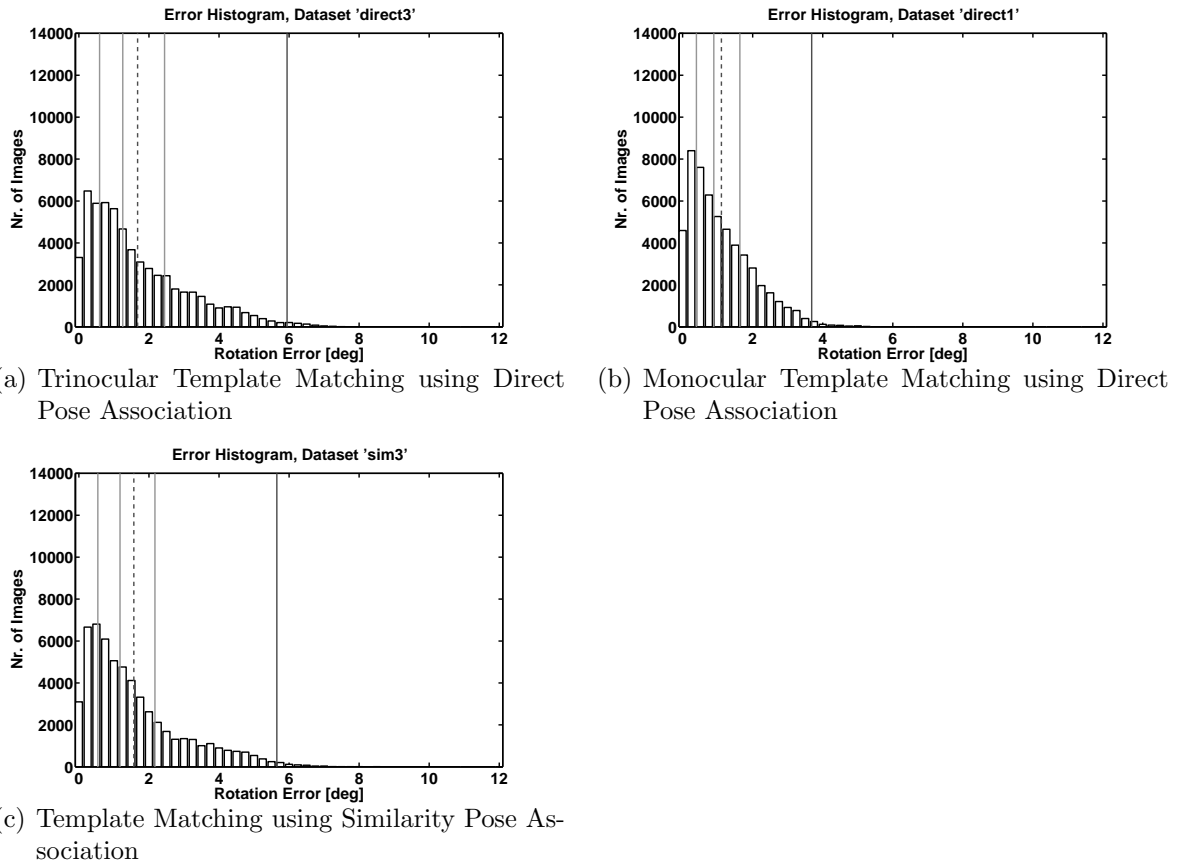


Figure 8.2: Rotation error histograms of pose estimation methods. Solid grey lines: 25% quantile, 50% quantile (median), 75% quantile. Dashed line: mean. Solid black line: 99% quantile.

the template resolution). This is to be expected from the principle and was described in [10, 82]. Since the templates are quantised, the probability of an image edge pixel leading to a vote for the correct pose is rather small. This increases the probability of voting for a wrong pose since all poses are equally bad.

Our proposed modifications (dilatation of the edges, Fig. 8.3(b) and distance voting, Fig. 8.3(c)) perform very well in terms of accuracy. The distance voting procedure is subject to more outliers as seen in the 99% quantiles (Fig. 8.4(d)). The FPM matching on dilated edges has a robust maximal error of about the angular template resolution. The distance voting yields a similar mean error and similar quantiles, but is more prone to outliers as the larger 99% quantile of $1.5\times$ the template resolution indicates. Reasons for these effects are also discussed in Sec. 8.3.5.

The Gradient Sign Table matching (Fig. 8.3(d)) which performs the distance transform on the fly performs slightly worse than the trinocular template matching. There are no peculiarities in the histograms (such as more outliers). This lead to the conclusion that the error is actually larger over all experiments.

Directly comparing the detection and uniqueness rates (Fig. 8.5) shows that we succeeded in setting up the methods such that a high detection rate is guaranteed. The disadvantage of the monocular matching easily seen: The method has the worst detection rate. The monocular method produces 3.5% false alarms (for missing oil caps) compared to the trinocular variant that does not increase the false alarm rate in this way.

The similarity pose association is almost as good as the trinocular template matching, but at the expense of a strongly reduced rate of unique results. This rate is even worse than those of the feature pose maps and the GST. The FPM and GST matching methods have also quite high detection and uniqueness rates with the distance voting outperforming the other two regarding uniqueness.

8.3.2 Influence of Hierarchical Evaluation

The experiments above evaluated all templates. This section investigates how the hierarchical coarse-to-fine search influences the localisation accuracy. We also compare how accurate the Gradient Sign Tables are when the distance-pose gradient is tabulated.

We used a spacing of 4/2/1 lattice steps for the Template Matching (every fourth pose value on the coarsest level for all DoF). The finest resolution used the same thresholds as the experiments involving all templates.

The Gradient Sign Tables as a pose refinement method were initialised also at every fourth lattice step and iterated to convergence. Since the method does not compute an absolute measure of fitness we used the position of a pose in the optimisation sequence as an interpolation weight. Thus the pose that was computed when the iteration stopped was given a weight of 1, the initial pose was given a weight of 0. Local minima in the average sequence position were detected and the values were used to compute weighted average over a fixed neighbourhood.

Fig. 8.6 through 8.8 give the results of these experiments. The hierarchical evaluation has an error about 20% larger than that of the full evaluation. The error measures of the GST

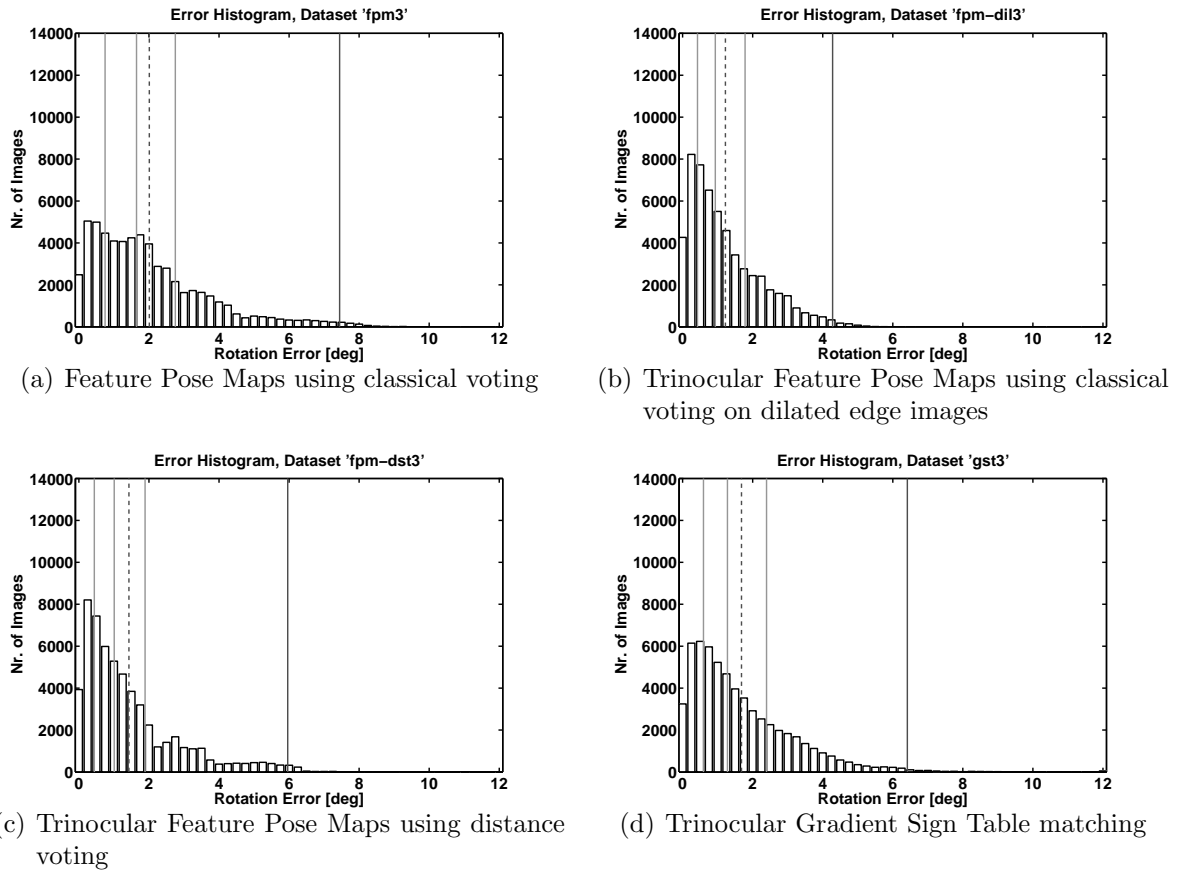


Figure 8.3: Rotation error histograms of pose estimation methods. Solid lines: 25% quantile, 50% quantile (median), 75% quantile. Dashed line: mean. Dash-dotted line: 99% quantile.

matching are all about twice as high as those of the trinocular Template Matching with full evaluation.

The maximal error of the GST matching (13°) slightly exceed 3 lattice positions, in contrast to 2 lattice positions for the Template Matching. Otherwise the histogram exhibits no peculiarities which indicates a larger noise in the estimated poses, but no systematic error.

Sec. 8.3.5 also discusses the reason for these results.

8.3.3 Accuracy of the Depth Estimation

The accuracy of the depth (object distance to camera) estimation is crucial for determining whether the oil cap is only placed on top of its socket or fastened in the socket. The

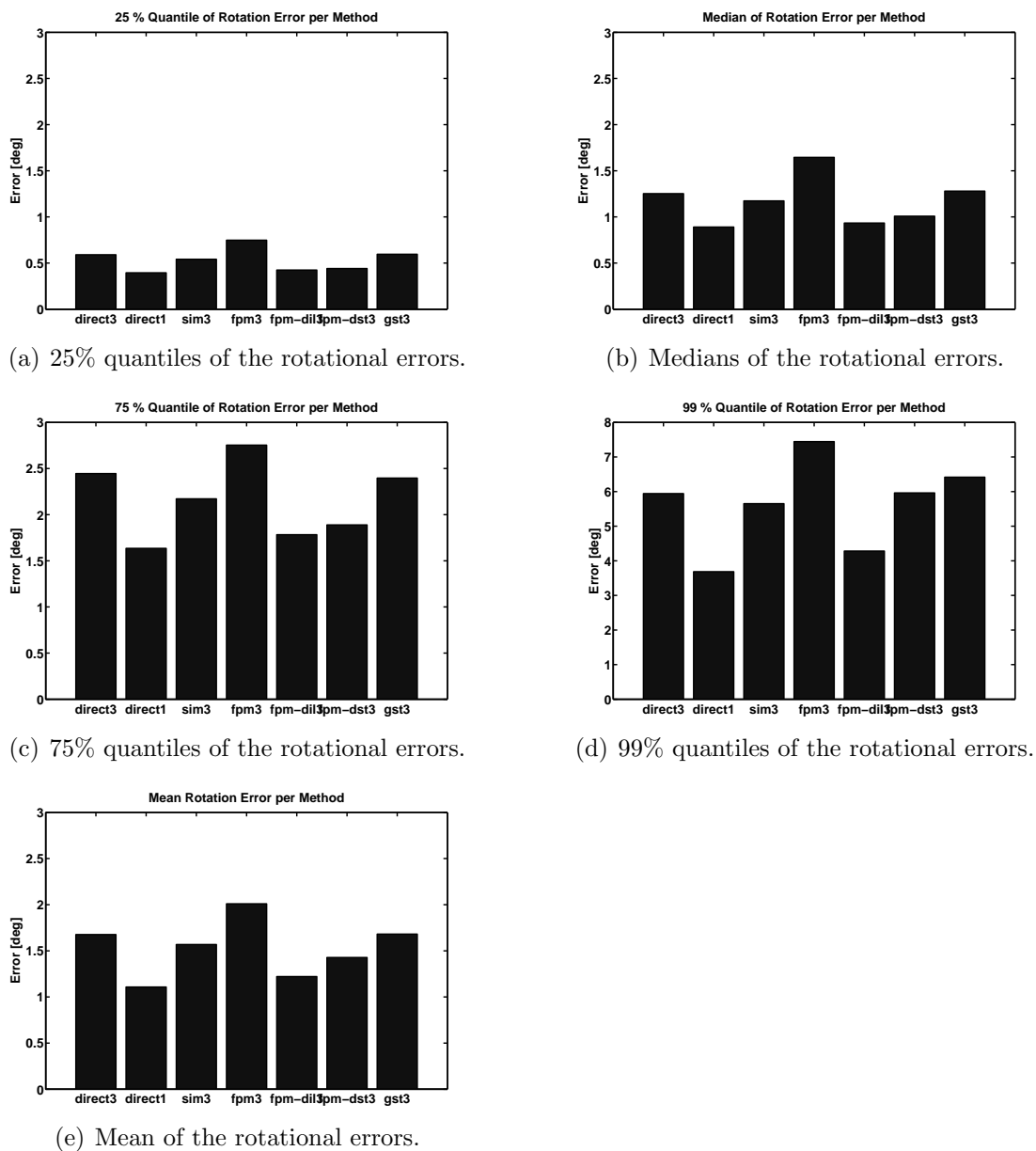


Figure 8.4: Quantiles and mean of the rotational errors for different pose estimation methods.

2D Template Matching in [79] solved this by looking at the object in an acute angle and comparing the v-coordinate of the matching template with a fixed value.

The depth estimation is also a weak point of monocular methods, which obtain their size hints from the scaling of the object alone. Trinocular methods can also use the disparity

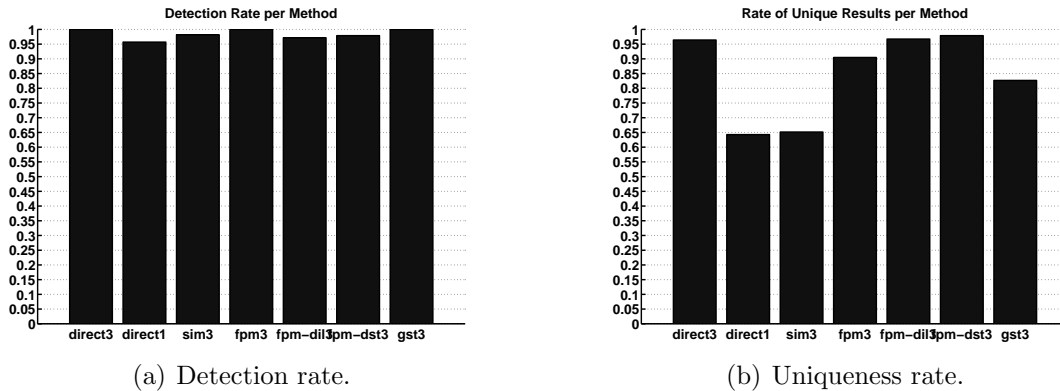


Figure 8.5: Detection and uniqueness rate for rotation angle estimation of different pose estimation methods. The detection rate indicates the number of images (relative to the total number) where any number of results was generated. The uniqueness rate indicates the number of images where a single result was generated.

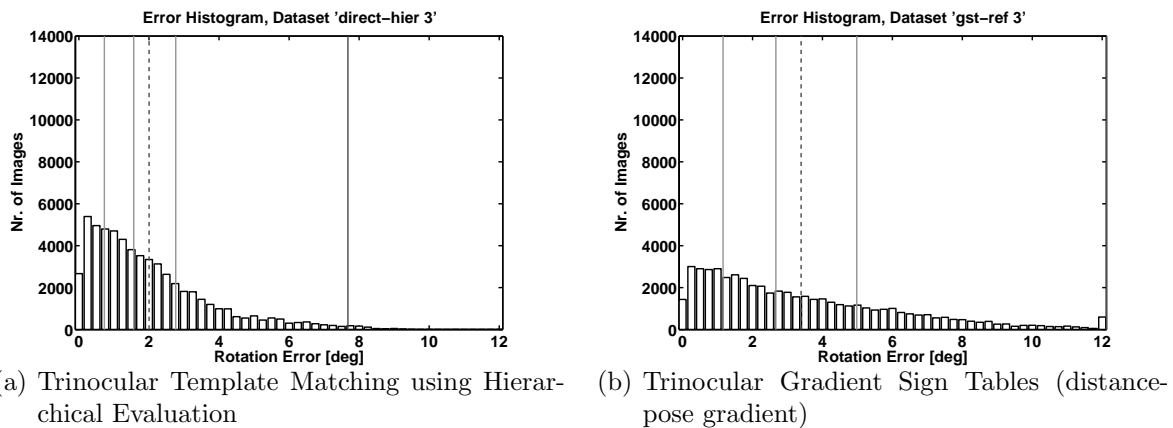


Figure 8.6: Rotation error histograms of hierarchical pose estimation methods. Solid lines: 25% quantile, 50% quantile (median), 75% quantile. Dashed line: mean. Dash-dotted line: 99% quantile.

— a much more accurate measure — to estimate the depth of the object.

We generated the templates with 2 mm horizontal and vertical resolution and 1 mm depth resolution. The rotation angles remained constant. The depth ranges were selected again from example images using manually selected 2D-3D correspondence points and subsequent 2D-3D pose estimation. Two sectors of templates were generated, one corresponding to small disparity and one to a larger disparity. Using these sectors different sensitivities of the non-linear depth-to-disparity function could be evened out.

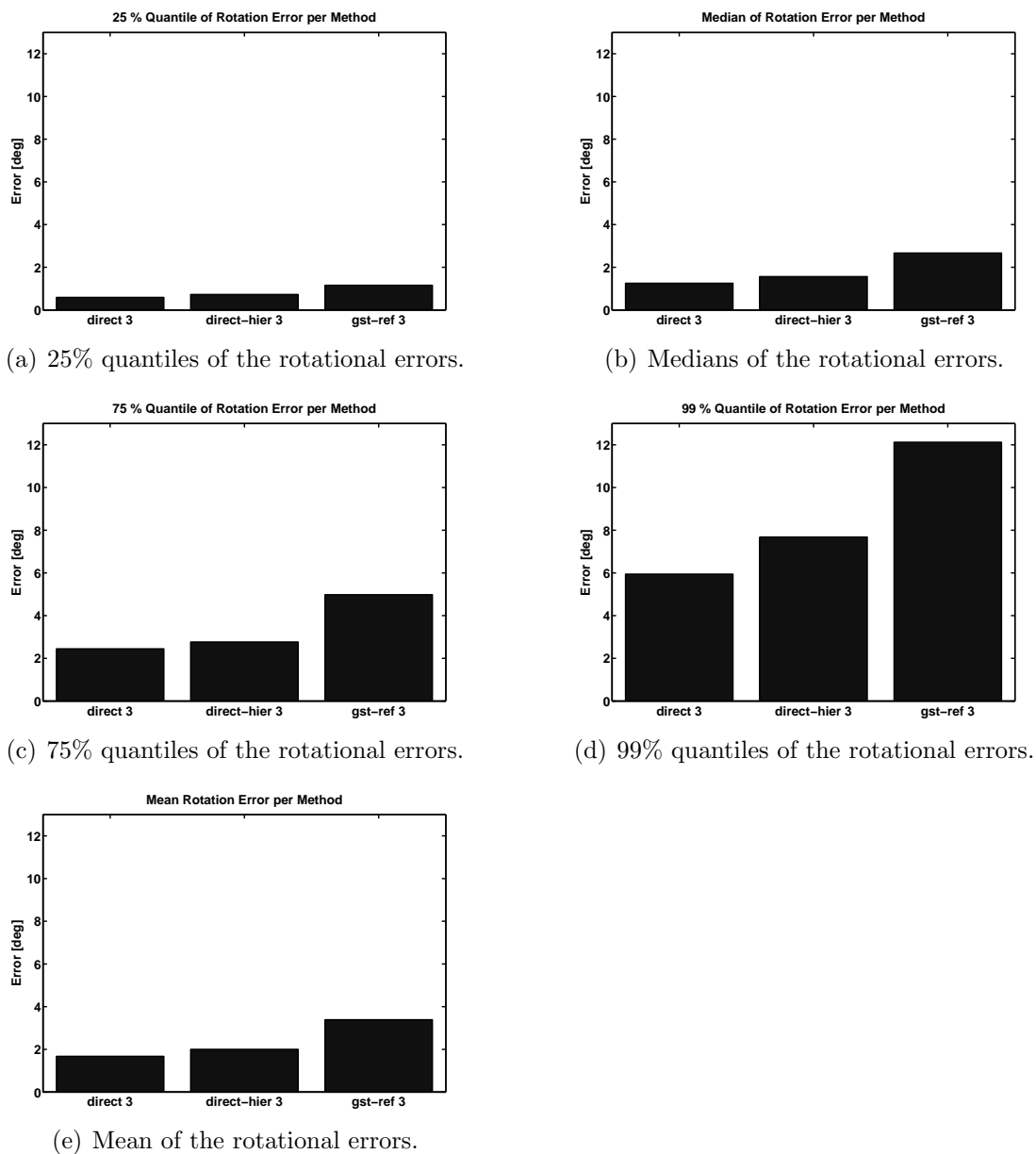


Figure 8.7: Quantiles and mean of the rotational errors for hierarchical pose estimation methods. Mind the different range.

We will not analyse the run-time and memory consumption in these experiments as they scale linearly with the number of templates and therefore provide little new insight.

Using Fig. 8.9(a) and 8.9(b) we compare the depth estimation capabilities of trinocular vs. monocular template matching. As expected, the monocular version performs much

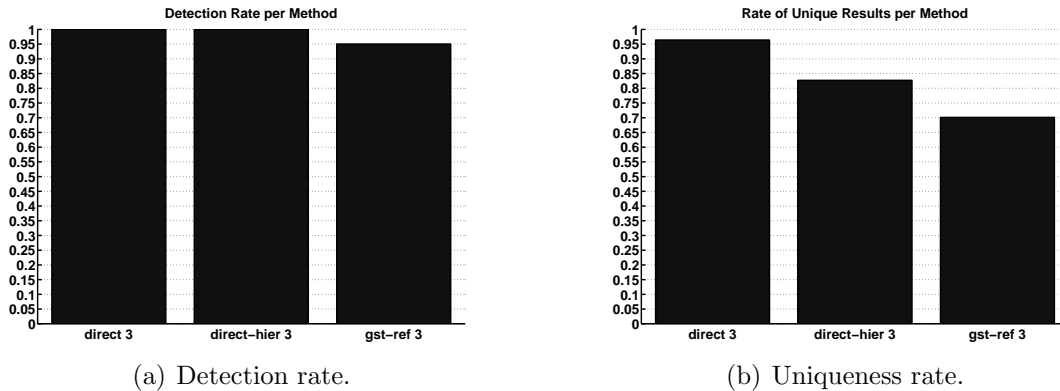


Figure 8.8: Detection and uniqueness rate for rotation angle estimation of hierarchical pose estimation methods. The detection rate indicates the number of images (relative to the total number) where any number of results was generated. The uniqueness rate indicates the number of images where a single result was generated.

worse than the trinocular version. The multiple peaks of the monocular errors indicate a multimodal distribution and therefore a higher sensitivity to local minima and outliers. The trinocular version is subject to outliers: These are caused by two of the 3500 images where the result is picked from the wrong sector due to similar local minima.

Fig. 8.9(a) and 8.9(c) show — as for the rotation experiments — that the reduction of the templates according to their similarity does not have a large effect on the quality of the matching.

Although Fig. 8.10(a) exhibits almost the same quantile and mean errors, one may observe a stronger trend to larger errors. This trend may worsen if e.g. the horizontal or vertical translation or the rotation of the object changes and is to be estimated together with the depth.

The use of dilated edge images (Fig. 8.10(b)) shows nearly identical values for the lower quantiles compared the trinocular Template Matching, but mean and robust maximum show a significantly larger number of outliers.

The use of distance voting for the FPM matching (Fig. 8.10(c)) yields excellent results, about twice as good as the Template Matching in the maximal error. The method is subject to fewer outliers and can therefore be considered more stable. This is also seen in much higher uniqueness rate compared to that of the template matching.

The Gradient Sign Table matching (Fig. 8.10(d)) is subject to a bi-modal error distribution

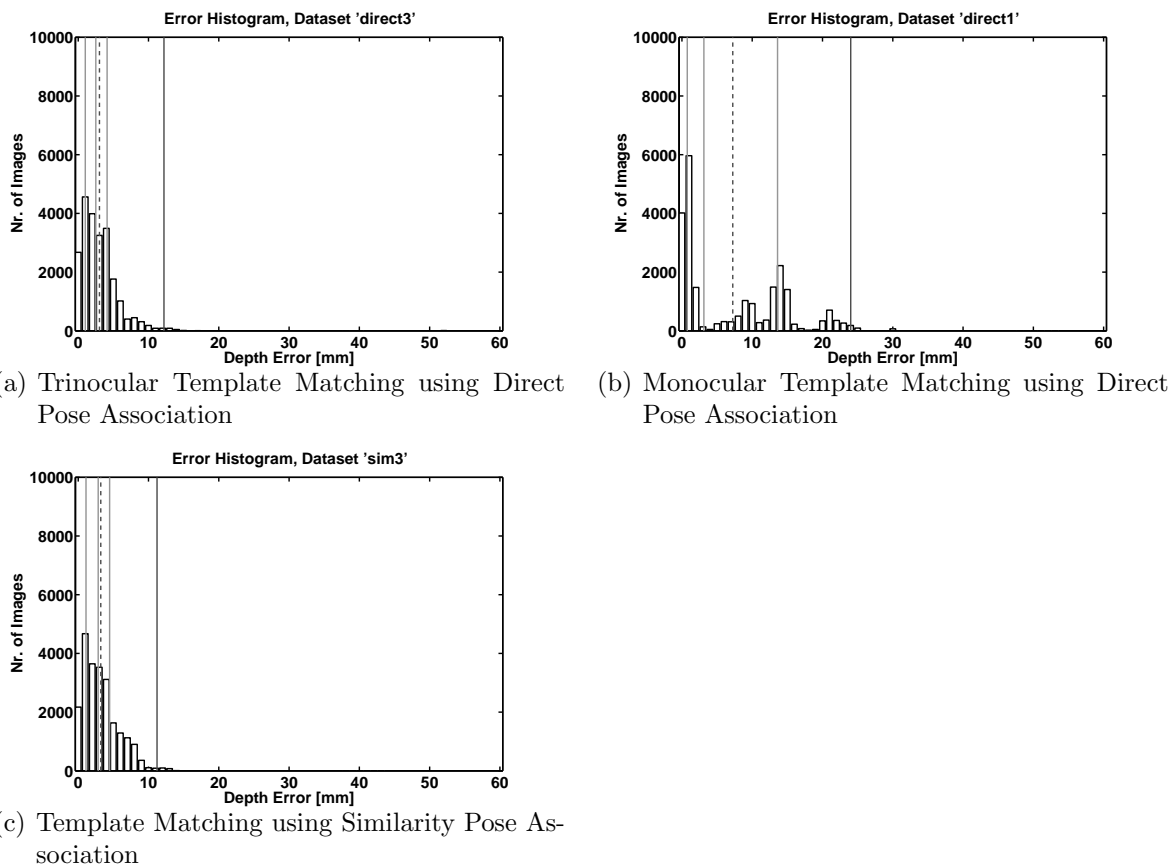


Figure 8.9: Depth error histograms of pose estimation methods. Solid grey lines: 25% quantile, 50% quantile (median), 75% quantile. Dashed line: mean. Solid black line: 99% quantile.

and therefore possesses similar lower quantiles and slightly larger mean and robust maximal values than the Template Matching.

The detection rates (Fig. 8.12(a)) are equally high for all methods, indicating that the matching parameters are set up correctly. The uniqueness rate (Fig. 8.12(b)) indicate that the depth estimation is a more complicated problem than the rotation angle determination (Fig. 8.5(b)). The non-perfect detection rate of the Template Matching results from matching parameter that are set a little bit too strict. Even at this too strict setting, the uniqueness rate is about 75%.

The conclusion here is that there are many similar looking templates at different depths. This is especially evident for the monocular Template Matching and the FPM matching. The Similarity Pose Association is partially affected by this. One reason here is that we

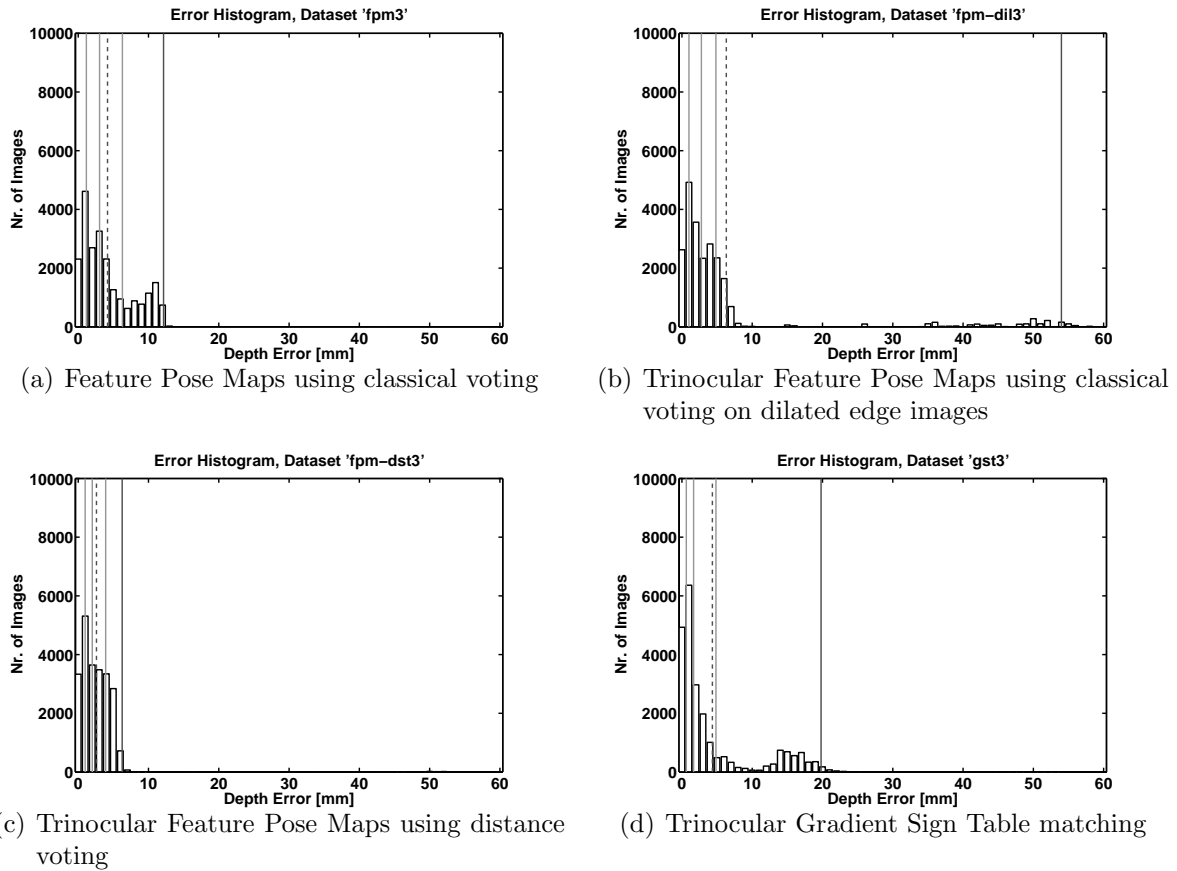


Figure 8.10: Depth error histograms of pose estimation methods. Solid lines: 25% quantile, 50% quantile (median), 75% quantile. Dashed line: mean. Dash-dotted line: 99% quantile.

used the same parameters as for the rotation experiments which might have to be set stricter when depth changes are considered. A detailed investigation of these parameters was omitted, because they are application dependent and will be determined in the set up phase after the deployment of an application. Refer to Sec. 8.3.5 for an analysis of these effects.

8.3.4 Use of Resources

The recognition methods under investigation in this thesis read the template files generated for Template Matching. In order to speed up loading the internal data structures are cached in one large file. With the exception of a header of about 100 bytes the file consists of the

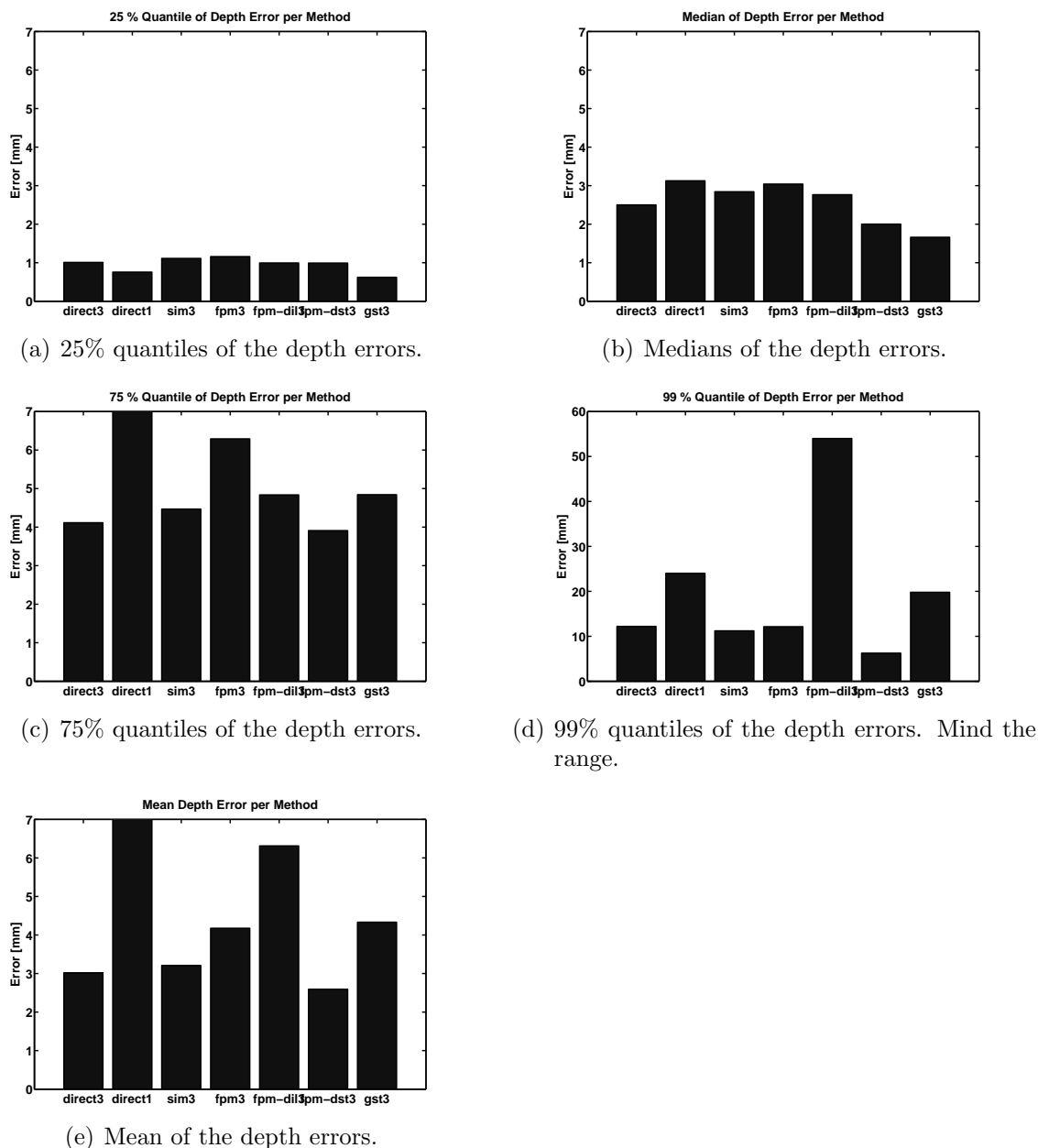


Figure 8.11: Quantiles and mean of the depth errors for different pose estimation methods. Please note the different scaling of the 99% quantile plot to include the largest value.

model information of the respective method only. Thus we compare the size of these cache files for the various methods in order to compare the relative memory requirements of the methods. Dynamic memory such as edge images etc. are not considered as — with the exception of the distance images for Template Matching and FPM with distance voting —

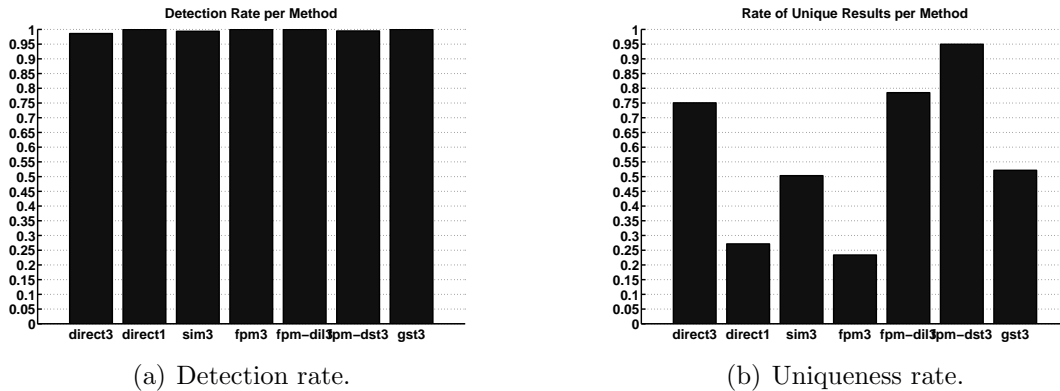


Figure 8.12: Detection and uniqueness rate for the depth estimation of different pose estimation methods. The detection rate indicates the number of images (relative to the total number) where any number of results was generated. The uniqueness rate indicates the number of images where a single result was generated.

all methods require the nearly same amount of dynamic memory.

As all methods have been optimised to about the same extent, this comparison is fair, but does not constitute a lower or upper limit on the memory consumption as there is still potential for more optimisation in addition to those already made.

Fig. 8.13 depicts the memory consumption of the rotation and distance experiments respectively. The distance accuracy (Fig. 8.13(b)) experiments required far fewer templates than the rotation experiments (Fig. 8.13(a)), resulting in the noticeable smaller memory footprint of the model data. Most interesting is that the FPM (which is used in all three FPM matching methods) requires about the same memory regardless of the number of templates.

This is caused by the image sized FPM tables per feature (edge direction). For unused pixel — which constitute the majority — at least a marker for an empty list has to be stored. In fact, our implementation stores two 32-bit values: Index into the list of tuple numbers and number of consecutive entries. A possible optimisation would be to store an RLE compressed bitmap indicating used pixels.

Another reason is the redundancy in the tuple number lists. For quite a number of pixels large portions of the lists are identical because the template overlap there. Using lists of list this could be compressed. These measures together will greatly reduce the memory consumption.

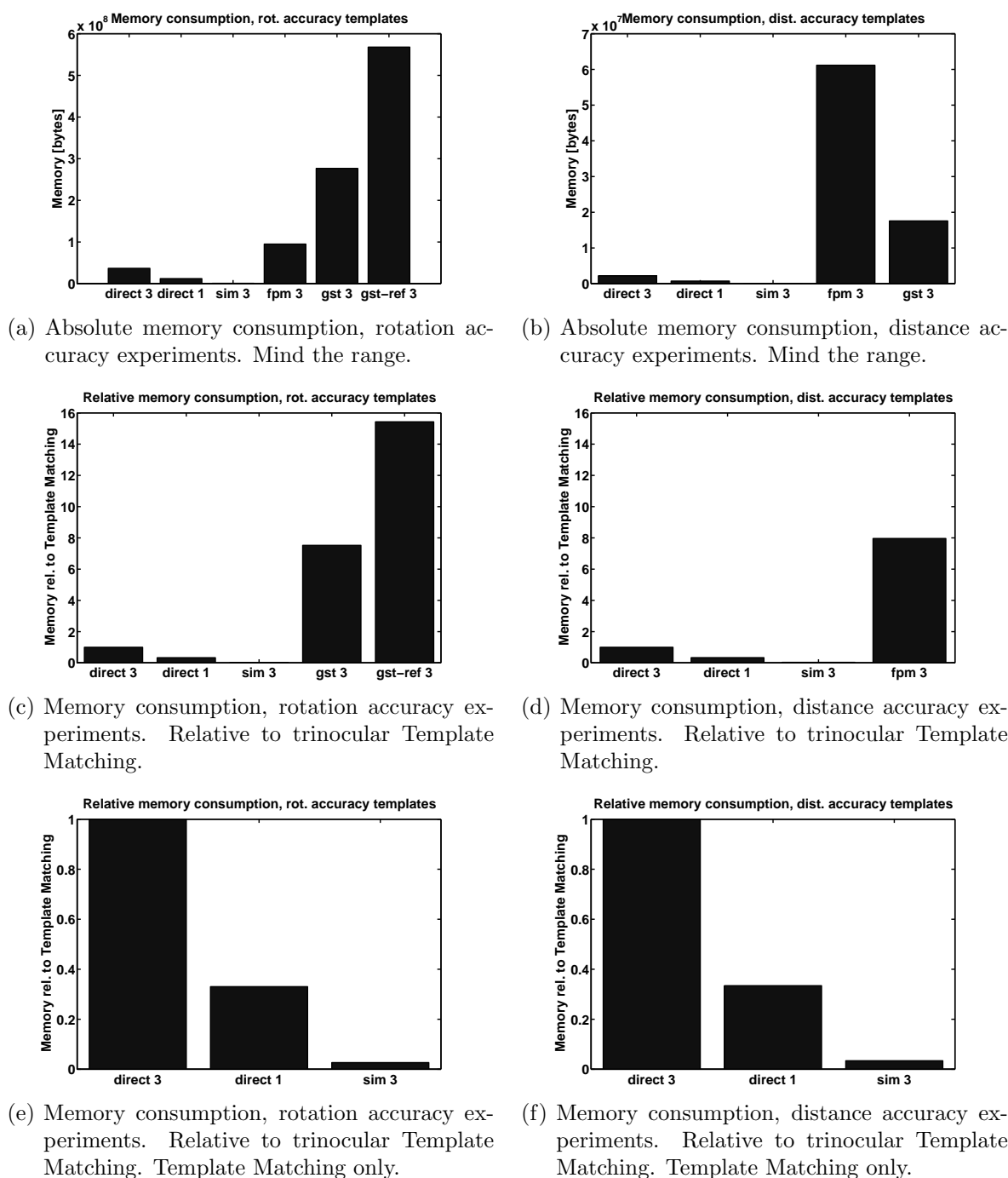


Figure 8.13: Memory consumption of various recognition methods. The size of the internal data structures forming the model information is depicted.

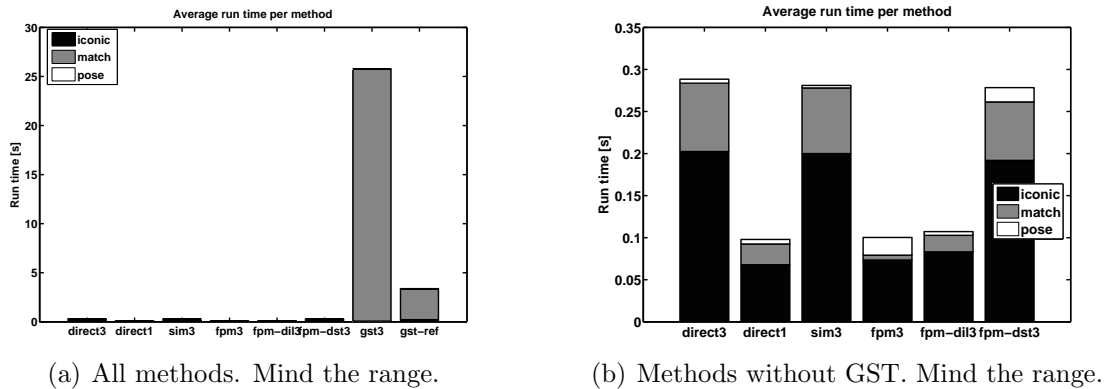


Figure 8.14: Run time of the rotation accuracy experiments. Averaged over 10 images. Mind the different ranges.

The image coordinate Gradient Sign Tables required about $8\times$ more memory than the trinocular template Matching for both experiments (Fig. 8.13(c) and 8.13(d)). The simple RLE compression scheme is obviously not sufficient and has to be replaced by a more elaborate method.

The pose Gradient Sign Tables required about $16\times$ more memory than the trinocular template Matching. Here the higher number of 5 DoF vs. 1 DoF (direction) above increases the memory consumption. The pose GST are implemented using 8 bit (6 bit length) long record per RLE strip, the image coordinate GST using 16 bit per strip (14 bit length). Although the number of DoF's is five times higher, the memory consumption only doubles. This leads to the conclusion that the 8 bit RLE strips are the more efficient compression method for this type of data.

Fig. 8.13(e) and 8.13(f) depict the relative memory consumption of the Template Matching methods. Unsurprisingly the monocular variant requires one third of the memory compared to the trinocular variant. The similarity compression on the other hand reduces the model size by an approximate factor of 30. Given the nearly identical matching results and processing times (see below) this is the most practical method.

Fig. 8.14 depicts the run time each pose estimation method required. Data was obtained using a commercial profiler. The time each program was actually executed (user time) was summed over all routines that belong to one group. The three groups with their labels are: iconic operations (edge extraction, distance transformation, dilatation, label iconic), matching (label match), and sub-lattice accurate pose computation (pose).

The pose refinement using distance-pose GST requires one order of magnitude more time than the template matching. This is caused by the multiple traversals of the list of pixels. One gradient descent step is very fast, the sheer number of steps takes up all the time. One has to keep in mind that GST was designed as a pose refinement method, but is used as a detection system here. It is very likely that other pose refinement methods, when used as detectors, suffer from similar problems.

The distance-image coordinate GST matching is another order of magnitude slower than that. This is caused by the large number of operations that are performed. Basically the same distance values are computed over and over, requiring all the time.

Fig. 8.14(b) gives the times of the non-GST methods. The pixel operations of the monocular Template Matching requires one third of the time of the trinocular variant. About two thirds of this time is spent computing the distance transforms as e.g. the times of the FPM matching shows, which does not perform the distance transform. The dilatation in contrast requires comparatively little time (fpm-dil).

The FPM methods require more time to find the local maxima in the accumulator than the Template Matching. The template matching reduces the number of potential matching templates during its operations, whereas the FPM has to sift through all poses at the end. The simple FPM matching requires more time for doing so, because the thresholds had to be lowered to get at least one result. This causes a lot of poses to pass the cheapest filter: suitable number of votes. The later filter criteria are stricter, but require more time.

The matching procedures of the FPM with classical voting require at most a quarter of the time of Template Matching. The distance voting requires a little less time than Template Matching. The latter processes a greater number of pixels, thus it requires more time.

8.3.5 Analysis of the Observations

In this section we analyse the observations above and propose further experiments to investigate the reasons for the effects. These experiments may yield an improvement in accuracy, but the methods are already accurate enough for the task at hand, thus we do not perform these experiments in this thesis.

The first point to be addressed is the smaller error of the monocular matching compared to the trinocular matching in the rotation experiments. The only difference between the two experiments is the number of cameras. Interpreting template matching as an optimisation

problem, we have changed the objective function by introducing the two other images. We sample the function at regular intervals and select the lowest local minimum.

As the histogram is stretched (seen by all quantile and the mean becoming larger) we can safely assume that the effect is not caused by an increased number of outliers. In terms of an optimisation problem it means that the additional images cause the objective function to become flatter (at least along the dimensions of the rotation parameters as the larger errors in the depth experiments show). This means that at the same amount of noise in the function values the positions of the local minima are scattered over a wider area.

The most likely explanation is that a parallax effect occurs: The edges of the object are slightly bevelled. This is considered in the 3D model, rendered and therefore part of the templates. The problem here is that the templates are only accurate descriptions of the object's appearance at the rendering poses. Objects that deviate from the grid poses are found by searching for the closest image point. This is an efficiency increasing approximation that might associate image pixels to template pixels that are actually different spatial points. The resulting averaging over the images flattens the objective function as the function values get larger faster near the local minimum than away from it. A small increase in Chamfer distances and a small decrease of the overlap (both $\sim 1\%$) can be found in the trinocular matching results. The amount of this effect could be verified by simulating various template spacings and bevel radii.

The performance of the classical FPM matching is — as expected — rather poor due to inappropriate spacing of the templates with respect to the image resolution. The development of the proposed extensions was therefore not only interesting but necessary.

The most surprising outcome of these experiments is that FPM matching on dilated edges yields the second best results concerning the rotation angles but yields such high number of outliers in depth. The purpose of the dilatation is to distribute the object contour over a range of pixels until a sufficient number overlaps with the template. This also makes the position of the object in the image rather uncertain. If the pixel position of the object becomes uncertain, its disparity becomes also uncertain. This in turn leads to uncertain depth values, the effect we observe here in form a larger number of outliers.

For the same reason the rotation angles become more accurate: The dilatation leads basically eliminates the influence of the translation on the rotation that cause the monocular matching to be better than the trinocular one. Thus the changes in the value of objective function due to object rotation are weighted higher relatively to the changes due to trans-

lation. The overall result is a better estimation of the rotation parameters at the expense of a much worse estimation of the translational parameters.

This explanation corroborates the parallax theory above: Due to the discrete translations of the template and the resulting slightly incorrect assignments of image pixels to template pixels the value of the objective function due to rotation errors competes with the value due to translation errors. Eliminating the translation dependent term by dilating the edges makes the matching more sensitive to the rotation deviations.

The newly devised FPM matching using distance voting exhibits a slightly better stability than the Template Matching for both rotation and depth estimation. This is due to the similar objective function. The fact that distance voting yields better results is most likely caused by a better interpolation. The weighting factor of distance voting (proportional to chamfer distance) is slightly better suited than that of the template matching (proportional to chamfer distance divided by overlap).

The performance differences between Template Matching and distance-image coordinate GST — both for rotation and depth estimation — can be explained by the different distance functions. Template matching computes the distance from the template pixel to the nearest image pixel, whereas GST computes the distance from an image edge pixel to the nearest template pixel. The functions have similar — but not identical — recognition capabilities.

The most relevant fact is that unassigned template pixel do not incur a penalty as it does in Template Matching: Computing the overlap parameter is rather straightforward in Template Matching, but less so in GST matching. Using the same definition of the overlap, one might achieve values greater than one. This happens if fuzzy image edges produce additional edge pixels. Since we cannot know the maximal number of image pixels (belonging to the object) we normalise by the maximal number of template pixels, which might be less. This simple procedure increases the overlap allowing more templates to pass the thresholds, leading to a smaller rate of unique results.

Finally we compare the methods above with the state of the art as given in Sec. 1.2.3 where the same object is used. There a maximum of 4° and median error of about 2° is given. Our worst case errors are 7° (trinocular Template Matching) and 5° (monocular Template Matching). The median error of both these methods is about 1° and therefore better than that of the method from the literature.

The reason for the higher rate of outliers in our method at a comparative median error level is also the aforementioned parallax effect. The method [79] from Sec. 1.2.3 can obtain

a more accurate position as it tests positions in pixel-wide distances not 2 mm distances as our methods. At a resolution of about 200 pixel per 60 mm (estimated from illustrations) is much higher than ours, the effect of compensating translational error with rotation changes is less strong in [79].

The distance-pose GST has a higher pose noise because the iteration number as a quality measure for a pose is not as distinctive as the other measures. The fact that the gradient descent stops at a certain pose does not mean the pose actually fits the model well. It simply means that this pose is a local minimum if it is reached from many starting positions.

Additionally poses at the edge of the lattice are disadvantaged because they can be reached from fewer starting poses which — due to the averaging — increases the quantisation noise as fewer values are integrated to one measure.

Together with the slightly higher error of the hierarchical evaluation of the Template Matching we conclude that the objective function (Chamfer distance) has a number of similar local minima in the vicinity of the global minimum. In order to improve the matching results and to achieve the same matching quality using the hierarchical evaluation the objective function has to be smoothed. In the simplest case this might be achieved by averaging the matching results of the neighbouring lattice positions. This will surely lead to higher computation times even if all templates are evaluated anyway.

Alternatively the spacing of the lattice might be chosen too large. Since we identified this as the cause for the strong parallax effect it is reasonable to assume it to be the reason for the larger noise of the hierarchical match. As we in fact sub-sample the already too coarse sampled objective function we increase the parallax effect even more.

Altogether this leads to the conclusion that the equidistant spacing in metric coordinates is not the optimal implementation, however convenient it might be. Future improvements should consider this and try to combine pixel accurate translations with metric lattices.

8.4 Object Classification Performance

The classification performance is evaluated by performing the localisation for both the black and the white oil cap and then deciding for the class by comparing one of various measures (e.g. chamfer distance) of the best match. Keep in mind that the edges were extracted contrast invariant, i.e. a left black, right white template edge matches also a left

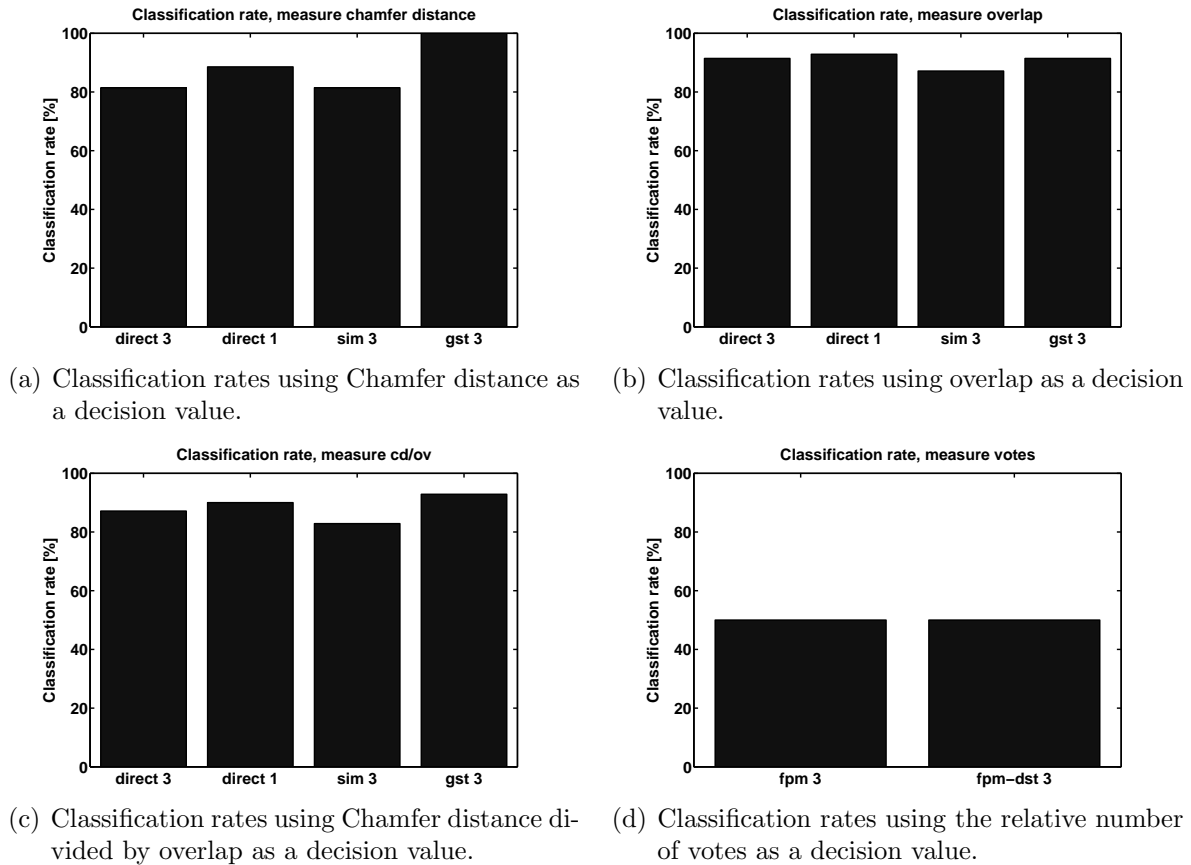


Figure 8.15: Classification performance by decision value.

white, right black image edge. We chose to so because otherwise the discrimination power regarding objects of similar grey value could not be evaluated.

We generated the templates for both objects over the same pose parameters. The ranges of the pose parameters were set up by performing pose estimations using manually entered point correspondences. This was done for the images of both objects (black and white oil cap) and the slightly different ranges were generated for both objects.

The Template Matching and GST methods were investigated using the measures chamfer distance (marked cd in the plots), overlap (marked ov), and Chamfer distance divided by the overlap (marked by cd/ov in the plots). Classification for the FPM methods was done by comparing the number of votes relative to the number of expected votes as known from the FPM generation process (marked qual).

This constitutes a Nearest-Neighbour-Classifier with rejection for each method and distance function. We do not perform elaborate computation for the decision values such as Maha-

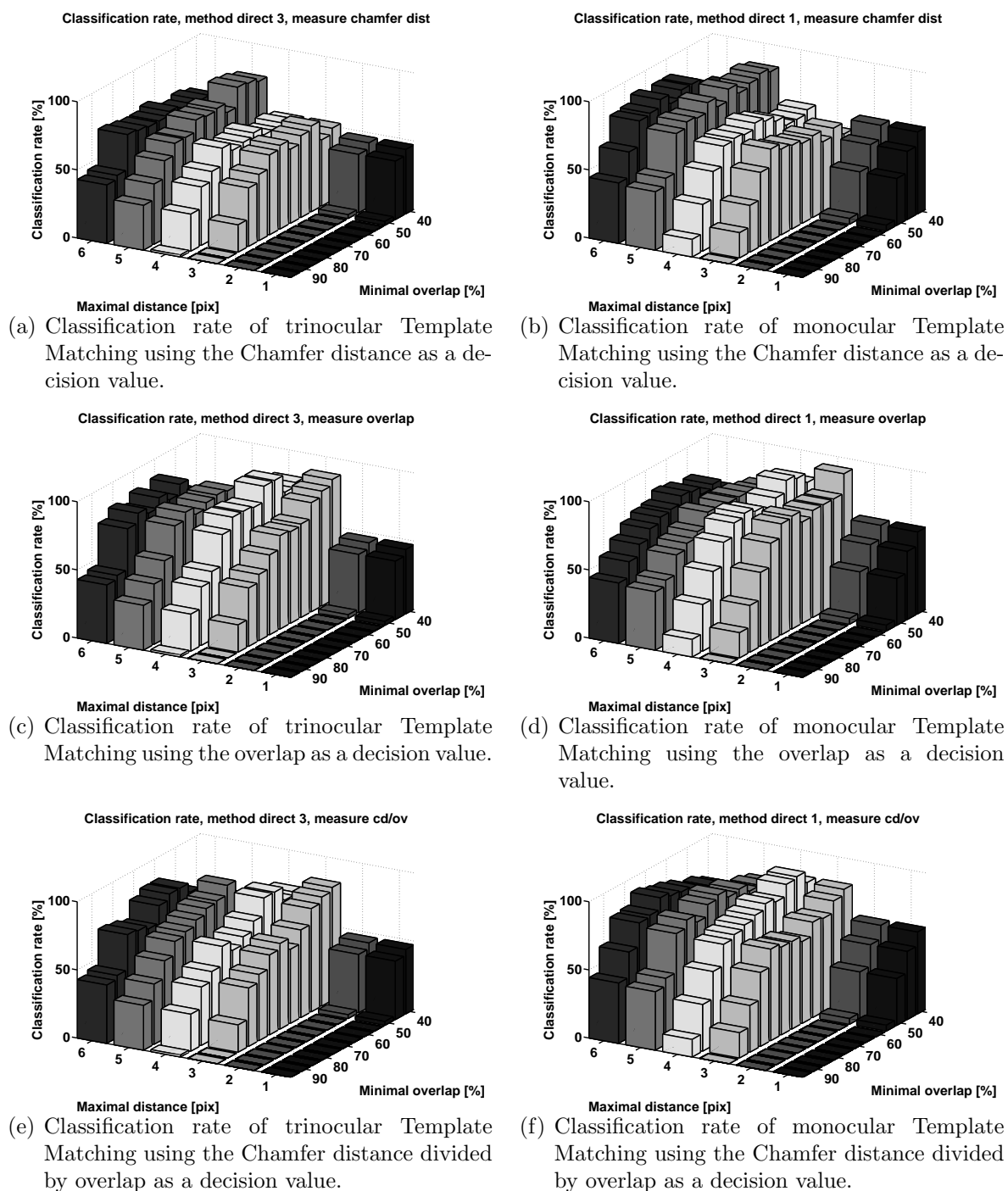


Figure 8.16: Classification performance of the trinocular and the monocular template matching using various decision values.

lanobis Distances or Confidence Mapping.

The recognition thresholds μ_l and θ_l of the Template Matching methods were varied over a range of values. For FPM the size of the structuring element (denoted by dilatation radius) and the threshold of the number of votes normalised to the expected number of votes (overlap) were varied. In a similar manner the maximal chamfer distance and overlap were varied for FPM with distance voting. The GST used the same distance function as the distance image generation for Template Matching and FPM with distance voting.

The first observation we can make is that GST with Chamfer distance as decision value achieves a classification rate of 100% (Fig. 8.15). Overlap, Chamfer distance divided by overlap, and Chamfer distance for the Template Matching methods follow closely. FPM, regardless of voting scheme yield only a 50% rate. All images of white oil caps are classified as black oil caps.

The distance-image coordinate GST matching performs an on-line distance transformation of the template. Thus the distance values is not subject to image noise, missing pixels etc. The overlap value on the other hand is subjected to all these influences. Therefore the Chamfer distance is a better decision value, as both objects are subject to these effects.

Since a good part of the template pixels of the white oil cap are shading related and therefore depend on the specific lighting conditions the white oil cap has a higher recognition rate using GST than Template Matching. GST is more robust against missing pixels in images. We assume that it is more sensitive to extraneous pixels as those increase the overlap of otherwise not matching templates, leading to false positives.

Additionally influences from outside the corridor (Sec. 6.2) are suppressed. As we have seen in the accuracy evaluation this is beneficial for simple cases — yielding a better pose — but leads to a number of outliers (the bimodal error distribution). In the light of this experiment we can conclude that the distance transformed template is more sensitive to the shape of the object, thus better at telling apart which object is present, if the Chamfer distance is used. It might be interesting to see if the pose accuracy of the method increases if the Chamfer distance is used as a weight instead of the Chamfer distance overlap ratio.

In general the monocular Template Matching had a slightly higher recognition rate than the trinocular version. This is also caused because the monocular version does not have to make sure all three images fit optimally, just equally well. As mentioned before this causes the individual template to fit a little less good in the trinocular case. This in turn can cause misclassifications in critical cases.

Additionally the peak of the classification rate is achieved at a lower Chamfer distance when classifying using the overlap than the Chamfer distance (e.g. Fig. 8.16(a) and 8.16(c)). This indicates that a Chamfer distance of three (about 1.5 pixel) is the critical difference between different objects. Smaller distances probably decrease the classification rate because some images cannot be classified due to a failure to detect the object. At higher distances the white oil cap is assigned the label “black”.

This basically means the objects are very similar, otherwise the critical difference would cover a wider range. For objects with very different appearance e.g. cube and sphere the noise related difference between images of the same class would be about the same as in this example. The shape related difference — where the e.g. sphere image-cube template starts to count template pixels as “matching”, thus where the overlap value starts to increase — is much higher.

For the same reason the best classifying Chamfer distance threshold is higher when using the Chamfer distance as the classification value, while at the same time the classification rate is lower. Only the less critical images can be classified correctly. In order to do so, greater shape differences have to be assumed.

Furthermore when classifying using the Chamfer distance divided by the overlap, the trinocular Template reaches its best classification rate at a lower chamfer distance than the monocular version. The similarity pose associate Template Matching lies in between the two. The Similarity Pose Association has a slightly worse classification rate than the Direct Pose Association. As we intentionally use similar, but not identical, templates we decrease the discrimination capabilities because the template fit slightly less good, even at the best match position.

It seems advisable to use the Chamfer distance threshold of the best classification rate as a measure for object similarity. If there is a broad band of thresholds with similar classification rates the objects can be regarded as easy to classify.

The FPM matching (Fig. 8.17(a) and 8.17(b)) — regardless of the size of the structuring element for dilatation and also regardless of the voting mechanism — classified all objects as “black”. There are different reasons depending on the voting scheme.

The classical voting scheme with dilatation evens out the shape differences. Thus both models fit both images almost equally well. The white templates then fit the images depicting the white oil cap slightly worse than the shape ambiguous black template does. Consequentially the image is classified as a black object.

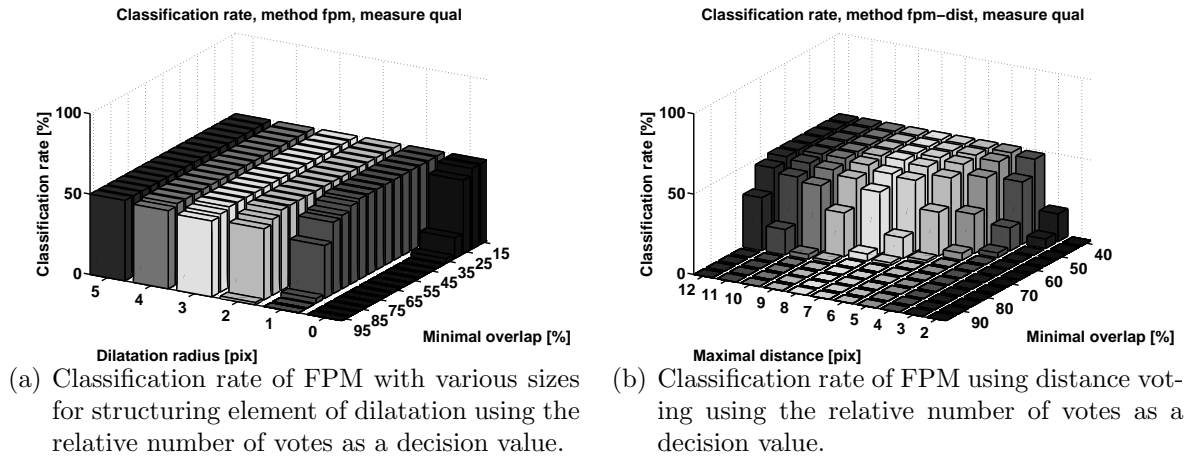


Figure 8.17: Classification performance of the FPM matching methods.

It might be advisable to introduce a provision for this case in the classification algorithm: If both classes fit the image almost equally well, the image is rejected as ambiguous. This is a different case than a rejection due to a missing object.

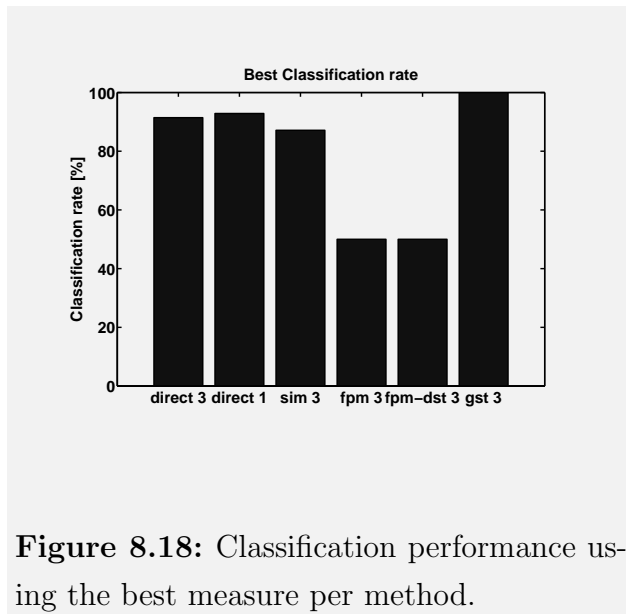


Figure 8.18: Classification performance using the best measure per method.

localisation accuracy.

Again the Similarity Pose Association is the best compromise between speed, memory consumption, and recognition rate. If memory and run-time are of no concern, the GST provides an unsurpassed recognition rate. Further development should be directed to perform the distance transform of the template off-line and store the data in a memory

We cannot search an explanation in the template rendering process as the same templates are used for Template Matching and FPM matching. Together with the results of the accuracy experiments we conclude that the bottom-up approach is less robust than the top-down matching.

The overall classification rate (Fig. 8.18) exceeds 88% for all three Template Matching procedures, which is a very good result despite the simple decision strategy. It is likely that with a finer lattice the classification rate increases together with the

conserving way. Thus the advantages of GST and Template Matching could be combined in one matching procedure.

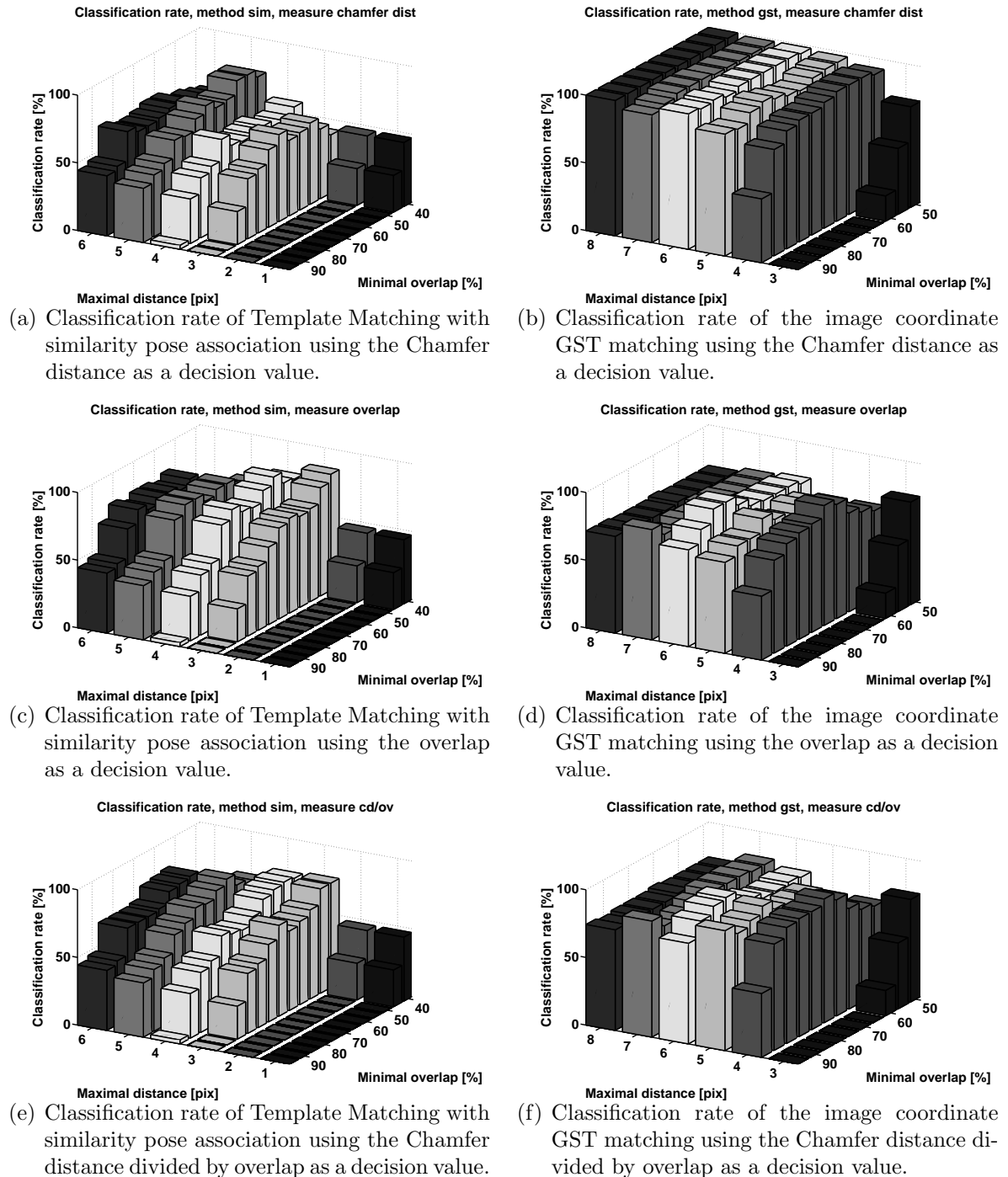


Figure 8.19: Classification performance of the Template Matching with similarity pose association and distance-image coordinate GST matching using various decision values.

8.5 Summary

The pose estimation and type classification of the rigid objects (oil caps) are characterised by the following performance numbers:

Feature	Performance
Trinocular orientation estimation	7°, 1.75 lattice steps (max. error)
Monocular orientation estimation	5°, 1.25 lattice steps (max. error)
Trinocular depth estimation (FPM, distance voting)	8 mm (99% quantile)
Trinocular depth estimation (Template Matching, similarity pose association)	12 mm (99% quantile)
Monocular depth estimation	24 mm (99% quantile)
Memory reduction by similarity pose association	30×
Trinocular Template Matching classification rate	91%
Monocular Template Matching classification rate	92%
Best classification rate	100%
Template Matching time	100 ms per image
FPM matching time	33 ms per image
Ratio preprocessing to matching time	> 2 : 1

The following non-quantitative insights have been found among other:

- Trinocular Template Matching with similarity pose association is the best compromise regarding memory consumption, run-time and recognition capabilities.
- The Chamfer distance threshold associated with the highest classification rate can be used as an indication of similarity between object classes.
- Gradient Sign Tables have a higher error (3 lattice steps vs. 2 lattice steps of the trinocular Template Matching) and require 16× more memory than trinocular Template Matching.
- Gradient Sign Tables achieve a higher recognition rate than due to distance transformed templates.
- Pose refinement methods can be used as object detectors with slightly reduced accuracy, but this constitutes not an appropriate use case due to the vastly increased run time.

9 Obtaining the Trajectory of Tubes and Cables

9.1 Goal of Investigation

Determining the 3D trajectory of flexible objects such as cables and tubes is a task of enormous importance in industrial quality inspection. The application scenario in Fig. 9.1 illustrates this.

An object (c) is to be inspected, but may be partially occluded by a cable or tube, fixed only on one side (b). This scenario occurs quite often in engine production, where cables or tubes that connect the engine with the remaining car systems are placed on top of the engine during transport from engine production to car integration. Human inspectors have to move the cable aside, perform the check, and place the cable back.

This task is very hard to automate. A suitable grasping strategy is to check if the cable passes through a volume around the inspected object and to position the gripper at the highest point within this

volume. The major obstacle is the accurate acquisition of the cable trajectory, such that a robot (a) may grasp it, hold it aside, and put it back after the camera (e) has recorded the inspection image. A further complication stems from the fact that the loose end of the cable (d) may be found at arbitrary places (“dangling rope problem”).

Due to these circumstances solutions to this problems are still subject to research. Car manufacturers have not yet deployed any computer vision systems coping with this problem.

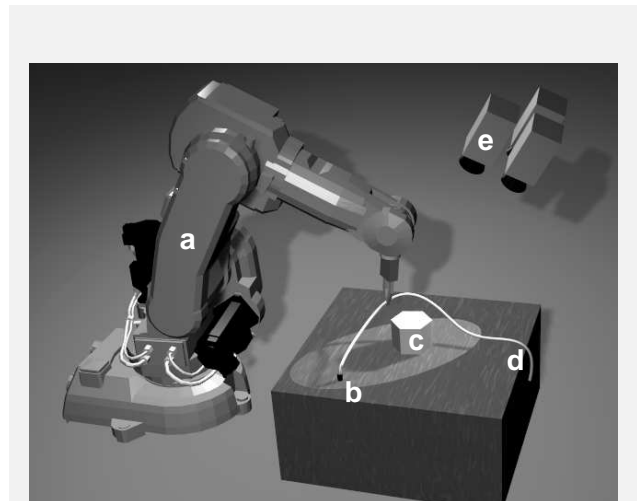


Figure 9.1: Application scenario. a: robot, b: fixed end of cable, c: inspected object, d: dangling cable end, e: inspection cameras

In this scenario we follow the tube from its fixed end — which is assumed to be at a known position and orientation — and follow it to its loose end. We do so by modelling a short piece of the tube as a straight and rigid object: the tracer. We assume that the positive x-axis of this simple model points along the tangent of the tube. The following algorithm is used to obtain the trajectory of the tube:

Algorithm 9.1 Tube tracing from fixed end to loose end.

```

Initialise pose  $\Phi$  to the fixed end
Initialise trajectory  $T_0 \leftarrow \Phi, i \leftarrow 1$ 
repeat
  Adjust pose  $\Phi$  of tracer to image tuple starting at  $T_{i-1}$ 
   $T_i \leftarrow \Phi, i \leftarrow i + 1$ 
  Move tracer along its rotated x-axis (follow tube)
until Tracer does not match or a fixed length is exceeded

```

This problem, our solution, and an initial evaluation was published in [58]. The tracer in [58] has 4 degrees of freedom: Spatial position and rotation around the line of sight. In some cases this leads to unstable behaviour of the optimisation algorithm as the gradient along the tube is almost flat and therefore very sensitive to image noise.

In this thesis we reduce the degrees of freedom to three. Given a starting position, the depth along the line of sight is one degree of freedom as well as the rotation around the line of sight. Additionally the tracer may be displaced laterally with respect to the assumed tangent after rotation. This ensures that the tracer does not move along the tube during the matching of the current tube segment.

9.2 Experimental Setup

We recorded a set of eight image tuples (Table 9.1) depicting tubes and cables with diameters between 7 and 57 mm. We processed each image tuple with the following algorithms (abbreviations for plots in first column):

Abbreviation	Algorithm
MOCCD	Multicocular Contracting Curve Density, Sec. 5.2
CCD	Monocular Contracting Curve Density, Sec. 5.2
MO snake	Multicocular Active Contour, Sec. 5.1
snake	Monocular Active Contour, Sec. 5.1
MOGST	Multicocular Gradient Sign Table, Sec. 6.2
GST	Monocular Gradient Sign Table, Sec. 6.2

These methods were used for pose adjustment according to Algorithm 9.1. Ground truth was obtained by illuminating a number of points with a laser pointer and computing the sub-pixel accurate position of the laser spot using the weighted mean of the pixel values. The spatial coordinate of the illuminated point was computed by non-linear optimisation of Eq. 3.2, i.e. Bundle Adjustment. Since this method computes points on the surface of the tube or cable and our methods above reconstruct the central curve we evaluate the methods by computing the ground truth-central curve distance reduced by the radius.



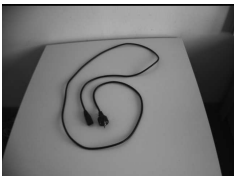
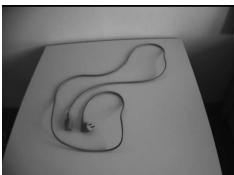


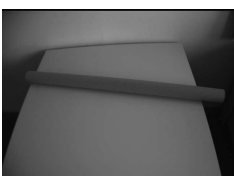

The most complicated scenes are Exp. 8 and 5. In Exp. 8 the tube is nearly invisible. A careful contrast enhancement can make it visible to a human observer. Thus sufficient information exists to distinguish between tube and table cloth. Exp. 5 exhibits at many places along the cable no contrast to the background on one side, while the other side is visible due to shading and shadow. This moves the apparent position of the cable and increases its apparent depth during monocular evaluation.

We used tracers of correct and constant radius, with a length of 1 cm, except Exp. 7 where a tracer length of 2 cm was necessary due to the large diameter of the tube. The position was updated in steps half the length of the tracer. The orientation of the last tracer position was used as the initial value for the next iteration. For the CCD we always used five perpendiculars on either side. The grey level statistics were computed for each perpendicular line separately; no merging e.g. of the inner statistics took place.

The experiments using Active Contours were based on the same model as the CCD. The tracer therefore was not flexible and can be seen as a borderline case of snakes: The deformation cost was set to infinity. In fact, we optimised the same degrees of freedom as the CCD experiments, both for comparability as well as stability.

Experiments with classical snakes (e.g. Ziplock-snakes) and modifications thereof (e.g. one-sided Ziplock-snakes) showed that these methods are too sensitive to work properly even on the comparatively simple Exp. 1 and 2. Convergence of the classic Ziplock snake depended

Table 9.1: Input data depicting tubes and cables.

Exp. Nr	Image	Description
1		Diameter: 24 mm, 15 pixel. Reconstruction of right part only. Maximal contrast.
2		Diameter: 15 mm, 8 pixel. Tube elevating vertically from table. Maximal contrast.
3		Diameter: 7 mm, 5 pixel. Black power cable. Reconstruction between connectors. Maximal contrast.
4		Diameter: 7 mm, 5 pixel. Grey power cable. Reconstruction between connectors. Reduced contrast.
5		Diameter: 7 mm, 5 pixel. White power cable. Reconstruction between connectors. Largely no contrast.
6		Diameter: 8 mm, 7 pixel. Monitor cable. Reconstruction between ballasts only. Partially no contrast on one side.
7		Diameter: 57 mm, 34 pixel. Paper tube. Poor contrast on both sides.
8		Diameter: 24 mm, 15 pixel. Tube from Exp. 1 on dark blue table cloth. Almost no contrast.

largely on the similarity between the initial and the real curvature of the tube. The one-sided Ziplock snake was very sensitive to topological defects as the direct optimisation of the central points may move it back to the already covered area of the tube.

The ensuing topological defect is turned into a kink orthogonal to the real trajectory. In order to avoid this behaviour, one has to introduce a hard constraint requesting a fixed distance and a soft constraint regarding a constant tangent of two adjacent central points along the snake. Additional constraints were required to couple rotation and lateral displacement of the boundary segment that is to be optimised. Interestingly the combination of all these constraints leads to exactly the same degrees of freedom as used for the CCD trace, combined with the tracer movement (fixed distance constraint). We therefore used the CCD tracer with identical parameters and performed a grid search on the degrees of freedom, using the image energy (blurred negative gradient image normalized to maximal gradient in the image) as the sole energy term.

All experiments of the multiocular and the monocular CCD were conducted using identical parameters. The Active Contour experiments used identical parameters where applicable. The depth change and lateral displacement were varied by \pm one radius at a resolution of one millimetre, the rotation by $\pm 30^\circ$ at a resolution of 1° . The sole exception was Exp. 7, where the lateral displacement and depth were varied by ± 5 mm in order to limit the run-time.

The Gradient Sign Tables were computed on-demand per input pixel using the 2D normals and points of the projected tracer. Since the gradient signs change massively when the aperture problem occurs — thus when the tracer runs vertical or horizontal — we disabled the processing of this image. If e.g. the tracer runs horizontal we use camera 0 and the vertical camera. We use edge pixels for features, extracted using a Sobel edge extraction with a fixed threshold. This threshold was chosen manually for each image separately such that the edge was sufficiently complete with a small number of clutter edge pixels.

We do not compare the run-times of the methods as the implementation was done in MATLAB and therefore a fair comparison — which includes all factors (e.g. CPU cache) instead of the raw number of floating point operations only — is neither possible nor meaningful.

Table 9.2: Accuracy of the various tracers. The distance (error) between estimated tube surface and ground truth is displayed in mm. Columns: Number of ground truth points passed (num), minimal error (min), average error (avg), maximal error (max).

Exp.	MOCCD				MO snake				MOGST			
	num	min	avg	max	num	min	avg	max	num	min	avg	max
1	12	0.0	1.7	4.7	11	0.3	1.0	2.3	11	0.1	1.9	7.9
2	19	0.0	1.3	4.5	19	0.2	1.3	3.2	19	0.1	1.5	5.2
3	32	0.0	2.5	22.0	32	0.2	22.8	77.5	32	0.0	3.2	23.4
4	36	0.1	3.6	19.3	36	0.0	30.0	95.3	36	0.1	17.0	54.8
5	0				28	0.0	42.8	87.4	28	0.5	18.7	44.2
6	9	0.2	40.2	129.3	30	0.1	18.6	73.2	30	0.2	17.9	84.6
7	7	0.2	2.2	5.7	7	0.0	5.1	18.9	7	1.2	5.3	14.0
8	0				0				13	2.4	13.3	36.2
	CCD				snake				GST			
1	13	0.6	17.7	65.8	12	0.3	13.7	64.3	11	0.0	4.9	28.5
2	19	0.0	51.4	121.8	19	0.1	55.6	147.1	19	0.8	23.2	73.6
3	13	0.9	78.3	164.0	32	1.0	204.5	462.1	32	0.1	34.7	141.7
4	5	0.8	25.6	57.5	36	0.1	198.2	436.4	36	0.7	173.2	718.5
5	0				28	0.1	155.9	327.0	13	7.8	185.1	461.2
6	0				30	0.9	202.6	383.9	30	0.2	119.7	484.2
7	8	2.9	59.6	129.2	7	0.7	11.2	23.0	7	1.4	9.8	16.1
8	0				0				10	0.3	127.0	296.0

9.3 Accuracy of Trajectory Estimation

Since we have ground truth for only a few (10–30) points on the tubes, we give the numerical error of the methods along with our observations of the intermediate matching results. We made those observations by displaying the re-projected tracer in the images. Thus some effects (like depth errors) are easily explained.

Table 9.2 summarises the numerical results of the experiments. The maximal, minimal, and average errors of the ground truth points are displayed per method. A ground truth point was projected onto the central axis. The distance from this projection to the measured point was recorded. Only points falling onto a tracer were considered. The computation correctly considered the topology, i.e. only correctly ordered points were compared. The number of those points is also displayed. We considered only the trajectory sections from the start to the loss of track.

Exp. 1 and 2 ran flawlessly for all six algorithms. The tracer follows the tube from the

start for the pre-determined length. The MOCCD is accurate to about 5 mm, while the monocular variant displayed the typical problems of monocular depth estimation, resulting in an at least tenfold average error (Exp. 1) over the multiocular variant.

The multiocular snake also follows the image of the tube well. The maximal spatial error of the multiocular variant is even smaller than that of the MOCCD, caused by the better fit due to the exhaustive search. The monocular snake performs slightly better in Exp. 1 and slightly worse in Exp. 2 compared to the monocular CCD. The monocular snake performs worse than the multiocular version.

The MOGST tracer achieves a slightly higher error, which is due to the edge noise. The monocular GST is the most accurate of the three monocular methods, due to the suppression of shading effects by the edge extraction. The errors are still about $10\times$ larger than those of the multiocular variant.

In Exp. 3, at one trajectory step of the MOCCD, the tracer converged such that in one image it matched the shading edge. The other images could not compensate due to the aperture problem, i.e. the tube ran horizontally at that position while the image 0 tracer was at the shading edge. The monocular tracer yielded a higher error due to the poor signal-to-noise ratio in the images: the scene contains much more distant cable sections than any other image, thus the cable is much smaller there. This lead to premature stops of the optimisation as the perpendiculars became too short (less than 3 pixel on either side) to compute a stable statistics. The monocular CCD solely computes the depth from the estimated width of the cable and yields incorrect data in such a case.

Both snake variants followed the shading edge, much like the CCD. The resulting depth error is much higher due to the fact that the snake usually found its best fit when it was turned with respect to the tangent. The multiocular snake was barely able to compensate this, resulting in an error of about 8 cm. The monocular variant could also follow the complete cable, in contrast to the monocular CCD. The depths as estimated by the snake are basically meaningless. In order to compensate the missing second edge, the ribbon snake collapsed to a single line. Since the search range is limited per iteration, the tracer increased its distance from the camera as it moved along the cable.

The GST tracers could follow the outer edge of the tube well, due to the appropriate edge extraction threshold. This threshold resulted in edge pixels along the shading edge, which had no influence on the tracer. At some positions along the cable, one outer edge could not be extracted, thus the higher maximal error and small average error. Both GST tracer

have similar errors as the CCD tracers.

Exp. 4 exhibits a strong shading edge and a reduced contrast between one side of the cable and the table. Both CCD variants converged to the shading edge. Despite this, the MOCCD can estimate the depth fairly well, mostly due to the shading being at similar position and therefore nearly correct depth. Thus, the average error increases only slightly. Compared to the MOCCD the monocular variant loses the cable after a short distance.

The multiocular snake exhibited a similar behaviour as the MOCCD, just that different sides of the snake followed the shading edge. The result is a larger error, mostly due to a wrong depth estimation. The monocular snake nearly collapsed to a single line by moving away from the camera.

Similar behaviour is observed for the GST tracers. Both follow the shading edge. Due to the edge noise and decrease in apparent cable width the signal-to-noise ratio drops farther, thus the higher error of the monocular variant. Despite this the monocular GST is more robust than the monocular CCD as it could follow the complete cable, which is due to the edge threshold.

In Exp. 5 and 6 the almost non-existent contrast on one side combined with a distinct shadow/shading edge on the other side causes both the CCD and snake algorithms to place the tracer outside the cable. This only works well on the plain background, a more structured background would cause the tracer to follow the background structures. We therefore consider these two experiments beyond the capabilities of the methods and present no data.

Exp. 5 is challenging for the snake, although both variants followed the shading edge. At one iteration, the snake projected into one image was completely off the cable. Given a more structured background this would have caused a loss of track. The monocular variant collapsed as in the previous experiments.

The multiocular GST tracer handles this image better than the multiocular snake as it follows the shading edge on the correct side. The monocular GST loses the cable at the first sharp bend due to the change of sides of the real and the shading edge.

Exp. 6 was handled by both snake variants with a behaviour similar to Exp. 4. In contrast to the CCD it could follow the complete cable. The same holds for the GST tracers.

Exp. 7 worked well with both CCD variants despite the low contrast, due to the large size of the object and the longer tracer. Again the monocular variant obtained an incorrect

depth due to an incorrect size estimation. The multiocular variant — which effectively is a binocular CCD due to the nearly horizontal tube — matched with a slightly higher error than in Exp. 1 and 2. Both methods traversed the table edge equally well. The monocular variant yielded a better depth estimation due to the higher contrast to the black background.

The snakes could also follow the tube, although with a larger error. This is caused by edge extraction and subsequent blurring using a Gaussian filter with two pixel radius as in the other experiments. As the shading of this tube is much smoother due to the larger radius, the maximum of the image energy was less well defined. This led to a larger error, mostly in depth.

The GST tracers could also follow the tube for its entire length despite the clutter edges in the shadow in front of the tube. Even missing edge pixels (about 1 cm near the table's edge) could be bridged. Again a higher robustness to smooth shading edges is demonstrated by the smaller error of the monocular GST tracer. The error of the multiocular GST is higher than that of the multiocular CCD, which is due to the clutter pixels and the edge position noise.

The dark blue table cloth in Exp. 8 combined with the black tube reduced the contrast even further than in Exp. 5 and 6. Both CCD variants diverged from the starting pose in random directions. Similar behaviour resulted from the application of the snakes.

With careful selection of the edge threshold the tube could be segmented, but a lot of clutter pixels were present. The multiocular GST could therefore follow the tube for the complete length, but at a large error. The monocular GST could follow the tube for about two thirds of its length and then loses it due to a nearly uniform distribution of clutter edge pixels.

The tracing approach is very suitable to solve the “dangling rope problem”. As mentioned above, the constraints that are required to make the snakes work properly are identical with the tracing approach. Since the CCD uses the same tracing approach, but uses a different optimisation strategy, it might be possible to replace the optimisation of snakes by the maximum-a-posterior-probability function of the CCD. To do so, the various — usually ad-hoc defined — energies have to be recast as probabilities in order to be integrated into the CCD framework. The more challenging task is the integration of hard constraints, which turn the system of equalities at the heart of the Gauss-Newton step into a system of inequalities.

The Gradient Sign Tables can also be used as an optimisation procedure for Active Contours. The reformulation of the Gradient Sign in terms of simple decisions is a viable alternative to the tabulation of those signs. Although no run-time investigation can be made using the non-optimised MATLAB implementations, the number of operations per pixel is much smaller for the GST than for e.g. snakes. Both robustness and accuracy of GST are comparable to the snakes and thus constitute a real alternative.

9.4 Summary

The pose estimation of the flexible objects (tubes and cables) are characterised by the following performance numbers:

Feature	Performance
Minimal average error, multiocular (mm)	1.0
Maximal average error, multiocular (mm)	42.0
Minimal average error, monocular (mm)	4.9
Maximal average error, monocular (mm)	204.5

The following non-quantitative insights have been found among other:

- Classical snake approaches (Ziplock) or modifications thereof (one-sided Ziplock) are not suitable to solve the “dangling rope problem”.
- The constraints to resolve topological problems of snakes yield the tracing approach.
- The tracing approach is capable of solving the “dangling rope problem”.
- Snakes are more robust with respect to missing edges or small objects, due to the blurring of the image, which operates on filters of constant size. The CCD is more accurate due to the adaptive edge filter, which imposes a minimal object size.
- The GST tracer is of comparable accuracy as CCD and snakes. The GST tracer is more robust to smooth edges, faster, and has a straightforward implementation.
- The CCD is more suitable for larger tubes with two edges. Snakes are more suitable for objects below the minimal size of the CCD or if only one edge is visible.
- The imaging conditions should be set such that both sides of the tube are visible. If not possible, an edge due to shading can be used in some cases.
- The monocular variants require two edges, otherwise the depth estimation is wrong. In that case, one has to revert to a 2D tracing of the tube.

Part IV

Closing Remarks

10 Summary

10.1 A-Priori Questions Answered

In Sec. 2.1 we laid out the questions this thesis is supposed to answer. In this section we will summarise the results of the evaluation to give the answers to these questions. We therefore repeat the questions and provide the respective answers.

Given the same object representation, which is better: a bottom-up or a top-down approach to object recognition? In Sec. 4.1 we presented Template Matching and in Sec. 6.2 we introduced a new method: Gradient Sign Tables. Both approaches are top-down methods. In Sec. 4.2 the corresponding bottom-up method — Feature Pose Maps (FPM) — was presented. Both methods have been evaluated in Chap. 8 regarding pose estimation accuracy and classification rate using identical input images and model information.

Template Matching performs better than FPM regarding memory consumption, classification rate, and pose accuracy. The FPM using the newly developed distance voting is slightly better regarding pose accuracy as it is less prone to outliers. The classical FPM methods have a smaller run time, the FPM with distance voting has nearly identical run times as Template Matching. Due to the higher memory consumption of FPM and the nearly identical other features we deem **Template Matching, the top-down method** the better matching procedure.

Is it better to perform the distance transform of the model or the current image?

We investigated this question using the newly developed Gradient Sign Table matching. This method performs an on-line distance transform of the model, saving memory. Even using Run Length Encoding the memory requirements were ten times higher, the run times one hundred times. The resulting pose accuracy was slightly worse than that of e.g. Template Matching. The classification rate was 100% for the GST vs. 91% for the Template Matching. All in all we conclude due to high run times and memory consumption the distance transform of the **image** is the better method.

Does multiocular evaluation always perform better, regardless of the object recognition method?

We investigated this for rigid objects in Chap. 8 and for tubes and cables in Chap. 9 using Template Matching, the Contracting Curve Density algorithm (Sec. 5.2), and Active Contours (Sec. 5.1). The depth information obtained by the multiocular variants was always better than that of the monocular variant, regardless of algorithm. Additionally, the multiocular variant was more robust in detecting the object. Although it is subject to accuracy reducing parallax effects on bevelled edges, we deem the **multiocular** variant better as those effect are easily compensated by better depth values and nearly identical classification rates.

Can we simplify the Closest-Feature Method, especially regarding the matching procedure and the model?

In Sec. 6.2 we presented Gradient Sign Tables, a newly developed model and matching procedure. The matching procedure provides a pose refinement algorithm that requires **one table look-up per pixel** in order to map feature presence to direction of pose improvement. The resulting algorithm has accuracy properties that are comparable to the Contracting Curve Density algorithm and Active Contours. Furthermore the approach is easily understood and straightforward to implement.

10.2 Additional Insights

In addition to the questions above we gained the following insights:

Exterior camera parameter require calibration images in the far corners of the calibration volume for an accurate and stable calibration. In Sec. 7 we evaluated the camera

calibration we use in this thesis. In addition to the rules of thumb regarding calibration rig placement as found in the literature, we could identify another rule that is important for the stability of calibration in multiocular camera systems.

Exterior camera parameters can be optimised on spatial errors. The usual non-linear minimisation of the reprojection error using Bundle Adjustment can be preceded by a minimisation of the spatial differences of the rig point coordinates. These coordinates are required for and obtained during internal calibration anyway. As this optimisation is subject to very few parameters (it does optimise the camera-to-camera transforms only) the computation is much faster than the optimisation of the complete problem. It reduces the reprojection error by about $\frac{2}{3}$.

Overexposure leads to inaccurate corner detection. Thus during calibration one should reduce the exposure time of the camera.

Images taken beyond a certain distance reduce the accuracy of the focal length estimation. For a camera of 70° field of view and VGA resolution this distance is between 120 cm and 160 cm.

All internal camera parameters are subject to local minima during the calibration. Including additional images at same location alleviates the effect.

The constraints to resolve topological problems of snakes yield the tracing approach for computing the trajectory of tubes and cables.

The CCD is more suitable for larger tubes; snakes are more suitable for smaller cables. The CCD requires a minimal size due to the grey value statistics. Snakes are limited by the Sampling Theorem only.

11 Outlook

11.1 Calibration

The calibration system presented in this thesis proved its accuracy and utility in calibrations of many different camera systems, ranging from experimental devices with 64×48 pixel over multispectral setups (one near-infrared, one thermal infrared camera) to regular industrial cameras at Megapixel resolutions.

The chequerboard detector is fast enough to be run in real time, displaying the detection result as overlay over the video stream. This does not alleviate the user of such a system to present the calibration rig in appropriate positions such that the ensuing optimisation is numerically stable and yields meaningful results. An end-user calibration system should perform a continuous supervision of the incoming rig coordinates and make suggestions on result improving rig positions.

11.2 Template Matching

The similarity pose association checks the Template with the smallest Euclidean distance in pose space for similarity. It might be interesting to investigate other distance measures and/or optimisation procedures. The problem is likely to be NP-complete, so approximations have to be made.

A reduction of the memory consumption of the individual template would be also of great practical relevance. An approach based on relative pixel offsets instead of the currently used absolute positions is worth investigating. The relative offsets from one template pixel to the next might take fewer bits than the absolute coordinates.

11.3 Feature Pose Maps

The current implementation uses an input image sized table to store the Feature Pose Map. It might be beneficial to restrict the table to the used space, e.g. by means of a bounding box. Additionally, the pose indices could be stored as pose index increments, thus using fewer bits per vote.

11.4 Gradient Sign Tables

The Gradient Sign Tables (GST) as presented here constitute an extreme: Computation is replaced by table lookups. This leads to a large memory consumption. As we have seen the matching results are almost as good as those of the methods with longer development history. Additionally one iteration of the pose refinement is very fast. It would be interesting, and surely of practical value, if the memory requirements could be reduced while adhering to the basic principle.

One way to do so would be to replace the pure table lookup with some cheap computations. Using the tube tracer as an example we will illustrate this. The GST of the tracer for the lateral displacement is particularly simple: pixels in a corridor close to the border hold the sign that will move the border closer to the pixel. Given the projection of the tracer in the image, this sign can be computed using a scalar product and some thresholds.

Given these improvements to the Gradient Sign Table matching and suitable object models a broader evaluation should be performed on a variety of pose improvement problems.

11.5 Trajectory of Tubes and Cables

Template Matching and Feature Pose Maps can be used to detect tubes in image tuples. This is done by applying the two methods using all templates not just those around the current pose along the trajectory. This is likely to reconstruct all tubes within the volume spanned by the templates. As such it opens a broader space of problems where even the starting point is unknown, e.g. bin picking of tubes and cables.

11.6 Other Issues

So far we have investigated appearance-based object recognition methods. A further interesting area of study are the model-centric object recognition methods. The method in [75] is an interesting starting point. If its object model can be made more robust and general, as well as its radius of convergence can be increased, the practical value of this method might be even greater than that of Template Matching. The reason is the faster model generation, where the CAD model has to be converted into a similar representation, which can be done without rendering.

Part V

Appendix

A Definitions

A.1 Image Tuple

Ordered set of N_c images, recorded synchronously. It is assumed that the number of cameras remains constant during different phases (cf. Sec. 3.1) of the application, unless stated otherwise. If the scene remains unchanged, it is possible to move the camera and regard these sequentially recorded images as a tuple too. Most algorithms allow to skip cameras during the recognition phase, suffering only from recognition stability so as if the camera has not been there in the first place.

A.2 Object Recognition Algorithms

2 dimensional object recognition algorithms operate on single grey or colour images only. All extracted degrees of freedom are limited to the image plane.

3 dimensional object recognition algorithms operate on 3D data such as range data (e.g. from RADAR or LIDAR) or voxel data (e.g. from tomography). The extracted degrees of freedom include the distance between object and sensor.

2.5 dimensional object recognition algorithms operate on single images. The extracted degrees of freedom are confined in 3D space. Only those degrees of freedom that produce appearance changes in the image can be extracted.

The term *2.5 dimensional* is borrowed from machining, where a 2.5 dimensional object denotes objects that can be machined out of a solid block of material from the top. Only

material from the top of the block can be removed, thus caverns in the object cannot be fabricated. This metaphor lends itself very well to object recognition: Only the foremost surface of the objects is visible in an image.

In this sense range images are 2.5 dimensional too. We, however, count them as 3D data as they can be transferred directly to three dimensional, even metric, data from a single sensor only. This allows the direct comparison with the 3D model data, a property that is not present in 2.5 dimensional algorithms.

A.3 Camera Identifier

Unique integer identifying one physical camera. The first camera has identifier 0. It is application dependent how the mapping of camera identifier to real camera devices is implemented.

A.4 Mesh, Facet, Vertex

As in generative computer graphics the object models are called meshes. Each mesh consists of facets (also called faces), the planar polygons. Each facet consists of three or more vertices, the corner points of the polygon. It is only a matter of computational performance whether the facet consists of a list of the point coordinates itself or of a list of indices into a larger list of the point coordinates. The first version is easier to read and therefore used in the algorithms descriptions, the second version is faster and therefore used in the implementations.

A.5 Appearance Based Object Recognition

Object recognition method that is purely based on the image of the object. Usually one image (appearance) per pose of the object is stored in a database to be recognised. Template matching and image classification fall into this category.

A.6 Segmentation Based Object Recognition Algorithm

Object recognition based on segmenting the object from the background. Depending on the actual algorithm it may or may not require a starting pose to be refined. A simple approach consists of binarising the object by a fixed treshold and computing the pose from

the outline of the object. The assumed object type is then verified from the similarity between model and outline. Active Contours (snakes) are segmentation algorithms.

A.7 Constraint, Soft Constraint, Hard Constraint

Requirement imposed on parameters during optimisation procedures. Soft constraints are meant to shape the objective function such that the desired solution becomes the closest local or even global solution and may be violated if the objective function requires it. Hard constraints on the other hand must not be violated. They for instance denote physical impossibilities, prior knowledge or other limits of the parameter space.

A.8 Distance Transform

The distance transform assigns each image pixel the distance to the closest feature pixel from the input image. Depending on the distance metric, which usually approximates the Eukclidean distance, the distance transform can be computed with a run-time complexity of $O(n)$ for n pixel. This is done by using the transitivity property of the minimum relation. Fig. A.1 illustrates the feature image and its distance transform.

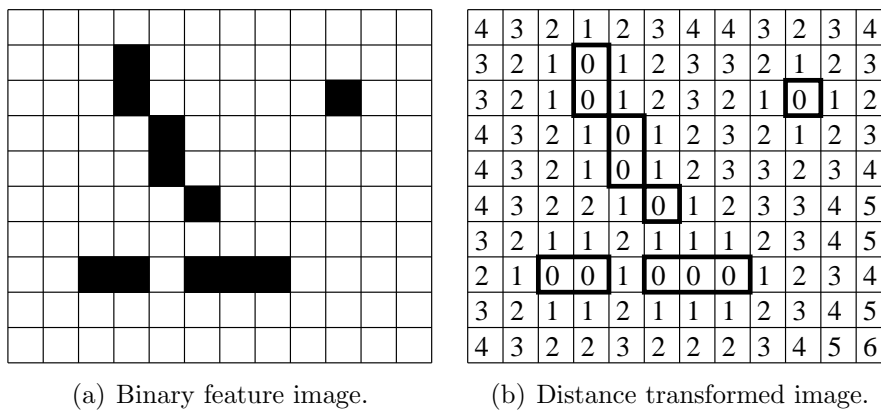


Figure A.1: Distance transform of a binary feature image.

A.9 Epipolar Line

If one camera of a multiocular setup observes a point, the epipolar line allows for a simple selection of possible matches in a second camera. The point in space and the two pinholes of the cameras form a plane in space. Projecting this plane into the second camera yields

the epipolar line. It may be computed from the external calibration and the 2D coordinates of the point in the first camera alone. Only points on (or close to, in the presence of noise) the epipolar line are candidates that have to be matched with the point.

A.10 Run Length Encoding

Replace a sequence of consecutive identical symbols by a tuple of the symbol and the number of symbols involved: AAAABBB is stored as ((A,4),(B,3)). If the number of symbols is small (e.g. 4), the symbol and the length can be stored in a byte together.

Bibliography

- [1] AICON 3D Systems GmbH. Homepage (<http://www.aicon.de>). URL <http://www.aicon.de>.
- [2] H. Araujo, R. Carceroni, and C. Brown. A fully projective formulation to improve the accuracy of lowe's pose-estimation algorithm, 1998. URL citeseer.ist.psu.edu/araujo98fully.html.
- [3] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking, 2002. URL citeseer.ist.psu.edu/maskell01tutorial.html.
- [4] P. Aschwanden. *Experimenteller Vergleich von Korrelationskriterien in der Bildanalyse*. Hartung-Gorre Verlag Konstanz, 1993.
- [5] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Upper Saddle River, NJ 07458, USA, 1982. ISBN 0-13-165316-4.
- [6] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992. ISSN 0162-8828. doi: <http://dx.doi.org/10.1109/34.121791>.
- [7] J. R. Beveridge. *Local Search Algorithms for Geometric Object Recognition: Optimal Correspondence and Pose*. PhD thesis, University of Massachusetts at Amherst, 1993.
- [8] J. R. Beveridge and E. M. Riseman. How easy is matching 2d line models using local search? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6): 564–579, 1997. URL citeseer.ist.psu.edu/beveridge97how.html.
- [9] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, London, 1998. URL <http://www.robots.ox.ac.uk/~contours/>.
- [10] G. Blasko and P. Fua. Real-time 3d object recognition for automatic tracker initialization, 2001. URL citeseer.ist.psu.edu/blasko01realtime.html.

-
- [11] J.-Y. Bouguet. Camera calibration toolbox for matlab (http://www.vision.caltech.edu/bouguetj/calib_doc/).
- [12] P. Brigger, J. Hoeg, and M. Unser. B-Spline Snakes: A Flexible Tool for Parametric Contour Detection. *IEEE Transactions on Image Processing*, 9(9):1484–1496, September 2000.
- [13] D. C. Brown. Decentring distortions of lenses. *Photogrammetric Engineering*, 32(4):444–462, 1966.
- [14] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical Report 124, 1994. URL citeseer.ist.psu.edu/76182.html.
- [15] R. Cabido, A. S. Montemayor, and Á. Sánchez. Hardware-accelerated template matching. In J. S. Marques, N. P. de la Blanca, and P. Pina, editors, *IbPRIA (1)*, volume 3522 of *Lecture Notes in Computer Science*, pages 691–698. Springer, 2005. ISBN 3-540-26153-2.
- [16] R. Campbell and P. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210, 2001.
- [17] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *ICCV*, pages 694–699, 1995.
- [18] S. Chaudhuri and A. N. Rajagopalan. *Depth from Defocus, A Real Aperture Imaging Approach*. Springer, 1999.
- [19] L. D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 53(2):211–218, 1991. URL citeseer.nj.nec.com/cohen91active.html.
- [20] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Lecture Notes in Computer Science*, 1407:484–??, 1998. URL citeseer.ist.psu.edu/cootes98active.html.
- [21] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. ISBN 0201095289.
- [22] P. d’Angelo. Estimation of spatial contours using multi-view snakes. Master’s thesis, University of Applied Sciences, Ulm, Germany, Dec 2003.

- [23] P. d'Angelo, C. Wöhler, and L. Krüger. Model based multi-view active contours for quality inspection. In *Int. Conf. on Computer Vision and Graphics*, Warsaw, Poland, 2004.
- [24] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972. ISSN 0001-0782. doi: <http://doi.acm.org/10.1145/361237.361242>.
- [25] M. Ellenrieder. *Optimal Viewpoint Selection for Industrial Machine Vision and Inspection of Flexible Objects*. Dissertation, Universität Bielefeld, Technische Fakultät, 2005.
- [26] M. M. Ellenrieder, L. Krüger, D. Stöbel, and M. Hanheide. A versatile model-based visibility measure for geometric primitives. In H. Kälviäinen, J. Parkkinen, and A. Kaarna, editors, *SCIA*, volume 3540 of *Lecture Notes in Computer Science*, pages 669–678. Springer, 2005. ISBN 3-540-26320-9.
- [27] C. Y. et. al. Persistence of vision raytracer (<http://www.povray.org>). URL <http://www.povray.org>.
- [28] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.
- [29] R. A. Finkel and J. L. Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Inf.*, 4:1–9, 1974.
- [30] A. Fitzgibbon. Robust registration of 2d and 3d point sets, 2001. URL citeseer.ist.psu.edu/fitzgibbon01robust.html.
- [31] J. G. Fryer and D. C. Brown. Lens distortion for close-range photogrammetry. *Photogrammetric Engineering and Remote Sensing*, 52(1):51–58, 1986.
- [32] P. Fua and Y. G. Leclerc. Model driven edge detection. *Machine Vision and Applications*, 3(1):45–56, 1990.
- [33] H. Fuchs, Z. M. Kedem, and B. F. Naylor. On visible surface generation by a priori tree structures. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 124–133, New York, NY, USA, 1980. ACM Press. ISBN 0-89791-021-4. doi: <http://doi.acm.org/10.1145/800250.807481>.

-
- [34] J. Gause, P. Cheung, and W. Luk. Reconfigurable shapeadaptive template matching architectures, 2002. URL citeseer.ist.psu.edu/gause02reconfigurable.html.
- [35] D. M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *ICCV99*, pages 87–93, 1999.
- [36] A. Habib, Y. Lee, and M. Morgan. Automatic matching and three-dimensional reconstruction of free-form linear features from stereo images. *Journal of Photogrammetric Engineering and Remote Sensing*, pages 189–197, 2003.
- [37] R. Hanek. The Contracting Curve Density Algorithm and its Application to Model-based Image Segmentation. In *IEEE Conf. Computer Vision and Pattern Recognition*, Kauai, Hawaii, USA, pages I:797–804, 2001. URL <http://www9.in.tum.de/papers/2001/CVPR-2001-Hanek.abstract.html>.
- [38] R. Hanek. *Fitting parametric curve models to images using local self-adapting separation criteria*. PhD thesis, Department of Informatics, Technische Universitt Mnchen, 2003.
- [39] R. Hanek and M. Beetz. The contracting curve density algorithm: Fitting parametric curve models to images using local self-adapting separation criteria. *Int. J. Comput. Vision*, 59(3):233–258, 2004. ISSN 0920-5691. doi: <http://dx.doi.org/10.1023/B:VISI.0000025799.44214.29>.
- [40] R. Hanek, T. Schmitt, S. Buck, and M. Beetz. Fast Image-based Object Localization in Natural Scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2002*, Lausanne, pages 116–122, 2002. URL citeseer.csail.mit.edu/hanek02fast.html.
- [41] R. M. Haralick, H. Joo, C. Lee, X. Zhuang, V. G. Vaidya, and M. B. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426–1446, 1989.
- [42] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *4th ALVEY Vision Conference*, pages 147–151, 1988.
- [43] J. Heikkilä and O. Silvén. A four-step camera calibration procedure with implicit image correction. In *Proceedings of the IEEE Conference on Computer Vision and*

- Pattern Recognition*, pages 1106–1112. Institute of Electrical and Electronics Engineers, 1997.
- [44] S. Hezel, A. Kugel, R. Manner, and D. M. Gavrilu. Fpga-based template matching using distance transforms. *fccm*, 00:89, 2002. ISSN 1082-3409. doi: <http://doi.ieeeecomputersociety.org/10.1109/FPGA.2002.1106664>.
- [45] M. Hinz, K. D. Toennies, M. Grohmann, and R. Pohle. Active double-contour for segmentation of vessels in digital subtraction angiography. In *Proceedings of SPIE*, volume 4322, pages 1554–1562, 2001. URL citeseer.nj.nec.com/558693.html.
- [46] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation of nonorthogonal problems. *Technometrics*, 12:55–67, 1970.
- [47] D. Huffman. A method for the construction of minimum-redundancy codes. 40(9): 1098–1101, September 1952.
- [48] D. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5.2:195–212, 1990.
- [49] Intel. Open computer vision library (<http://www.intel.com>). URL <http://www.intel.com>.
- [50] ISRA Vision Systems AG. Homepage (<http://www.isravision.com>). URL <http://www.isravision.com>.
- [51] B. Jähne. *Digital Image Processing*. Springer-Verlag, Berlin, 5 edition, June 2001.
- [52] N. Kämpchen. Modellbasierte lagebestimmung von objekten in stereobildsequenzen. Master’s thesis, University of Stuttgart, Stuttgart, Germany, 2001.
- [53] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [54] T. Koelzow and L. Krüger. Matching of a 3d model into a 2d image using a hypothesize-and-test alignment method. volume 4791, pages 222–232. SPIE, 2002. URL <http://link.aip.org/link/?PSI/4791/222/1>.
- [55] T. Kölzow. *Wissensbasierte Entwicklungsumgebung fr Bildanalyzesysteme aus dem industriellen Bereich*. Dissertation, Universität Bielefeld, Technische Fakultät, 2003.

- [56] A. Koschan. What is new in computational stereo since 1989: A survey on current stereo papers, 1993. URL <http://iristown.engr.utk.edu/~koschan/paper/Stereo-Report2.1.pdf>.
- [57] M. Krauss. Integration des depth-from-defocus-prinzips in einen pose-estimation-algorithmus. Master's thesis, Technical University Ilmenau, Ilmenau, Germany, jun 2006.
- [58] L. Krueger and M. M. Ellenrieder. Pose estimation using the multiocular extended contracting curve density algorithm. In G. Greiner, J. Hornegger, H. Niemann, and M. Stamminger, editors, *Vision, Modeling, and Visualization.*, pages 41–48, Nov. 2005.
- [59] L. Krüger, C. Wöhler, A. Würz-Wessel, and F. Stein. In-factory calibration of multiocular camera systems. In *SPIE Photonics Europe*, pages pp. 126–137, 2004.
- [60] B. Lamiroy and P. Gros. Rapid object indexing and recognition using enhanced geometric hashing. In *ECCV (1)*, pages 59–70, 1996. URL citeseer.ist.psu.edu/lamiroy96rapid.html.
- [61] A. Leow, M.-C. Chiang, H. Protas, P. M. Thompson, L. A. Vese, and H. S. C. Huang. Linear and non-linear geometric object matching with implicit representation. In *ICPR*, pages 710–713, 2004.
- [62] C. Loader. Local search algorithms for 2d geometric object recognition. Master's thesis, University of Western Australia, 1995.
- [63] D. G. Lowe. Fitting parametrized three-dimensional models to images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 441–450, 1991.
- [64] T. Luhmann, S. Robson, C. Reeves, P. Wainwright, and S. Kyle. *Close Range Photogrammetry: Principles, Methods and Applications*. Whittles Publishing, 2006.
- [65] K. Madsen, H. B. Nielsen, and O. Tingleff. Methods for non-linear least squares problems, jul 1999. URL <http://www2.imm.dtu.dk/pubdb/p.php?660>.
- [66] S. Mason. *Expert system based design of photogrammetric networks*. PhD thesis, ETH Zurich, 1994.
- [67] W. Matusik. Image-based visual hulls. In *Master of Science Thesis*, 2001. URL citeseer.ist.psu.edu/matusik01imagebased.html.

- [68] W. Neuenschwander, P. Fua, L. Iverson, G. Szekely, and O. Kubler. Ziplock Snakes. In *International Journal of Computer Vision*, 25(3):191–201, December 1997. URL citeseer.nj.nec.com/neuenschwander97ziplock.html.
- [69] Y. Ohtake, A. Belyaev, and H.-P. Seidel. A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions. In *SMI '03: Proc. Shape Modeling Int. 2003*, page 292, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1909-1.
- [70] N. Pessel, J. Opderbecke, and M.-J. Aldon. Camera self-calibration in underwater environment. In *WSCG*, 2003.
- [71] M. Reinhold, M. Grzegorzec, J. Denzler, and H. Niemann. Appearance-Based Recognition of 3-D Objects by Cluttered Background and Occlusions. *Pattern Recognition*, 38(5):739–753, 2005.
- [72] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, 2001. URL citeseer.ist.psu.edu/article/scharstein01taxonomy.html.
- [73] N. Stolte and A. Kaufman. Parallel spatial enumeration of implicit surfaces using interval arithmetic for octree generation and its direct visualization. *Implicit Surfaces'98*, pages 81–88, 1998.
- [74] D. Stöbel, M. Hanheide, G. Sagerer, L. Krüger, and M. M. Ellenrieder. Feature and viewpoint selection for industrial car assembly. In C. E. Rasmussen, H. H. Bühlhoff, B. Schölkopf, and M. A. Giese, editors, *DAGM-Symposium*, volume 3175 of *Lecture Notes in Computer Science*, pages 528–535. Springer, 2004. ISBN 3-540-22945-0.
- [75] S. Sullivan, L. Sandford, and J. Ponce. Using geometric distance fits for 3-d object modeling and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(12):1183–1196, 1994. ISSN 0162-8828. doi: <http://dx.doi.org/10.1109/34.387489>.
- [76] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372, London, UK, 2000. Springer-Verlag. ISBN 3-540-67973-1.

- [77] R. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. *CVPR*, 86:364–374, 86.
- [78] H. van Dijck. *Object Recognition with Stereo Vision and Geometric Hashing*. PhD thesis, Universiteit Twente, 1999.
- [79] C. von Bank, D. Gavrilu, and C. Wöhler. A visual quality inspection system based on a hierarchical 3d pose estimation algorithm. In *DAGM-Symposium*, pages 179–186, 2003.
- [80] M. Watanabe and S. K. Nayar. Telecentric optics for computational vision. In *ECCV (2)*, pages 439–451, 1996. URL citeseer.ist.psu.edu/watanabe97telecentric.html.
- [81] W. Wells. Statistical approaches to feature-based object recognition, 1997. URL citeseer.nj.nec.com/article/wells97statistical.html.
- [82] M. Westling and L. Davis. Object recognition by fast hypothesis generation and reasoning about object interactions. In *International Conference on Pattern Recognition*, pages IV: 148–153, 1996.
- [83] D. J. Williams and M. Shah. A Fast Algorithm for Active Contours and Curvature Estimation. *Image Understanding*, 55(1):14–26, 1992.
- [84] H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science & Engineering*, 4(4):10–21, /1997. URL citeseer.ist.psu.edu/wolfson97geometric.html.
- [85] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000. URL citeseer.ist.psu.edu/zhang98flexible.html.
- [86] Z. Zhang. Camera calibration with one-dimensional objects, 2002. URL citeseer.ist.psu.edu/zhang02camera.html.
- [87] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *IEEE International Conference on Computer Vision*, pages 666–673. Institute of Electrical and Electronics Engineers, 1999.