INAUGURAL — DISSERTATION

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht – Karls – Universität
Heidelberg

vorgelegt von
Dipl.-Ing. Andreas Haja
aus Hannover

Tag der mündlichen Prüfung: 18.12.2008

Graph-based Spatial Motion Tracking Using Affine-covariant Regions

Gutachter: Prof. Dr. Bernd Jähne
Prof. Dr. Dr. h.c. Hans Georg Bock

Abstract

This thesis considers the task of spatial motion reconstruction from image sequences using a stereoscopic camera setup. In a variety of fields, such as flow analysis in physics or the measurement of oscillation characteristics and damping behavior in mechanical engineering, efficient and accurate methods for motion analysis are of great importance.

This work discusses each algorithmic step of the motion reconstruction problem using a set of freely available image sequences. The presented concepts and evaluation results are of a generic nature and may thus be applied to a multitude of applications in various fields, where motion can be observed by two calibrated cameras.

The first step in the processing chain of a motion reconstruction algorithm is concerned with the automated detection of salient locations (=*features* or *regions*) within each image of a given sequence. In this thesis, detection is directly performed on the natural texture of the observed objects instead of using artificial marker elements (as with many currently available methods). As one of the major contributions of this work, five well-known detection methods from the contemporary literature are compared to each other with regard to several performance measures, such as localization accuracy or the robustness under perspective distortions. The given results extend the available literature on the topic and facilitate the well-founded selection of appropriate detectors according to the requirements of specific target applications.

In the second step, both *spatial* and *temporal* correspondences have to be established between features extracted from different images. With the former, two images taken at the same time instant but with different cameras are considered (stereo reconstruction) while with the latter, correspondences are sought between temporally adjacent images from the same camera instead (monocular feature tracking). With most classical methods, an observed object is either spatially reconstructed at a single time instant yielding a set of three-dimensional coordinates, *or* its motion is analyzed separately within each camera yielding a set of two-dimensional trajectories.

A major contribution of this thesis is a concept for the unification of both stereo reconstruction and monocular tracking. Based on sets of two-dimensional trajectories from each camera of a stereo setup, the proposed method uses a graph-based approach to find correspondences not between single features but between entire trajectories instead. Thereby, the influence of locally ambiguous correspondences is mitigated significantly. The resulting spatial trajectories contain both the three-dimensional structure *and* the motion of the observed objects at the same time. To the best knowledge of the author, a similar concept does not yet exist in the literature. In a detailed evaluation, the superiority of the new method is demonstrated.

Zusammenfassung

Die vorliegende Arbeit behandelt das Problem der räumlichen Bewegungsrekonstruktion aus Bildsequenzen unter Verwendung eines stereoskopischen Kameraaufbaus. Die zuverlässige und genaue Bestimmung von Bewegungsparametern spielt eine bedeutende Rolle in einer Vielzahl von Anwendungsgebieten, z.B. der Analyse von Strömungsfeldern in der Physik oder der Messung von Schwingungscharakteristiken und des Dämpfungsverhaltens im Maschinenbau.

Im Rahmen dieser Arbeit wird jeder Verarbeitungsschritt des Rekonstruktionsproblems anhand von frei verfügbaren Bildsequenzen diskutiert. Die vorgestellten Konzepte und die Untersuchungsergebnisse sind allgemeiner Natur und können daher auf eine Vielzahl von Anwendungsfällen übertragen werden, in denen die Beobachtung von Bewegung mittels zweier kalibrierter Kameras möglich ist.

Der erste Schritt in der vorgestellten Verarbeitungskette befasst sich mit der automatischen Detektion geeigneter Merkmale (oder Regionen) in jedem Einzelbild einer Bildsequenz. Im Rahmen dieser Arbeit erfolgt dieser Schritt direkt auf Basis der Eigentextur der beobachteten Objekte, d.h. es werden keine künstlichen Messmarken oder sonstige Markierungselemente verwendet. Als ein wesentlicher Beitrag dieser Arbeit werden fünf populäre Detektionsmethoden aus der Literatur hinsichtlich verschiedener Leistungskriterien miteinander verglichen. Diese beinhalten z.B. die Positionsgenauigkeit der detektierten Merkmale und deren Robustheit gegenüber perspektivischen Verzerrungen des Bildinhaltes. Die umfangreichen Untersuchungsergebnisse ergänzen die vorhandene Literatur zum Thema und ermöglichen die wohlbegründete Auswahl eines geeigneten Detektionsverfahrens anhand der Erfordernisse einer Zielapplikation.

Im zweiten Schritt werden sowohl *räumliche* als auch *zeitliche* Korrespondenzen zwischen Merkmalen aus verschiedenen Bildern extrahiert. Erstere werden aus Bilddaten gewonnen, die zum gleichen Zeitpunkt von unterschiedlichen Kameras erzeugt wurden (Stereorekonstruktion). Letztere hingegen stammen aus zeitlich benachbarten Bildern der gleichen Bildsequenz, d.h. die Aufnahme erfolgt unter Verwendung einer einzelnen Kamera (monokulare Merkmalsverfolgung). Die meisten klassischen Methoden befassen sich entweder mit der dreidimensionalen Rekonstruktion eines Objektes zu einem Zeitpunkt *oder* mit der Analyse dessen zweidimensionaler Bewegung.

Ein weiterer Beitrag dieser Arbeit besteht in einem Konzept zur Vereinigung von Stereorekonstruktion und monokularer Merkmalsverfolgung. Dieses beinhaltet im Kern einen graphenbasierten Ansatz zur Korrespondenzanalyse, der anstelle von Einzelmerkmalen aus zwei Bildern zweidimensionale Merkmalstrajektorien aus mehreren Bildern als Datenbasis verwendet. Hierdurch wird der Einfluss von Mehrdeutigkeiten deutlich gesenkt. Ergebnisse dieses Verarbeitungsschrittes sind sowohl die räumliche Struktur des beobachteten Objektes als auch dessen Bewegung. Nach Kenntnis des Autors existiert in der Literatur derzeit kein vergleichbares Verfahren. Die Leistungsfähigkeit der neuen Methode wird anhand von detaillierten Untersuchungen demonstriert.

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die zum Gelingen dieser Arbeit beigetragen haben. Mein Dank für die fachliche Betreuung der Arbeit gilt Prof. Bernd Jähne von der Universität Heidelberg und Dr. Steffen Abraham, die mich mit zahlreichen Anregungen und Ideen nach Kräften unterstützt haben. Weiterhin danke ich den Hildesheimer Doktoranden und Kollegen der Robert Bosch GmbH für die Zeit, die ich mit vielen Fragen und Diskussionen beanspruchen durfte.

Meiner Frau Petra danke ich für ihre Geduld, ihr ausdauerndes Verständnis und ihre emotionale Unterstützung, die mich durch die Verfassung dieser Schrift getragen haben. Auch meiner Familie sei an dieser Stelle mein besonderer Dank ausgesprochen.

Nicht zuletzt muss hier mein Opa Kurt Schilling erwähnt werden, der mit seiner unermüdlichen und geduldigen Mathe-Nachhilfe das Entstehen dieser Arbeit überhaupt erst ermöglicht hat. Tausend Dank!

Contents

List of Symbols and Terminology

General Symbols

I_i	gray-scale image at time index i
$\mathbf{x}$	two-dimensional position in an image I with $\mathbf{x} = [x, y]^T$
$\mu(\mathbf{x})$	second-order moment matrix at image position $\mathbf{x}$
$L_x(\mathbf{x}), L_y(\mathbf{x})$	local derivatives in the direction of x and y at position $\mathbf{x}$ in I
$g(\sigma)$	Gaussian convolution kernel with standard deviation σ
$H_{i,i+1}$	image-to-image homography
$\mathbf{P}_i^u$	3D-point u at time index i with $u \in U$, where U is the total no. of points
Λ	covariance matrix
λ	eigenvalue of a matrix

Camera-related Symbols

R	rotation matrix in three-dimensional euclidean space
U	pinhole camera intrinsic calibration matrix
$\mathbf{x_0}$	camera principal point $\mathbf{x_0} = [x_0, y_0]$
k_1, k_2, p_1, p_2	camera radial and tangential distortion parameters
f_c	camera focal length
b	base width
d	disparity
$\mathbf{O}_1, \mathbf{O}_2$	origins of first and second camera in world coordinates

Region Detection and Tracking

r_i^m	region m in image I_i with $m \in M$, where M is total no. of regions
$r_i^{l'}$	region in image I_i projected into image I_{i+1} using a homography $H_{i,i+1}$
$r_i^m \leftrightarrow r_{i+1}^n$	region correspondence in two neighboring images
s_r, h_r	region scale and shape
s_u	region shape uniqueness
n_m	no. of neighbors for a given region r_i^m within the same image
$d_G(r_i^m, r_{i+1}^n)$	gating distance between two regions for correspondence assignment
$\mathcal{R}_i$	set of regions in image I_i
$\mathcal{L}$	set of initial region correspondences (including ambiguities)
$\mathcal{L}_o$	set of unique region correspondences (after combinatorial optimization)
p_i, p_o	region inlier and outlier score

p_l	track length
p_n	no. of tracks
ϕ	turn-table rotation angle
$\mathbf{r}$	turn-table rotation axis

Distance Measures

$d_o(r_i^{m\prime}, r_{i+1}^n)$	(symmetric) area overlap error of a region correspondence
$d_l(r_i^{m\prime}, r_{i+1}^n)$	(symmetric) position error of a region correspondence
$d_d(r_i^m, r_{i+1}^n)$	euclidean descriptor distance between two regions
$d_e(r_i^m, r_{i+1}^n)$	epipolar distance between two regions
$d_{o,max}$	threshold on the max. permissible overlap
$d_{d,max}$	threshold on the max. permissible descriptor distance
$d_{e,max}$	threshold on the max. permissible epipolar distance
d_r	residual error of turn-table model estimation
d_p	residual error of regression plane estimation
$\mathcal{C}_d$	set of descriptor-based region correspondences
$\mathcal{C}_o$	set of overlap-based region correspondences
$\mathcal{C}_{d,i}$	inlier elements from $\mathcal{C}_d$ which also exist in $\mathcal{C}_o$
$\mathcal{C}_{d,o}$	outlier elements from $\mathcal{C}_d$ which do not exist in $\mathcal{C}_o$

Bayesian Filtering

$s_{1:i}$	sequence of filter states
$z_{1:i}$	sequence of measurements
$p(s_i \vert z_{1:i-1})$	a-priori probability distribution
$p(s_i \vert z_{1:i})$	a-posteriori probability distribution
$p(s_i \vert z_i)$	state likelihood in the light of the current measurement
P_i^-	*a-priori* estimate error covariance
P_i	*a-posteriori* estimate error covariance
E, F, G	state transition matrix, control matrix and measurement matrix
K	gain or blending factor
S	measurement noise covariance
Q	process noise covariance
$\hat{s}_i^-, \hat{s}_i$	a-priori and a-posteriori estimates of the process state at time step i
ν_i	measurement innovation

Graph-related Symbols

$\mathcal{D}$	set of interdependent tracks prior to graph construction
g	single graph
$\mathcal{G}$	set of graphs

A	graph adjacency matrix
C	cost matrix for weighted graph traversal
$\mathcal{F}$	graph node dependency list
$\mathcal{N}, \mathcal{E}$	graph nodes and edges
$\mathcal{N}_s, \mathcal{N}_e$	no. of start and end nodes in g
m_e	extended path coherence weight
c_e	extended cost function for weighted graph traversal
P	predecessor matrix for graph traversal

Terminology

HARAFF	Harris-affine region detector
HESAFF	Hessian-affine region detector
IBR	Intensity-based region detector
EBR	Extrema-based region detector
MSER	Maximally-stable extremal regions detector
SIFT	Scale-invariant feature transform
SPIN	Spin-images
positives	accepted region correspondences after combinatorial optimization
negatives	rejected region correspondences after combinatorial optimization
KF	Kalman-filter
EKF	extended Kalman-filter
UKF	unscented Kalman-filter
LoG	Laplacian-of-Gaussian
DoG	Difference-of-Gaussian

1 Introduction

Classification and general overview. This thesis considers the task of spatial motion reconstruction from image sequences using a stereoscopic camera setup. In a variety of fields, knowledge on the motion of observed objects is of great importance. In environmental physics for example, velocity fields within flows are frequently observed by means of two fiber optic endoscopes. Using the latter, the motion analysis is performed by tracking a large number of small, neutrally buoyant particles through a sequence of endoscopic images (3D-PTV). In mechanical engineering, knowledge of the oscillation characteristics of a rigid object is often used to gauge the damping behavior of shock-absorbing elements. Since some time now, such processes are often analyzed by means of computer vision techniques, mostly using specially designed photometric marker elements.

This work considers each algorithmic step of the motion reconstruction problem using a set of freely available image sequences. The presented concepts and evaluation results are of a generic nature and may thus be applied to a multitude of applications in various fields, where motion can be observed by two calibrated cameras.

Normally, the first step in the processing chain of a motion reconstruction algorithm is concerned with the automated identification of salient locations (=features or regions) within each image of a given sequence. At this point, available methods may be divided into two classes. With *active methods*, the observed object is manually (*i.e.* actively) covered with artificial (and often coded) marker elements, which allow for a unique, accurate and fast localization within the images by means of photogrammetric techniques. In cases where such markers are not available or their application is not feasible, the natural texture of the observed object has to be used instead. Such *passive methods* must provide a means of detecting texture elements which allow for a unique and reliable (re-)localization within all images of a sequence, even under viewpoint changes due to object or camera motion. This thesis is focused on passive detection methods only.

A major contribution is the detailed and thorough comparison of five well-known detection methods from the contemporary literature with regard to a number of performance measures. The given results extend the available publications on the topic and facilitate the well-founded selection of an appropriate detector according to the requirements of specific target applications, such as localization accuracy or the robustness under perspective distortions.

In the second step, both spatial and temporal correspondences have to be established between features extracted from different images. With the former, two images taken at the same time instant but with different cameras are considered (stereo reconstruc-

tion) while with the latter, correspondences are sought between temporally adjacent images from the same camera instead (monocular feature tracking). In the literature, both approaches are usually treated separately: Either an observed object is spatially reconstructed at a single time instant yielding a set of three-dimensional coordinates, or its motion is analyzed separately within each camera yielding a set of two-dimensional trajectories.

The second major contribution of this work is a concept for the unification of both stereo reconstruction and monocular tracking. Based on sets of monocular trajectories from each camera of the stereo setup, the proposed method uses a graph-based approach to find correspondences not between single features but between entire trajectories instead. Thereby, the influence of locally ambiguous correspondences is mitigated significantly. The resulting spatial trajectories contain both the three-dimensional structure *and* the motion of the observed objects at the same time. To the best knowledge of the author, a similar concept does not yet exist in the literature. In this context, a novel graph-based technique for monocular region tracking is proposed as well. It is shown in a detailed evaluation that the latter is superior to standard techniques with regard to several performance measures.

In the following, a detailed overview of the general structure and of the contents of each chapter is given.

Feature detection and monocular tracking. In computer vision, tracking means to maintain correspondence between salient structures over multiple frames of an image sequence. The attribute 'monocular' indicates, that only a single camera is used and hence the resulting trajectories are two-dimensional. Generally, the multitude of existing tracking concepts can be divided into two broad categories, *dense* and *sparse* methods.

Dense tracking methods try to provide information on the motion of every pixel within an image. Without claiming completeness, two major techniques can be attributed to this group, namely optical flow methods and correlation-based methods, which rank among the earliest tracking approaches. Assuming that appearance changes between a pair of images are small, a window of predefined size around a location in the first image is used to determine the corresponding location in the second one by shifting an equally-sized window within the proximity of the original location and by maximizing the signal-correlation between both windows. If the appearance change between frames is too large, tracking failures frequently occur with this method. Therefore, and because of their comparatively high computational load, correlation-based techniques are seldom used nowadays in the context of tracking. A concept that is closely related to signal correlation is the estimation of the *optical flow*. In computer vision, the latter denotes a vector field, that provides information on the direction and velocity of a point through a sequence of images. Contrary to correlation-based methods, flow-based techniques are able to incorporate a more sophisticated model of structure appearance into the estimation process. However, they also require inter-image motion to be sufficiently small. In

order to successfully compute the optical flow on sequences with larger disparities, pyramidal approaches are often used that start on a coarse (down-sampled) representation of the original image and successively increase the resolution in order to improve estimation accuracy. A significant drawback of flow-based methods however is their sensitivity to illumination changes.

Sparse tracking methods on the other hand select only subsets of all pixels in an image. In the following, such a subset is referred to as a *feature*. The latter may consist of a single pixel or of a group of pixels (which is termed a *region*). As not every pixel in an image provides sufficiently discriminatory information for a unique and accurate re-localization in neighboring frames, a pre-selection of suitable candidates by an appropriate detection method is often advantageous over dense trackers, as unstable candidates can be sorted out prior to the actual tracking task. Depending on the selection method, single features are generally more robust to appearance variations, caused by view or illumination changes. A well-known example is the *Harris*-detector, which evaluates the image gradients within a circular support area around each position. A prominent feature is defined as one, that yields a strong signal variation in both coordinate directions within the support area, indicated by two sufficiently large eigenvalues of the second-order moment matrix. Edges and line crossings are examples of feature types, to which the *Harris*-detector responds very well. Typically, they are well-localized and robust against small perspective and illumination changes. In the case of larger perspective changes, detection accuracy rapidly degrades for most classical detection methods. The major reason for this behavior lies in the symmetry of the support area, which is mostly circular and of fixed size.

To this end, several more robust approaches have been developed within the last years, which aim at extracting stable features that change co-variantly with the image transformation. Such methods usually try to automatically adapt the shape and size of the support area in accordance with the local image structures, *e.g.* by estimating the parameters of an enclosing ellipse. In the literature, they are most often referred to as *affine-covariant region detectors*. Under the assumption of both a sufficiently small scale and a planar object surface, a perspective transformation may well be approximated by an affine one - hence the attribute 'affine-covariant'.

In chapter 2 of this thesis, a selection of five well-known affine-covariant region detectors from the literature is compared to each other with respect to a number of appropriate performance measures, using a set of freely available standard image sequences. Among the investigated measures rank (a) the localization accuracy in terms of both region position and shape of the enclosing support area, (b) the number and percentage of successfully matched regions between temporally neighboring images and (c) the dependency of localization accuracy on intrinsic properties such as scale, shape and the density of regions in a local neighborhood. The given results can be used to identify and remove error-prone regions from further processing on the basis of their properties alone. Both accuracy and robustness of a subsequent application (*e.g.* feature tracking) are thereby

significantly improved. The presented results extend the available literature on the topic and serve as a prerequisite for the subsequent chapter within this thesis. However, they may as well be used in a self-contained way for the selection of suitable detectors with respect to the needs of specific target applications.

Chapter 3 is then concerned with the task of two-dimensional (*i.e.* monocular) feature tracking. Here, the assessment of the very same affine-covariant detectors will be continued in terms of their suitability for tracking in monocular image sequences. Three algorithms of increasing complexity based on the detection methods from the first part are presented, evaluated in detail and compared against a well-known tracking method from the literature. The first (and most simple) of these methods searches for region-to-region correspondences within a circular gating region of constant size and decides on the pair with highest similarity. The latter is expressed in terms of the euclidean distance between histogram-based region descriptors, which are computed from the image intensity signal within the region support area. The second algorithm additionally employs a Bayesian filtering framework to predict the presumed location of features in future images. Correspondence ambiguities (which occur if a feature from the first image associates to several features from the second image and vice versa) are resolved by means of combinatorial optimization. The third algorithm preserves the entire multitude of ambiguous correspondences in a graph-like structure until the end of the sequence or for a predefined number of frames. Unique trajectories are extracted from these graphs by means of shortest-path search. The algorithm offers a convenient way of integrating additional measures other than descriptor similarity into the tracking process (such as a path coherence model), making it both flexible and robust against locally ambiguous regions. This is especially advantageous in the case of repetitive image structures in a local environment. The graph-based tracking algorithm is the second major contribution of this thesis, which is in many ways superior to standard tracking approaches from the literature. The presented results serve as an essential component of the last part of this thesis but may as well be used independently for applications that require robust and accurate two-dimensional feature tracking.

Spatial reconstruction and binocular tracking. In chapter 4, motion analysis is extended from the monocular single-camera case to binocular tracking with two cameras. Compared to monocular tracking, corresponding features are now observed by different cameras. Depending on both measurement setup and camera hardware, this might lead to synchronization errors, strong perspective distortions between the two viewpoints and different imaging characteristics (such as brightness, contrast or sensor noise). Also, instead of searching for potential region correspondences within a two-dimensional gating area as with monocular tracking, a one-dimensional search along the respective epipolar line is sufficient (given the camera calibration parameters). In classical stereo reconstruction, only two images taken at the same time instant are considered, giving rise to mismatches (especially in the case of low descriptor distinctiveness). Depending on

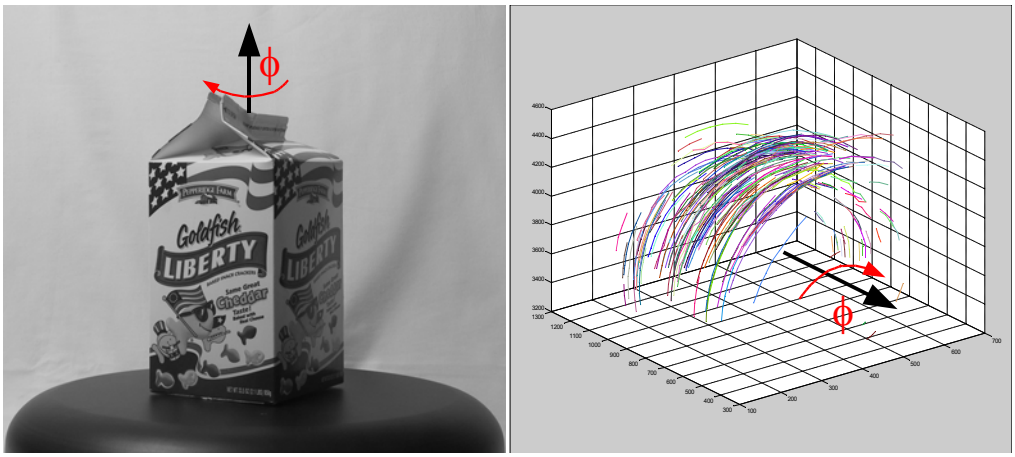


Figure 1.1: Measurement object on top of an automated turn-table (left) and a corresponding set of three-dimensional trajectories (right) over a sequence of frames. The black arrow indicates the turn-table rotation axis, while ϕ denotes the degree of rotation.

both focal length and distance of the two cameras from each other, such mismatches may lead to severe errors in spatial depth of the reconstructed points. One of the major contributions of this thesis is thus the proposition of a novel stereo technique, which greatly reduces the number of such matching errors, especially with less distinctive region descriptors. In the proposed approach, affine-covariant regions in both cameras are tracked separately from each other using the results from chapter 3. Then, correspondence analysis is performed based on the resulting sets of monocular trajectories. The algorithmic core of the proposed approach is a graph-based matching procedure similar to the one used in graph-based monocular tracking. It provides a convenient means of integrating all potential correspondence candidates between both camera views into a set of spatial graphs in order to mitigate the influence of locally ambiguous features. From the resulting set of graphs, three-dimensional trajectories may be obtained by means of shortest-path search. These contain both the three-dimensional structure *and* the motion of the observed objects at the same time.

The left side of figure 1.1 shows a single frame taken from one of the evaluated image sequences as an illustrative example. The object on top of the automated turn-table is rotated around the indicated axis by a fixed increment ϕ between every two frames. On the right side, a set of three-dimensional trajectories can be seen, which has been obtained using the proposed binocular tracking approach.

2 Evaluation of Region Detectors

2.1 Chapter Introduction

In this chapter, a selection of five well-known methods for the detection of salient features from grayscale images is evaluated and compared to each other with regard to their accuracy. Although such an assessment is of great importance for the well-founded selection of appropriate detectors for a given target application (such as tracking or stereo reconstruction), there exists no exhaustive comparative evaluation in the available literature yet.

In computer vision, many tasks rely on the accurate detection of low-level features in images such as matching applications, object recognition and retrieval methods or tracking applications. Generally, a feature can be defined as any salient structure in an image that enables a unique localization in both coordinate directions, *e.g.* a corner or a sufficiently textured image region. There exists a multitude of methods for the localization of such features, specialized on different forms and representations. Among the most commonly used features rank structures such as edge and line intersections, corners, or blobs and ridges.

While the computational complexity for the detection of such relatively simple feature types is low, they exhibit a significant drawback. In the case of image distortions, such as scale changes or perspective effects due to object or camera motion in world space, the reliable detection and thus also the assignment of corresponding features between two frames of an image sequence is significantly degraded. Most often, this problem is related to the circular convolution kernels used for the computation of image gradients. Although some methods are at least able to compensate for variations of the image scale, such as the scale-adaptive blob-detection method proposed in [Lin98], significant changes in perspective still pose a severe problem. To this purpose, a new class of region detectors has emerged in recent years: Methods such as the *Harris-affine* and *Hessian-affine* detector [MS04] or the *maximally-stable extremal regions* detector [MCUP04] are covariant to affine transformations, which makes them sufficiently robust to moderate perspective distortions on a local scale. The term *covariant* refers to the ability to change or vary in accordance with a given transformation. These methods have been used in a great variety of applications, such as panoramic image fusion [BL03] or two-dimensional tracking [DB06] and are generally referred to as *affine-covariant* region detectors. For applications, where significant object or camera motion and thus perspective effects can be expected, the use of such methods is usually advantageous over most

classical approaches (*e.g.* the well-known Harris-detector) - albeit at the cost of an increased computational complexity. In the context of this work, the expressions *feature* and *region* are used in a similar context. While a feature in the original sense usually refers to a two-dimensional location in an image, a region includes a support area around this location as well. If not stated otherwise or dictated by the context, the reference to an image region usually includes the meaning of a feature as well.

In matching or tracking applications, correspondences between the regions of neighboring frames have to be reliably identified. This task can be achieved by means of a descriptive measure, which is usually computed from the image content within the (geometrically and photometrically normalized) support area. In the literature, a wide variety of such methods exist, which are usually termed *region descriptors*. An excellent overview of such methods may be found in [MS05]. In order to assign two regions to each other, the normalized euclidean or Mahalanobis distance between the associated descriptors in the high-dimensional descriptor space has to be computed. In the case of assignment ambiguities, which occur if several features in one frame claim the same feature in another frame or vice-versa, dependencies have to be further resolved on the basis of an appropriate optimization approach.

In the existing literature, a great number of both affine-covariant detectors and appropriate descriptors already exists. Depending on the demands of the target application, on the measurement environment but also on intrinsic properties of each method, not every detector-descriptor combination provides satisfactory results. To this end, two relevant surveys exist in the contemporary literature. In [MTS$^+$05] and [MS05], a selection of the most popular representatives of region detectors and descriptors is compared to each other in different combinations with regard to measures such as the percentage of successfully matched regions, the robustness against certain image transformations (*e.g.* scale or perspective), and computational complexity. Also, the effort is made to find for each detector the most appropriate descriptor. The underlying image sequences generally show planar scenes, which greatly simplifies a meaningful performance assessment. In [MP07], the evaluation of the same methods is further extended to three-dimensional arbitrary rigid objects.

However, there exists no comparative evaluation with regard to the accuracy of the detected regions in terms of two-dimensional image position or shape of the surrounding support area. For a great variety of applications, the knowledge of the expectable accuracy is of great importance. If such information were available, a pre-selection of suitable detectors on the basis of application requirements would be possible (*e.g.* for pose estimation, tracking or stereo reconstruction). One of the major contributions of this chapter is thus a detailed evaluation of the five most-common region detectors with regard to their accuracy.

To this end, the image sequences introduced in the previous chapter are used, as they allow for the accurate estimation of groundtruth transformations between the object poses in neighboring frames. Based on the latter, corresponding features may be

identified using a measure for the relative overlap of the associated support area. It will be shown, that significant differences between the detectors exist. Also, it will be seen that localization accuracy in both position and shape of the support area often depends on intrinsic properties such as scale, shape or region density with an image. The given results enable the identification and removal of error-prone regions on the basis of their properties alone, which may greatly reduce the computational complexity of subsequent applications (*e.g.* monocular tracking).

In practice, homographies for correspondence assignment are usually not available. Therefore, region descriptors such as the *scale-invariant feature transform* (SIFT) or *spin-images* (SPIN) are used to identify matches between the images of a sequence. Depending on the detector, a different set of correspondences than with overlap-based assignment results. It will also be shown in this chapter, that the accuracy and number of successful matches among the resulting correspondences are lower than with overlap-based matching. To this purpose, both SIFT and SPIN are compared against each other with regard to the above-mentioned measures. In order to reduce set differences and thus improve the accuracy of descriptor-based matching, several strategies for the removal of error-prone correspondences are discussed and compared. A new strategy - *shape uniqueness* - is proposed, which is able to identify potentially ambiguous and thus error-prone regions prior to the matching step. As a consequence, the complexity of descriptor-based correspondence search can be significantly reduced.

The evaluation results within this chapter serve as a useful supplement to existing comparative studies and facilitate the selection of appropriate detectors for target applications. The major contributions have also been published in [HJA08b].

This chapter is organized as follows: In the background section 2.2, five affine-covariant region detectors are introduced, along with a general overview of feature detection in computer vision. Further, two methods for histogram-based region description are briefly explained in the same section (SIFT and SPIN), as well as a method for homography estimation from feature correspondences. In section 2.5.1, the estimation of inter-frame homographies is discussed in detail. The latter serve as the basis for the thorough evaluation of the affine-covariant region detectors in section 2.5. Finally, a concluding summary is given in section 2.6.

2.2 Background

2.2.1 Camera Calibration

2.2.1.1 Motivation and Overview

When observing a scene by means of a camera, the three-dimensional world is reduced to a two-dimensional image. In the simple case of *central projection*, a 3D-point in space is mapped to the image plane by drawing a straight line through the center of

projection. The intersection of this line with the image plane represents the respective image position.

However, such a basic model is not sufficient if accurate metric information is sought, because effects such as radial lens distortions or lens misalignments are not accounted for. In both photogrammetry and computer vision, numerous approaches to devise more refined and sophisticated models have been developed over the decades. The estimation of these model parameters is generally referred to as *camera calibration*. In [Zha99], a rough partitioning of concurrently available methods into two broad groups is given:

Photogrammetric calibration refers to techniques, where a calibration object is observed whose exact geometry is known. Usually, such objects consist of several orthogonal planes and allow for highly accurate and efficient calibration [Fau93]. A significant drawback of such approaches is the need for a complex and often expensive measurement setup.

Self-calibration techniques on the other hand do not require any specially-devised calibration objects. Instead, they rely on arbitrary rigid *and* static objects in a scene, which generally provide two constraints on the internal camera parameters from a single camera displacement [MF92]: In order to estimate both internal and external camera parameters (as defined below), corresponding objects between three images taken with fixed internal parameters have to be identified. In this way, three-dimensional structure can be recovered up to a similarity [Har94][LF97]. However, both reliability and accuracy of such approaches are still inferior to photogrammetric methods [Bou98].

2.2.1.2 Calibration by Viewing a Plane From Unknown Orientations

A method that combines the advantages of both fields - high accuracy, ease of use and low equipment cost - is proposed in [Zha99]. The author presents a calibration technique which is based on a planar checkerboard pattern observed under several orientations. The cost and construction effort for this simple calibration object is negligible, compared to the above-mentioned photogrametric techniques. Without prior knowledge on motion, either the pattern or the camera can be moved around in the scene almost arbitrarily. Jean-Yves Bouguet [Bou99] provides a freely available toolbox[1], which is mainly based on the work of Zhang and has become a *de facto* standard in the computer vision community. It consists of four successive stages, which are briefly described in the following.

1. Image Acquisition. In figure 2.1, two different views of a checkerboard calibration pattern are shown. The crosses and rectangles indicate the position of the features

[1] `http://www.vision.caltech.edu/bouguetj/calib_doc/`

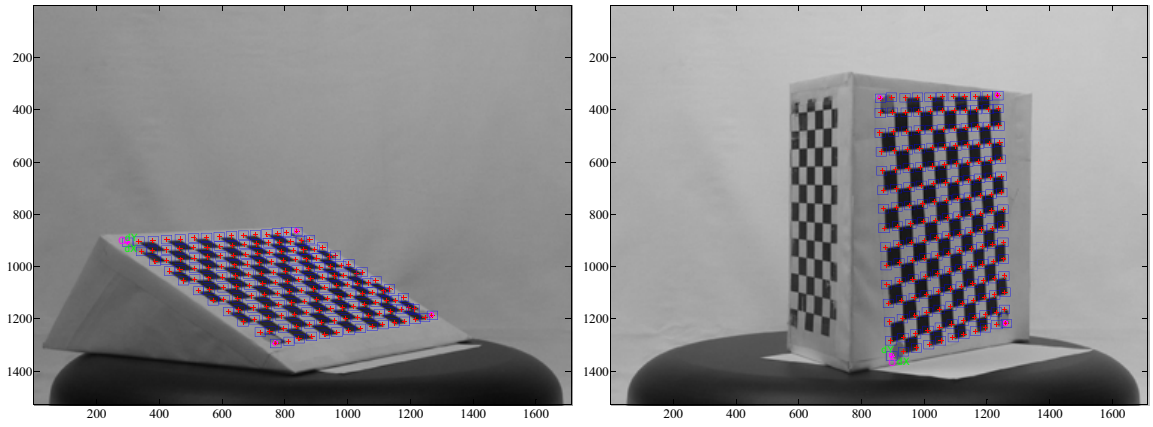


Figure 2.1: Two different views of the checkerboard calibration pattern. The crosses and rectangles show the position of the corners used by the algorithm.

(grid corners) used by the algorithm. By moving either the plane or the camera, several images under different orientations have to be taken in order to successfully estimate the camera parameters.

2. Feature Detection. Naturally, the quality of the estimated parameters directly depends on the accuracy of feature detection. Common choices for features are *e.g.* centers of gravity of circles or squares, line intersections or corners (as in a checkerboard pattern). The latter offer the advantage of highly accurate and well-distinguishable locations under a multitude of viewpoints and lighting conditions. Due to the known symmetry of the pattern, localization bias is avoided which makes such techniques one of the most widely used choice for 2D calibration at the time [MA07].

Feature detection is performed using the traditional corner detector of Harris and Stephens [HS88][MS04]. This method is based on the evaluation of the second moment matrix μ, which is computed directly from the image intensity signal I as

$$\mu(\mathbf{x}, \sigma_i, \sigma_d) = \sigma_d^2 g(\sigma_i) * \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_d) & L_x L_y(\mathbf{x}, \sigma_d) \\ L_x L_y(\mathbf{x}, \sigma_d) & L_y^2(\mathbf{x}, \sigma_d) \end{bmatrix}, \tag{2.1}$$

where L_x and L_y denote the local derivatives at position $I(\mathbf{x})$ where $\mathbf{x} = [x, y]^T$. Before derivation, the image is smoothed with a Gaussian kernel with standard deviation σ_d in order to mitigate aliasing effects. After computing the local derivatives for all coordinate directions, the results are smoothed again with a second Gaussian kernel $g(\sigma_i)$ in order to integrate sufficient support from the neighborhood around each location.

The Gaussian integration kernel g is dependent on σ_d via the relation $\sigma_d = k\sigma_i$ with $k < 1$. Sometimes, equation 2.1 is also referred to as *structure tensor* [Jäh05].

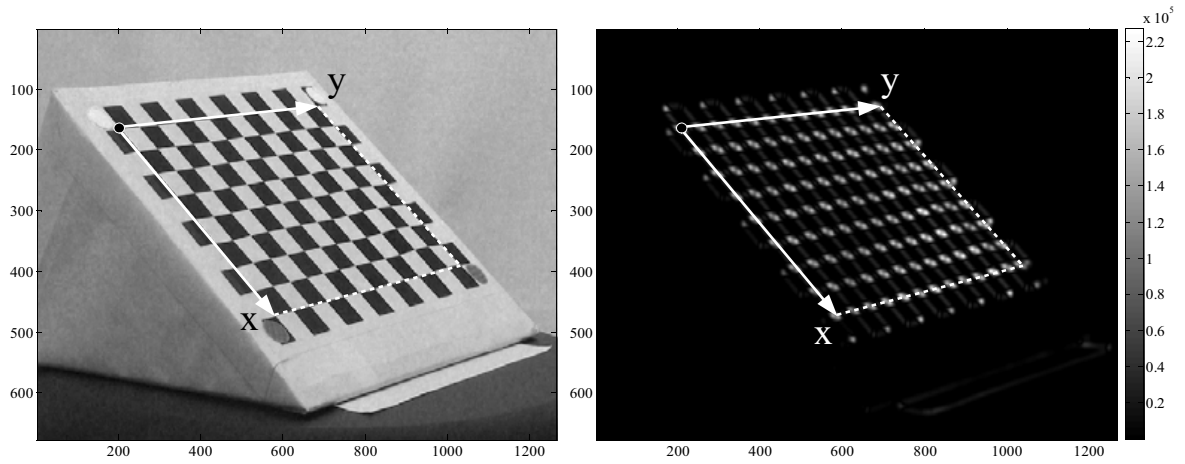


Figure 2.2: View of the calibration pattern (left) and the corresponding *cornerness* response image according to equation 2.2 with $\alpha = 0.05$ and $\sigma_i = 5$.

The two eigenvalues λ_1 and λ_2 of $\mu(\mathbf{x}, \sigma_i, \sigma_d)$ provide information on the intensity values within the integration region controlled by σ_i. Two small eigenvalues indicate a homogeneous region, a single large eigenvalue indicates a region with a mostly linear structure (such as an edge) and two large eigenvalues denote a region which allows for an accurate and unambiguous localization in the image plane in two dimensions (such as a corner). Usually, such a structure is present if the smaller eigenvalue exceeds a predefined threshold. Harris and Stephens suggest the following function for the detector response r_d where α is a measure for the so-called *cornerness*:

$$r_d = \det \mu - \alpha \text{trace}(\mu)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 \tag{2.2}$$

The parameter α penalizes regions whose structure is mainly linear (as with lines or edges) and is usually found in the range 0.1 ± 0.05 [MA07]. Features may be extracted by performing a non-maximum suppression on r_d. Figure 2.2 shows an image of the calibration pattern (left) and the corresponding *cornerness* response image according to equation 2.2 (with $\alpha = 0.05$, $\sigma_i = 5$ *pel*). It can be seen, that the response function around the checkerboard corners is significantly stronger than with the grid patches.

3. Estimation of extrinsic and intrinsic camera parameters. The relation between a 3D point $\mathbf{X} = [X, Y, Z, 1]$ given in homogeneous coordinates and its projection $\mathbf{x} = [x, y, 1]$ by a pinhole camera is given by

$$s\mathbf{x} = U[R \quad \mathbf{t}]\mathbf{X} \qquad \text{with} \quad U = \begin{bmatrix} f_c & c & x_0 \\ 0 & f_c & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

where s is an arbitrary scale factor and R and $\mathbf{t}$ describe the camera rotation and translation in the world coordinate system respectively. U is called the camera intrinsic matrix and the vector $[x_0, y_0]$ denotes the location of the principal point (which is the intersection of the optical axis with the image plane). The parameter f_c denotes the camera focal length while c describes the skewness of the pixel axes. Normally, the latter are assumed to be orthogonal and thus $c = 0$. For an exhaustive description of this model, the reader is referred to [HZ03]. The estimation process described in the following is mainly summarized from [Zha99].

By positioning the model plane at $Z = 0$ without loss of generality, equation 2.3 becomes

$$s\mathbf{x} = U[\mathbf{r_1} \quad \mathbf{r_2} \quad \mathbf{r_3} \quad \mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = U[\mathbf{r_1} \quad \mathbf{r_2} \quad \mathbf{t}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{2.4}$$

where r_i denotes the *i-th* column of the rotation matrix R. The reduced expression is a homography that relates a world point $\mathbf{X}$ to its image $\mathbf{x}$ via

$$s\mathbf{x} = H\mathbf{X} \qquad \text{with} \quad H = U[\mathbf{r_1} \quad \mathbf{r_2} \quad \mathbf{t}] \tag{2.5}$$

where the 3x3-matrix H is defined up to a scale factor.

Given n images of the checkerboard pattern and m corners within each image which are assumed to be corrupted by independent and normally distributed noise, a maximum likelihood estimate of the model parameters can be obtained by minimizing the functional

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \|\mathbf{x}_{ij} - \hat{\mathbf{x}}(U, R_i, \mathbf{t}_i, \mathbf{X}_j)\|^2, \tag{2.6}$$

where $\hat{\mathbf{x}}$ is the projection of the 3D point $\mathbf{X}_j$ in image I_i, according to equation 2.5 and $\mathbf{x}_{ij}$ denotes the actual corner location. The minimization of equation 2.6 is a nonlinear problem which is solved by using the Levenberg-Marquardt algorithm as described in [Mor77a]. For more detail on the estimation procedure and the need for an initial guess of the sought parameters, the reader is referred to [Zha99].

4. Estimation of radial and tangential distortion. The lens distortion model used in Bouguets calibration toolbox is mainly based on [Bro71]. Considering only the first two terms of radial distortion k_1 and k_2, the relation between the ideal distortion-free normalized image coordinates $[x, y]$ and the observed (distorted) normalized coordinates $[\hat{x}, \hat{y}]$ is given by

$$\hat{x}_r = x + x\left[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2\right] \tag{2.7}$$

$$\hat{y}_r = y + y\left[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2\right] . \tag{2.8}$$

The center of the radial distortion is assumed to be identical to the location of the principal point $[x_0, y_0]$.

However, centers of curvature of lens surfaces are not always strictly collinear. This introduces another common distortion type, *decentering distortion* which has both a radial and tangential component [HS97]. While the expression for radial distortion is identical to equations 2.7 and 2.8, tangential distortion is formulated as

$$\hat{x}_t = x + 2p_1xy + p_2\left[x^2 + y^2 + 2x^2\right] \tag{2.9}$$

$$\hat{y}_t = y + p_1\left[x^2 + y^2 + 2y^2\right] + 2p_2xy\ . \tag{2.10}$$

The additional parameters k_1, k_2 and p_1, p_2 (resulting from the linear combination of both radial and tangential model) can be easily included into the estimation procedure according to equation 2.6 by extending the second term of the absolute difference to $\hat{\mathbf{x}}(U, k_1, k_2, p_1, p_2, R_i, \mathbf{t}_i, \mathbf{X}_j)$.

The information provided within this section forms the basis of the more general description of the calibration procedure in section 2.4.

2.2.2 Feature Detection

2.2.2.1 Motivation and Overview

In computer vision, many tasks rely on the accurate detection of low-level features in images such as *matching applications*, *object recognition and retrieval methods* or *tracking applications*. This section gives a brief overview of available feature detectors and region detectors and tries to point out the major differences. Historically, the term *feature* refers to a point of special interest in an image, where some kind of prominent signal variation occurs. Generally, a feature or *interest point* is characterized by the following properties

1. It has a well-defined position $\mathbf{x} = [x, y]^T; \quad x, y \in \mathbb{R}$ in the image.

2. The area around $\mathbf{x}$ should exhibit a strong two-dimensional signal variation such that a unique localization is possible (*e.g.* corners and line crossings opposed to mere edges or straight lines).

However, as shown in [SMB00], detectors which only fulfill the two claims are often not sufficiently stable under projective transformations, especially if significant perspective distortions or changes of the image scale occur. For intensity-based methods (as introduced below), the major reason for this drawback lies in the symmetry of the convolution kernel used for detection, most often a circular Gaussian. Thus, a multitude of enhanced approaches has emerged over the last decade which aim at extracting features that change covariantly with the image transformation. Such methods usually try to adapt the shape of the respective convolution kernels to the structure of the underlying

Figure 2.3: When a planar object (*left*) is rotated in space, its image is transformed perspectively and (depending on the light source) photometrically. If a circular kernel is used for feature detection, the image content beneath is not the same after the transformation (*middle*). In the right image, the shape of the kernel is automatically adapted to the underlying image structure. Given such a region, it is possible to normalize against both geometric and photometric distortions in order to obtain viewpoint and illumination invariance.

image content (*e.g.* in [Bau00]). In the literature, they are most often referred to as *covariant region detectors*, because they provide additional information on the shape of the surrounding area around a salient location and change in accordance with the respective image transformation. In addition to the above claims, such detectors demand that

3. under both photometric and geometric transformations of the image, the *position and shape* of the feature and its support area should remain stable and transform accordingly.

Figure 2.3 illustrates the concept: When a planar object (*left*) is rotated in world space, its image is transformed perspectively and (depending on the light source) photometrically. If a circular support region is used for representing the area around a salient feature, the image content beneath is not the same after the transformation (*middle*). In the right image, the shape of the support region is automatically adapted to the underlying image structure. Given such a region, it is possible to normalize against both geometric and photometric distortions in order to obtain viewpoint and illumination invariance. Section 2.2.2.2 gives an overview of the historical development of such methods and describes in detail the five region detectors which are used throughout this work.

In [SMB00], an extensive overview of feature detectors is given, which is briefly summarized in the following: The group of available detectors can be broadly partitioned into three major categories: After extracting contours in the first step, *contour-based methods* search for maximal curvature or inflexion points, sometimes based on an additional polygonal approximation of the contour chains. Features are defined as contour intersections. Examples of such methods are [AB86],[HVS90] or [MS98]. *Intensity-based methods* generally compute an appropriate measure that identifies the presence

of feature points directly from the image intensity signal. One of the first signal-based detectors was described in [Mor77b], where the autocorrelation function of the intensity signal was exploited. While the authors used the first derivatives, Beaudet later proposed a method based on the second derivatives of the image intensity I [Bea78]. There, localization was based on the determinant of the Hessian matrix $\det(H) = I_{xx}I_{yy} - I_{xy}^2$ instead. Moravec's approach was later extended by Harris and Stephens in [HS88], where an eigenvalue analysis of the second-order moment matrix was used to derive a *cornerness*-measure. Within the camera-calibration procedure described in section 2.4, the Harris-detector was used for the detection of features on the checkerboard calibration pattern. Finally, *parametric models* try to fit a parametric intensity model to the image signal. The advantage of these approaches is that they generally achieve good sup-pixel accuracy. However, they are most often limited to specific types of interest points and are thus suitable to only a limited number of applications. Examples of such methods are [Roh92] or [BNM98].

2.2.2.2 Overview of Affine-covariant Region Detectors

In contrast to most classical feature detectors, affine-covariant methods are especially suited for application scenarios, where either the observing camera or the objects within the scene experience a significant viewpoint change. In such cases, a robust detection method is needed, that adapts the shape and size of the feature support region accordingly. Affine-covariant detectors provide this functionality. In [MS04] and in [MTS$^+$05], an extensive overview of such methods is given, which is briefly summarized in the following.

One of the first approaches to obtain a certain degree of invariance against scale changes was presented in [CP84]. The authors computed a resolution pyramid using Difference-of-Gaussian (DoG) filters and selected salient points of the filter response as features with a distinctive location in scale space. More than one decade later, Lindeberg proposed to use the Laplacian-Of-Gaussian (LoG) instead, among other derivative-based operators [Lin98]. There, a scale-space was constructed by successive smoothing of the original image with a Gaussian convolution kernel of increasing size. Each setting of the kernel variance corresponded to one discrete scale level. In [BL98], the notion of scale-invariant interest points was additionally investigated in the context of tracking. Shortly after the original publication of Lindeberg, Lowe proposed to replace the LoG-kernel with a DoG-kernel, where successive layers of scale space (smoothed with a Gaussian) are subtracted from each other [Low99]. The advantage of Lowes approach is a significant increase in computation speed that allows for the processing of several images per second [MS04] while obtaining similar results as with the DoG-based selection method. However, all quoted methods only obtain invariance against pure scale changes.

According to the results in [MTS$^+$05], the transformation of most interest for viewpoint changes is an *affine transformation* (or *affinity*). If the region scale is small enough,

an affinity is sufficient to locally model perspective distortions arising from viewpoint changes, provided that the object surface represented by the region can be locally approximated by a plane. The above-mentioned scale invariance is a special case of affine invariance: While for the former, scaling is always identical in both coordinate directions, this must not be the case for the latter. In the case of true affine transformations, scaling can be different in each direction (shearing). In this case, the localization, size and shape of a local structure are affected. For this reason, mere scale-invariant methods generally fail in the case of significant perspective transformations.

In the literature, there currently exist six commonly-used affine-covariant region detectors, which have been subject to extensive evaluation already [MTS$^+$05][MP07]. In the following, these are briefly described with regard to their major properties. Figure 2.4 shows a section of the first and seventh image of the *carton*-sequence (see section 2.3 for more detail), on which 15 randomly selected region correspondences between both images are shown for each detector. The turn-table rotation between the views is at $\phi = 30°$.

Harris- and Hessian-affine detector. In [MS04], a technique for the detection of scale- and affine-covariant regions was presented, based on either the scale-adapted second moment matrix (Harris-affine) or on the Hessian-matrix (Hessian-affine). Apart from the initial detection step, the methods for scale and shape selection are identical for both methods.

For the *Harris-affine detector*, the localization of candidate regions is performed using the scale-adapted corner detector of Harris and Stephens [MS04] which is based on the evaluation of the second-moment or autocorrelation matrix μ, computed directly from the image intensity signal I. The corresponding equation has been previously introduced in the context of corner detection for camera calibration in section 2.2.1.

For the *Hessian-affine detector*, initial feature detection is based on the second derivatives L_{ij} of the image intensity signal $I(\mathbf{x})$ instead:

$$H(\mathbf{x}, \sigma_d) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma_d) & L_{xy}(\mathbf{x}, \sigma_d) \\ L_{yx}(\mathbf{x}, \sigma_d) & L_{yy}(\mathbf{x}, \sigma_d) \end{bmatrix} . \tag{2.11}$$

This expression is commonly known as Hessian-matrix in the literature. Generally, the second derivatives give strong responses on blobs and ridges (opposed to corners and line intersections). A candidate feature is present at position $\mathbf{x}$, if $\det H(\mathbf{x}, \sigma_d)$ attains a local maximum. The standard deviation σ_d controlls the desired feature size.

The selection of a characteristic scale (defined by σ_d) is performed according to the method in [Lin98] for both Harris- and Hessian-affine regions. A characteristic scale is present, if the following expression attains a local maximum in scale space:

$$|LoG(\mathbf{x}, \sigma_d)| = \sigma_d^2 |L_{xx}(\mathbf{x}, \sigma_d) + L_{yy}(\mathbf{x}, \sigma_d)| \tag{2.12}$$

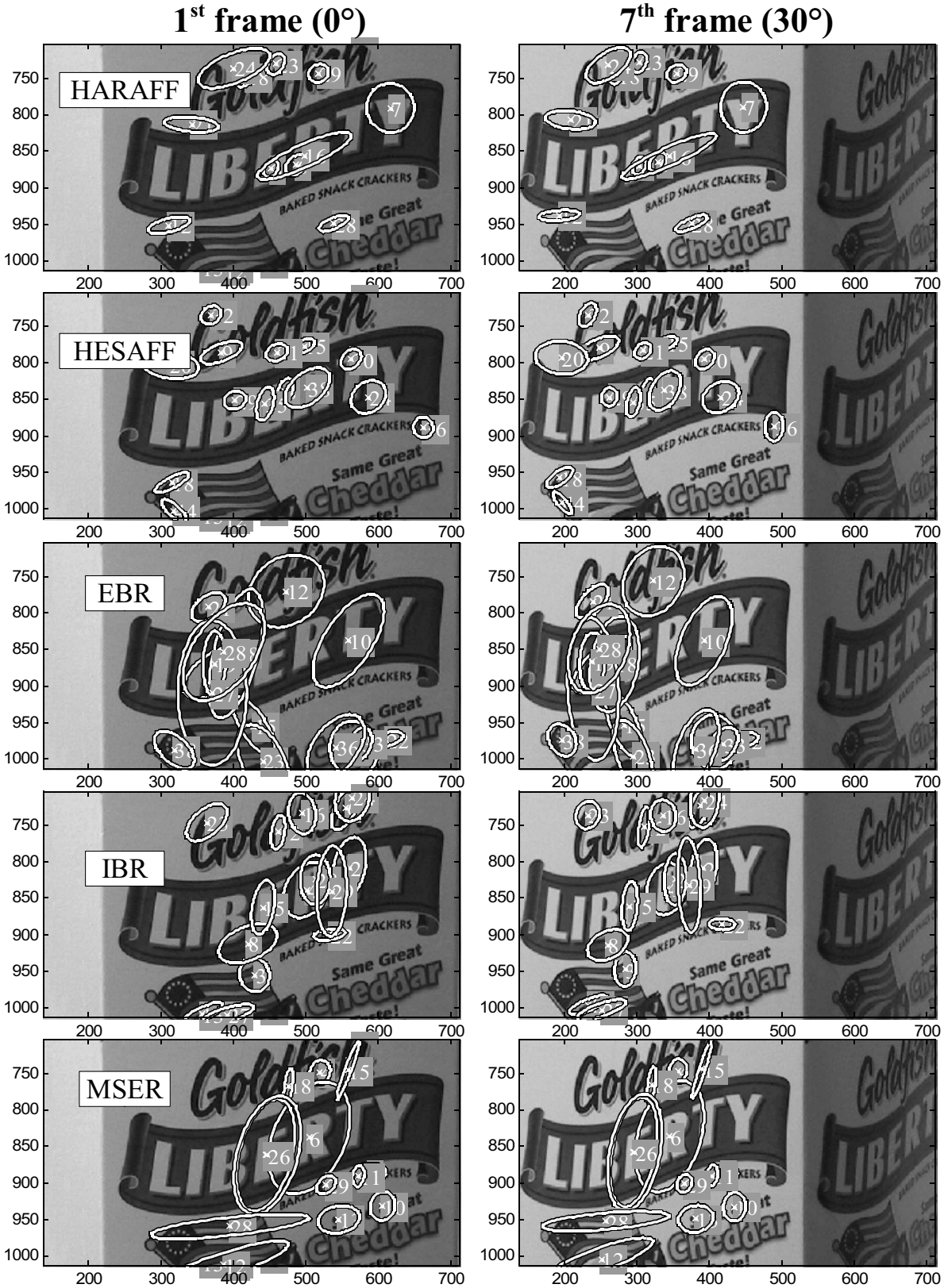


Figure 2.4: 15 randomly selected region correspondences are superimposed on a section of images 1 and 7 of the *carton*-sequence (see chapter 2.3 for more detail) for each detector (except *salient regions*). Corresponding regions are indicated via common numbers.

This function is equivalent to a matching filter and attains a maximum, if the local structure within the integration window is maximally similar to the *Laplacian-of-Gaussian* filter (LoG) (also known as *mexican hat*).

The estimation of affine region shape is performed iteratively according to the methods proposed in [LG97] and [Bau00]. There, the eigenvalues of the second-moment matrix μ from equation 2.1 are used to measure the elliptic shape of a point neighborhood. The transformation is sought which projects the affine-shaped pattern to one with equal eigenvalues [MTS$^+$05]. The latter is given by the square root of the second-moment matrix $\mu^{\frac{1}{2}}$. In the case of iterative convergence, the respective affine transformation has been identified up to an unknown two-dimensional rotation in the image plane.

The *computational complexity* for the detection of candidate points by means of second moments or the Hessian-matrix is $O(n)$, where n denotes the number of pixels in the image. Both scale and shape adaptation have complexity $O((m+k)p)$, where p is the number of candidate points, m is the number of investigated scales in the discrete scale space and k is the number of iterations in the shape adaptation algorithm.

Edge-based region detector. In [TVG04], an approach for the detection of affine-covariant regions was proposed, based on Harris corners [HS88] and Canny edges [Can86]. Tuytelaars and van Gool argue, that edges are typically stable features which are detectable over a range of viewpoints and under varying illumination. In their approach, the dimensionality of the parameter estimation problem is reduced significantly: instead of estimating all 6 parameters of an affine transformation, only a single dimension must be considered, if the local edge geometry is exploited.

The algorithm starts by detecting both corners and edges on a set of discrete scales. Starting from a corner location $\mathbf{x}_c = [x_c, y_c]^T$, two points $\mathbf{x}_1$ and $\mathbf{x}_2$ move away from the corner according to figure 2.5. The relative speed of both points is coupled through the equality of relative affine invariant parameters l_1 and l_2

$$l_i = \int \text{abs}\left(|\mathbf{x}_i^{(1)}(s_i) \quad \mathbf{x}_c - \mathbf{x}_i(s_i)|\right), \quad i = 1, 2 \tag{2.13}$$

where s_i is an arbitrary curve parameter in both directions and $\mathbf{x}_i^{(1)}$ is the first derivative of $\mathbf{x}_i$ with respect to s_i. Equation 2.13 ensures that the areas between the joint $< \mathbf{x}_c, \mathbf{x}_1 >$ and the edge *and* between the joint $< \mathbf{x}_c, \mathbf{x}_2 >$ and the edge remain identical, which is an affine invariant criterion ($l = l_1 = l_2$).

For each value l, a parallelogram is spanned by the corner point $\mathbf{x}_c$ and the vectors $\mathbf{x}_1(l) - \mathbf{x}_c$ and $\mathbf{x}_2(l) - \mathbf{x}_c$. This yields a one-dimensional family of parallelograms (and thus regions) as a function of l. A salient region is found, if several photometric functions attain an extremum over l. For brevity, the description of these functions has been omitted at this point. For more details, the reader is referred to [TVG04]. In figure 2.5, the principal concept is illustrated. Given a corner and two edge segments moving

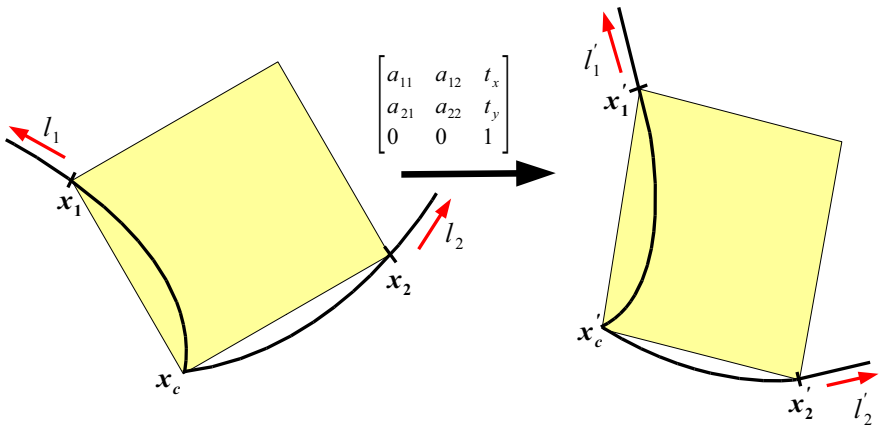


Figure 2.5: Given a corner $\mathbf{x}_c$ and two edge segments moving away from it, the resulting parallelogram (*left*) is defined by the vectors $\mathbf{x}_1(l) - \mathbf{x}_c$ and $\mathbf{x}_2(l) - \mathbf{x}_c$. Under an affine transformation, the region changes accordingly (*right*).

away from it, the resulting parallelogram (*left*) is covariant with an affine transformation. In contrast to the Harris-affine and Hessian-affine detectors, regions are not described by an ellipse but by the above-mentioned parallelograms. For better comparison of these methods, each parallelogram is replaced by the enclosed ellipse, which has the same first and second moments but is undefined up to a rotational degree of freedom [MTS+05]. Thus, this approximation should be avoided in practical applications.

The *computational complexity* for the detection of initial corners by means of the Harris corner detector is $O(n)$, where n is the number of pixels within an image. The complexity of region construction by means of evaluating the photometric functions along the edge segments is $O(pd)$, where p is the number of initial corners and d is the average number of edges in a neighborhood.

A general drawback of this method is that the edges it relies on in one image may be undetected, interrupted or connected in a different way in another image. Thus, the resulting regions may no longer be correctly transformed into each other.

Intensity-based region detector. Also in [TVG04], a second approach for the detection of affine-covariant regions was proposed, based on the localization of intensity extrema at multiple scales. Given a local intensity extremum $I(\mathbf{x}_0) = I_0$, the following function is evaluated along a set of rays emanating from $\mathbf{x}_0$:

$$f_I(t) = \frac{\text{abs}(I(t) - I_0)}{\max\left(\frac{\int_0^t \text{abs}(I(t) - I_0)dt}{t}, d\right)} \tag{2.14}$$

where t is an arbitrary parameter which describes the absolute distance from the center along each ray. The scalar d has simply been added so that a division by zero is

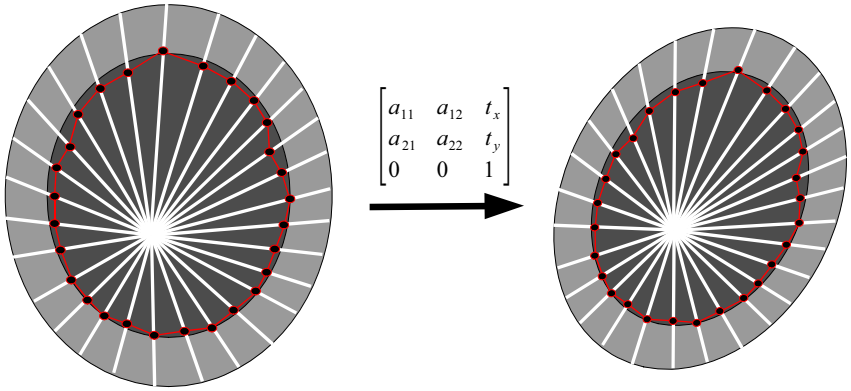


Figure 2.6: Given an intensity extremum and a set of connected components (found by evaluating equation 2.14), an elliptic region is defined (*left*). Under an affine transformation, the region changes accordingly (*right*).

avoided. Given the ray, the salient point for which equation 2.14 attains a maximum is invariant under affine geometric and linear photometric transformations. Typically, a maximum is reached if there is a sudden change in image intensity. The entire set of salient points from all rays is then linked into a connected component, from which ellipse parameters are estimated by computing the shape moments up to second order. In figure 2.6, the principal concept is illustrated. Given an intensity extremum and a set of connected components (found by evaluating equation 2.14), an elliptic region is defined (*left*). Under an affine transformation, the region changes accordingly (*right*).

The *computational complexity* for the detection of intensity extrema is $O(n)$, where n is the number of pixels. The complexity of region construction by means of evaluating equation 2.14 along each ray is $O(p)$, where p is the number of intensity extrema.

A general advantage of this method lies in the fact, that it does not rely on intermediate feature detection (such as corners or edges) but operates on the intensity function only. Thus, no image smoothing for gradient computation is required, which might be beneficial with regard to accuracy. Also, it requires only little computational effort.

Maximally-stable extremal regions detector. In [MCUP04], a method for the detection of affine-covariant regions in the context of robust wide-baseline stereo correspondence analysis was proposed, called maximally-stable extremal regions (MSER). These are defined by an extremal property of the intensity function within the region and on its outer boundary. The detection process is based on binarizing the image with a set of thresholds $t \in \{0, 1, ..., 255\}$. For each setting of t, a number of black and white regions of different size and shape results. Each region is transformed into a connected component, which serves as support area. The set of all connected components for every setting of t is termed *set of maximal regions*. If the image intensity is reversed, the result-

Table 2.1: Formal definition of the concept behind MSER, taken from [MCUP04].

Image I is a mapping $I : \mathcal{D} \subset \mathbb{Z}^2 \rightarrow \mathcal{S}$. Extremal regions are well-defined on images if:

1. $\mathcal{S}$ is totally ordered, *i.e.* reflexive, antisymmetric and transitive binary relation $\leq$ exists. In the original publication, only $\mathcal{S} \in \{0, 1, ..., 255\}$ is considered.

2. An adjacency (neighborhood) relation $\mathcal{A} \subset \mathcal{D} \times \mathcal{D}$ is defined. In the original publication, 4-neighborhoods were used, *i.e.* $p, q \in \mathcal{D}$ are adjacent (pAq), if $\sum_{i=1}^{d} |p_i - q_i| < 1$.

Region $\mathcal{Q}$ is a contiguous subset of $\mathcal{D}$, *i.e.* for each $p, q \in \mathcal{D}$ there is a sequence $p, a_1, a_2, ..., a_n, q$ and $pAa_1, a_iAa_{i+1}, a_nAq$.

(Outer) region boundary $\delta\mathcal{Q} = \{q \in \mathcal{D} \backslash \mathcal{Q} : \exists p \in \mathcal{Q} : qAp\}$, *i.e.* the boundary $\delta\mathcal{Q}$ of $\mathcal{Q}$ is the set of pixels being adjacent to at least one pixel of $\mathcal{Q}$ but not belonging to $\mathcal{Q}$.

Extremal region $\mathcal{Q} \subset \mathbb{D}$ is a region such that for all $p \in \mathcal{Q}, q \in \delta\mathcal{Q} : I(p) > I(q)$ (maximum intensity region) or $I(p) < I(q)$ (minimum intensity region).

Maximally stable extremal region (MSER). Let $\mathcal{Q}_1, ..., \mathcal{Q}_{i-1}, \mathcal{Q}_i, ...$ be a sequence of nested extremal regions, *i.e.* $\mathcal{Q}_i \subset \mathcal{Q}_{i+1}$. Extremal region $\mathcal{Q}_i$ is maximally stable if $q(i) = |\mathcal{Q}_{i+\Delta} \backslash \mathcal{Q}_{i-\Delta}| \backslash |\mathcal{Q}_i|$ has a local minimum at i ($|.|$ denotes cardinality). $\Delta \in \mathcal{S}$ is a parameter of the algorithm.

ing connected components are termed *minimal regions*. Table 2.1 has been taken from [MCUP04] and gives a formal definition of the concept behind MSER. Further details may be found in the original publication.

Regions where binarization is stable over a large range of thresholds t are of special interest, since they possess a number of desirable properties, which include (1) covariance to affine transformations of the image intensity, (2) the detection of regions over multiple scales and (3) high accuracy, since no smoothing of the image is involved. Especially the last property distinguishes the MSER detector from other detectors whose method of scale-space construction is based on successive convolution with a Gaussian smoothing kernel.

In the first step of the algorithm, all detected extremal regions are enumerated. This is done by first sorting all pixels according to their intensity. In the second step, pixels are marked in the image (either in decreasing or increasing order) and the list of growing and merging connected components and their areas is maintained using the union-find

algorithm [Sed83], which has almost linear computational complexity. The result is a data structure which stores the area of each connected component as a function of intensity. A merge of two components is viewed as termination of existence of the smaller component and an insertion of all pixels of the smaller component into the larger one. In the last step, the *maximally stable* regions are identified as those corresponding to thresholds were the relative area change as a function of relative change of threshold is at a local minimum [MTS$^+$05]. This corresponds to regions, where binarization is stable over a large range of thresholds.

The *computational complexity* of the sorting step is $O(n)$ if the range of image values is small, *e.g.* $n \in \{0, ..., 255\}$. The complexity of the union-find algorithm is $O(n \log \log n)$, which is almost linear and thus very fast.

Salient regions detector. In [KZB04], a method for the detection of affine-invariant salient regions is proposed, based on the probability density function (*pdf*) of intensity values computed over an elliptical region.

In the first step of the algorithm, candidate regions are found by evaluating the entropy of the *pdf* over a three-parameter family of ellipses at each pixel: The ellipse ϵ centered at position $\mathbf{x}$ is described by the length of its major axis s (which is the region scale), the ratio of major to minor axis λ (which is the region shape) and the orientation of the major axis ϕ. The *pdf* $p(I_\epsilon)$ of the intensity values within ϵ is used to compute the entropy as

$$\mathcal{H} = -\sum_{I_\epsilon} p(I_\epsilon) \log p(I_\epsilon) \tag{2.15}$$

Extremal points with regard to $\mathcal{H}$ are selected as candidate regions.

In the second step, these candidates are ranked by using the magnitude of the derivative of the *pdf* with respect to the scale s

$$\mathcal{W} = \frac{s^2}{2s-1} \sum_{I_\epsilon} \left| \frac{\delta p(I_\epsilon, s, \phi, \lambda)}{\delta s} \right| . \tag{2.16}$$

The saliency measure $\mathcal{Y}$ is then given by the product $\mathcal{Y} = \mathcal{H}\mathcal{W}$. From the list of all sorted salient regions, the best N are selected as final salient regions.

The *computational complexity* of the first step of the algorithm is $O(nl)$, where n is the total number of pixels and l is the number of combinations of the three ellipse parameters tried at each pixel. The complexity of the second step is $O(e)$, where e is the number of extrema provided by the first step of the algorithm.

A major disadvantage of this approach is the high computational effort of calculating the entropy at each pixel of the image. If a good shape accuracy is sought (which means a large l), the runtime of the algorithm increases significantly, which makes it unsuitable for a large number of practical applications. Therefore, this method has not been

included into the evaluation in chapters 2 and 3. Although its concept is theoretically appealing, the runtime of this algorithm exceeds all other detectors by several magnitudes according to [MTS$^+$05].

2.2.3 Rotation-invariant Region Descriptors

2.2.3.1 Motivation and Overview

Given two sets of affine-covariant regions detected in a pair of images (related by a perspective transformation), a descriptive entity is sought, which provides for each region in the first set a corresponding region in the second set. Generally, such an entity is termed a *region descriptor* in the computer vision literature. Based on different image properties as for example pixel intensities, differential structures (such as lines, corners or edges), color or texture, a multitude of methods has been developed over the last years. Region descriptors have been successfully applied so far in the fields of wide baseline matching [SZ02], object and texture recognition [FTVG04][LSP03], image retrieval [MS01] and panorama stitching [BL03].

Region descriptors are typically computed from the image information within an affine-covariant region. To this purpose, the latter is transformed (normalized) into an ellipse with identical eigenvalues, *i.e*. a circle. The one remaining degree of freedom after such normalization is an unknown two-dimensional rotation ϕ within the image plane. An appropriate descriptor should thus be invariant to the latter so that a reliable similarity measure can be computed for each potential region correspondence, regardless of ϕ.

In the following, an overview of currently available descriptors is given, which is mainly based on [MS05]. There, the multitude of methods has been divided into three classes, which are *differential descriptors*, *spatial frequency techniques* and *distribution-based descriptors*.

Differential descriptors. One of the first approaches to region description was based on the convolution of the image intensity signal with a set of Gaussian derivatives. The resulting *local jet*, which has been described in great detail in [KvD87], is a set of derivatives of order N of the image intensity signal I at position $\mathbf{x}$ at a specific scale σ_d:

$$J^N[I](\mathbf{x}, \sigma_d) = \left\{ L_{i_1..i_n}(\mathbf{x}, \sigma_d) | (\mathbf{x}, \sigma_d) \in I \times \mathbb{R}^+; n = 0, ..., N \right\} . \qquad (2.17)$$

$L_{i_1..i_n}(\mathbf{x}, \sigma_d)$ is the result of a convolution of the image signal with Gaussian derivatives (with $i_k \in \{x, y\}$). According to [tHRFSV93], invariance under 2D image rotation is reached with a set of differential invariants from the local jet up to second order $(N = 2)$.

An adequate distance measure has to take into account the different magnitudes of the descriptor components and the amounts of variability within. A measure suited for this purpose is the Mahalanobis distance [Cox93], which models component variations as

random variables with a Gaussian distribution based on a covariance matrix. The latter has to be estimated statistically from a large data sample. In general, a broad set of different image sequences is used for this purpose. The local jet has been successfully used so far in the field of image retrieval in [SM97].

Also based on the local jet from equation 2.17, *steerable filters* have been introduced in [FA91], which steer derivatives in a particular direction, making them largely invariant to rotation. In [Bau00] and [SZ02], the use of complex filters has been proposed, derived from the family $K(x, y, \phi) = f(x, y) \exp(i\phi)$, where ϕ is the filter orientation in the image plane. For $f(x, y) \exp(i\phi)$, Baumberg uses Gaussian derivatives, while Schaffalitzky and Zisserman employ a polynomial function instead.

The general disadvantage of derivative-based methods is the need for the estimation of a representative covariance matrix Λ in order to compute the correct (euclidean) distance between two descriptors. Depending on the choice of the dataset, different results for Λ may occur. Also, if only a specific type of images is used for the estimation, the applicability of the resulting descriptors to different image types might lead to erroneous results. For the given reasons, differential descriptors have not been used in the context of this work.

Spatial frequency techniques. A different approach to local region description is to decompose the image signal into a set of basis functions in the frequency domain by applying the Fourier transform. However, spatial relations between points are not explicit in this case, and the basis functions are infinite. Thus, such a method is difficult to adapt to a local approach [MS05]. One possible solution is offered by the Gabor transform [G$^+$46], which overcomes this problem but requires a large number of filters in order to sufficiently capture small changes in frequency and orientation.

Such filters as well as wavelets [VK95] are often used in the context of texture classification but are of limited use for efficient region description. As with differential descriptors, spatial frequency techniques are not used within this work.

Distribution-based descriptors. Generally, distribution-based methods use histograms to capture appearance and shape characteristics of a region. In [JH99] for example, a two-dimensional histogram was introduced, whose dimensions represent the distance from the respective region center and the normalized intensity value. This method, termed a *spin image*, has been further extended and applied to texture representation in [LSP03]. It is one of the descriptor-types used within this work.

In [ZW94], an approach which is robust to illumination changes has been proposed, relying on histograms of ordering and reciprocal relations between pixel intensities. The binary relations between intensities of several neighboring pixels are encoded by binary strings and a distribution of all possible combinations is represented by appropriate histograms [MS05].

In [Low04], the *scale invariant feature transform* has been proposed, which represents the affine-normalized intensity values within the enclosing elliptic support area around a feature position by a three-dimensional histogram of gradient locations and orientations. From the latter, a descriptive vector is finally assembled. Contributions to histogram bins are further weighted by the respective gradient magnitudes. To obtain a degree of illumination invariance, the vector is additionally normalized by the square root of the sum of its squared elements. This descriptor is the second type used within this work in addition to the above-mentioned spin-images.

Similar ideas have been exploited in [ATRB95] and in [BMP02]. Both *geometric histograms* and *shape context* compute a histogram which describes the distribution of edges within a region.

The general advantage of histogram-based approaches is the reproducibility of the resulting descriptors: The euclidean distance between descriptors can be directly computed, which renders unnecessary the fragile estimation of an adequate and representative covariance matrix for the equalization of descriptor dimensions. Also, representatives of this class of descriptors have shown a superior performance in comparative studies such as in [MS05] or [MP07].

In the following sections, two histogram-based approaches are described in more detail. Both are used for rotation-invariant region description within this work.

2.2.3.2 Scale-invariant Feature Transform

In [Low04], a combined region detector and descriptor has been proposed. The detection of candidate regions is performed similar to the Hessian-affine detector introduced in section 2.2.2.2. A candidate region is found, if the LoG-operator according to equation 2.12 attains a maximum in both image plane and scale space [Lin98]. The latter is computed by successive smoothing of the original image with a Gaussian convolution kernel. However, the resulting regions are of circular shape and thus not affine-covariant. Therefore, only the descriptor stage of SIFT is used within this work.

In the first step of descriptor computation, a consistent orientation is assigned to each region based on the distribution of gradients within. The general idea is to compute the descriptor relative to this dominant orientation, thus making it invariant to image rotations. For each intensity value $I(x, y)$ within the region, the gradient magnitude $m(x, y)$ and the orientation $\phi(x, y)$ are given as

$$\begin{aligned} m(x,y) &= \sqrt{(I(x+1,y)-I(x-1,y))^2+\cdots} \\ &\qquad \overline{\cdots (I(x,y+1)-I(x,y-1))^2} \qquad (2.18) \\ \phi(x,y) &= \arctan\left(\frac{I(x,y+1)-I(x,y-1)}{I(x+1,y)-I(x-1,y)}\right). \qquad (2.19) \end{aligned}$$

From sample points within the region, a histogram of gradient orientations is then

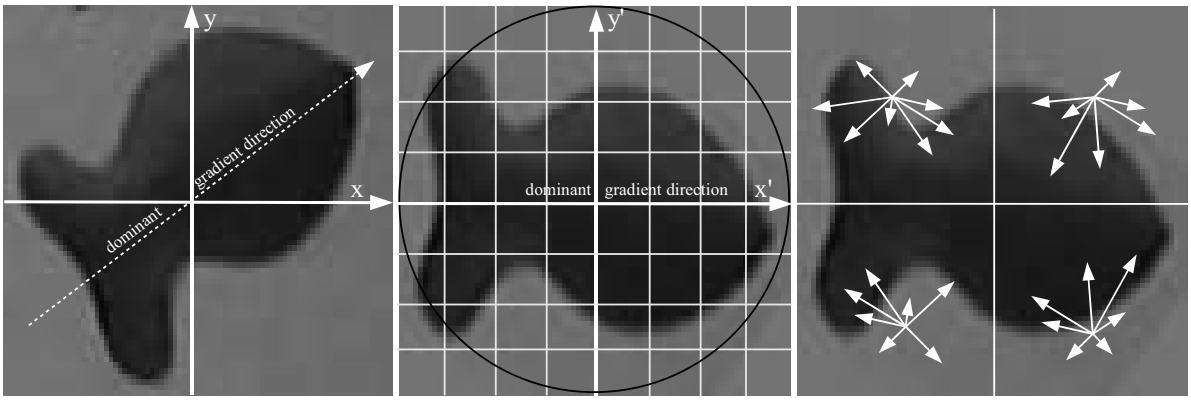


Figure 2.7: The dashed line indicates the dominant orientation of an example region (*left*), which serves as the new major axis of a relational coordinate system (*middle*). The figure shows an 8×8-set of sample clusters, in which the locally dominant orientations are computed according to equation 2.19. The set of 64 clusters is again subdivided into a smaller 2×2-set (*right*) each of which is assigned 8 average orientations computed from the 16 clusters within (note that the depicted vectors are solely for illustrative purposes and do not correspond to the real gradients).

computed. In the original publication, the latter consisted of 36 bins with a coverage of 360 degrees. Each histogram entry is further weighed with the corresponding $m(x, y)$ and the response of a 2D-Gaussian probability density function positioned at the region center. The purpose of the Gaussian is to avoid sudden changes in the descriptor caused by small changes in the region position, and to give less emphasis to gradients far from the center [Low04]. The dominant orientation of the region is then found by detecting the maximum bin in the histogram. Figure 2.7 (*left*) shows the dominant orientation for an example region. In the middle, the region has been rotated into the new (relative) coordinate system. The figure shows an 8×8 set of sample clusters, in which the dominant orientations are computed according to equation 2.19. The set of 64 clusters is again subdivided into a smaller 2×2-set (*right*) each of which is assigned 8 average orientations computed from the 16 clusters within (note that the depicted vectors are solely for illustrative purposes and do not correspond to the real gradients). The original SIFT-descriptor uses an array of 16×16 clusters, subdivided into a 4×4-set. This leads to a descriptive vector of dimension 128.

In order to avoid boundary effects, where a specific sample shifts from one histogram to another, each bin entry is further multiplied by a weight $1 - d$, where d is the distance to the center of the respective local histogram. Finally, illumination invariance is ensured by normalizing the descriptor by the square root of the sum of its squared components.

2.2.3.3 Spin-Images

In [LSP03], a histogram-based rotation-invariant descriptor has been proposed in the context of sparse texture representation. Inspired by the original work of Johnson [JH99], where a similar method was used for object recognition in cluttered 3D-scenes, a two-dimensional histogram is introduced, whose dimensions represent the distance d from the respective region center and the normalized intensity value i. Since both d and i are invariant to orthogonal transformations of the corresponding region, they are well-suited for the representation of affine-normalized patches, where the last remaining degree of freedom is a two-dimensional rotation.

In order to achieve further invariance to variations of the image intensity of the form $I \rightarrow aI + b$, a normalization is performed according to

$$I_n = \frac{I - \min(I)}{\max(I) - \min(I)} \,. \tag{2.20}$$

To avoid aliasing effects, each pixel within the support area contributes to more than a single histogram bin [KVD99]. For a specific bin with histogram coordinates (d_0, i_0), the contribution p of a pixel at distance d from the center and with intensity i is

$$p = \exp\left(-\frac{(d-d_0)^2}{2\alpha^2} - \frac{(i-i_o)^2}{2\beta^2}\right), \tag{2.21}$$

where the parameters α and β control the degree of smoothing of the histogram. Both parameters have a direct influence on the information content of the descriptor: if smoothing is chosen too strong, small variations of the region content will remain undetected.

In figure 2.8, the concept of spin-image construction is illustrated. The two circles denote the position of pixels with similar distance to the region center: In the 2D-histogram, they will be located in the same column. Pixels with bright luminance will contribute to a bin on the right side of the normalized intensity axis while dark pixels (as for example the center pixel) will be located on its left side. As can be seen, the histogram contains two dominant clusters, one with high intensity, the second with low intensity. This is consistent with the associated region (*left*), which mainly consists of two dominant gray values. The similarity between two spin-images can be efficiently computed using zero-mean normalized cross-correlation (ZNCC). For better comparability to other methods (such as SIFT), the resulting correlation coefficient can be easily related to the euclidean distance.

A major advantage of spin-images is their comparatively low computational complexity. However, the ordering of intensity values in the original region is lost in the process. In figure 2.9 for example, the spin-images associated with regions 1 and 4 are very similar (with a correlation coefficient of ZNCC $= 0.84$), although the region content is clearly different.

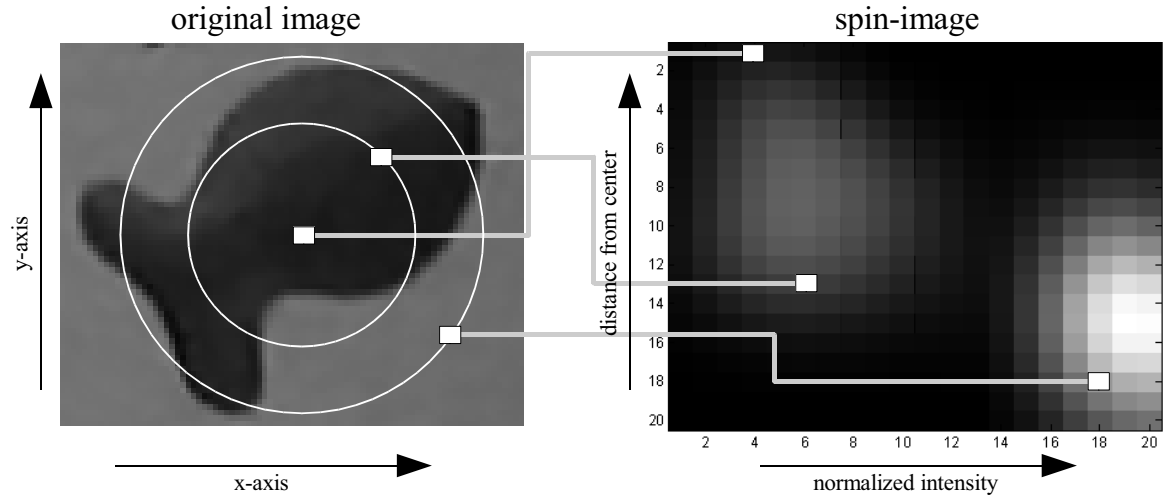


Figure 2.8: Concept of spin-image construction: A pixel in the original image (left) is mapped into a two-dimensional histogram of center-distance and normalized intensity (right). White circles in the original image denote the position of pixels with the same row index in the histogram (=spin-image).

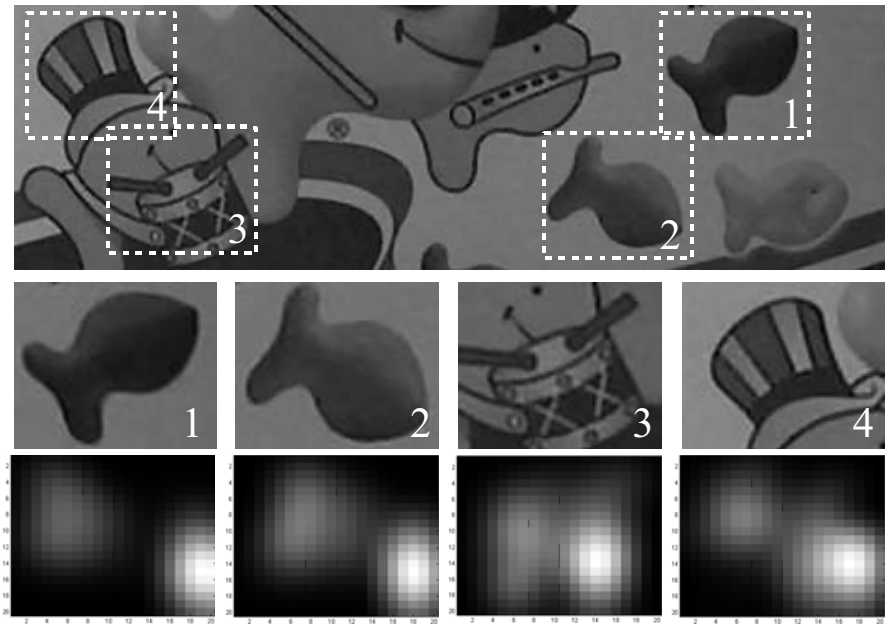


Figure 2.9: Examples of spin-images: Four different regions are extracted from the first frame of the *carton*-sequence and transformed into spin-images (bottom row).

In [MS05], spin-images have shown inferior to SIFT-descriptors with regard to their descriptive power in a number of experiments. In a direct comparison, SIFT may be described as a strong but expensive descriptor, while spin-images on the other hand are significantly less distinctive but at the same time much cheaper with regard to their computational complexity. This might prove beneficial for applications with limited hardware or real-time constraints. For such cases, the use of SPIN instead of SIFT can be advantageous.

2.2.4 Homography Estimation From Region Correspondences

In order to assess the accuracy of a set of regions in one image, the most likely correspondences among the set of regions within the next image have to be identified. If the object surfaces on which the two sets lie are planar, this can be achieved by means of inter-image homographies, which transforms both sets onto each other. Based on the resulting overlap between original and transformed regions, candidate correspondences may then be initially assigned to each other.

In this section, a method for the estimation of a homography between two projective planes from point-to-point correspondences is presented. The problem can be formulated as follows: Given a set of image points in homogeneous coordinates $\mathbf{x}_i = (x_i, y_i, 1)^T,\ x_i, y_i \in \mathbb{R}$ and a set of corresponding points $\mathbf{x}'_i = (x'_i, y'_i, 1)^T$, estimate the transformation H that relates the two sets such that

$$\mathbf{x}'_i = H\mathbf{x}_i \ . \tag{2.22}$$

Both $\mathbf{x}_i$ and $\mathbf{x}'_i$ are located in an image, each of which is being treated as a projective plane $\mathbb{P}^2$. Since both point sets are given in homogeneous coordinates, they are equal in direction but may differ in magnitude by a non-zero scale-factor.

Direct linear transformation (DLT) using point correspondences. The algorithm described in this section is the *Direct Linear Transformation (DLT)*, presented in great detail in [HZ03] and in [AJN05].

Firstly, a straight-forward method for determining H given a set of 4 correspondences of the form $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ is presented: Equation 2.22 may also be expressed using the vector cross-product as $\mathbf{x}_i \times (H\mathbf{x}_i) = \mathbf{0}$. This formulation will enable the derivation of a simple linear solution for H. By referring to the j-th row of H as h^{jT}, the right term of the cross-product can be rewritten as

$$H\mathbf{x}_i = \begin{pmatrix} h^{1T}\mathbf{x}_i \\ h^{2T}\mathbf{x}_i \\ h^{3T}\mathbf{x}_i \end{pmatrix} \ . \tag{2.23}$$

Defining $\mathbf{x}'_i = (x'_i, y'_i, w'_i)$ gives the cross-product explicitly as

$$\mathbf{x}_i \times (H\mathbf{x}_i) = \begin{pmatrix} y'_i h^{3T}\mathbf{x}_i - w'_i h^{2T}\mathbf{x}_i \\ w'_i h^{1T}\mathbf{x}_i - x'_i h^{3T}\mathbf{x}_i \\ x'_i h^{2T}\mathbf{x}_i - y'_i h^{1T}\mathbf{x}_i \end{pmatrix} = \mathbf{0}\,. \tag{2.24}$$

This may be rewritten as

$$\begin{bmatrix} \mathbf{0}^T & -w'_i\mathbf{x}_i^T & y'_i\mathbf{x}_i^T \\ w'_i\mathbf{x}_i^T & \mathbf{0}^T & -x'_i\mathbf{x}_i^T \\ -y'_i\mathbf{x}_i^T & x'_i\mathbf{x}_i^T & \mathbf{0}^T \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}\,. \tag{2.25}$$

The resulting equations have the form $A_i\mathbf{h} = \mathbf{0}$, where every A_i is a 3×9 matrix and $\mathbf{h}$ is a 9-vector which consists of the elements of H such that

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{bmatrix}, \qquad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}. \tag{2.26}$$

Although 2.25 consists of three equations, only two of them are linearly independent. Thus, each point correspondence provides only two equations in the entries of H [HZ03]. Therefore, 2.25 can be rewritten as

$$\begin{bmatrix} \mathbf{0}^T & -w'_i\mathbf{x}_i^T & y'_i\mathbf{x}_i^T \\ w'_i\mathbf{x}_i^T & \mathbf{0}^T & -x'_i\mathbf{x}_i^T \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}\,. \tag{2.27}$$

Given a set of four point correspondences from a plane, a set of equations is obtained of the form $A\mathbf{h} = 0$, where the matrix A consists of the concatenated rows of A_i for each correspondence and $\mathbf{h}$ is the vector of the unknown entries of H. Given four point correspondences, an exact solution can be found for the parameters of H.

In practice, equation 2.22 is never exactly satisfied, due to noise. Therefore, assuming that $\mathbf{x}'_i$ is corrupted by zero-mean Gaussian noise with covariance matrix $\Lambda_{\mathbf{x}'}$, the maximum-likelihood estimation of H is obtained by minimization of the functional

$$J = \sum (\mathbf{x}'_i - \hat{\mathbf{x}}'_i)^T \Lambda_{\mathbf{x}'}^{-1} (\mathbf{x}'_i - \hat{\mathbf{x}}'_i), \qquad \text{where} \quad \hat{\mathbf{x}}'_i = \frac{1}{\mathbf{h}^{3T}\mathbf{x}_i} \begin{bmatrix} \mathbf{h}^{1T}\mathbf{x}_i \\ \mathbf{h}^{2T}\mathbf{x}_i \end{bmatrix}. \tag{2.28}$$

This is a nonlinear least-squares estimation problem, where the parameters of H are sought such that $\|\mathbf{x}'_i - \hat{\mathbf{x}}'_i\|$ attains a minimum. For $A\mathbf{h} = 0$, a non-zero solution is sought, that minimizes a suitable cost function subject to the constraint $\|\mathbf{h}\| = 1$. This is identical to the problem of finding the minimum of the quotient $\|A\mathbf{h}\| \backslash \|\mathbf{h}\|$. The solution is the unit eigenvector of $A^T A$ corresponding to the smallest eigenvalue [AJN05].

Algorithm 1: Direct linear transformation (DLT) on point correspondences

Objective:
Given a set of two-dimensional point correspondences of the form $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, determine a 2D-homography matrix H such that $\mathbf{x}'_i = H\mathbf{x}_i$.

Algorithm:

1. For each correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, compute A_i. Usually, only the first two rows are needed.
2. Assemble each 2×9 matrix A_i into a single $2n \times 9$ matrix A.
3. Obtain a singular value decomposition (SVD) of A as UDV^T. The solution for $\mathbf{h}$ is the last column of V.
4. Determine H from $\mathbf{h}$.

Table 2.2: Estimation of homographies using the DLT-algorithm.

Hartley and Zisserman point out that normalization of A is essential: Due to different coordinate centers or the transformation of image coordinates by similarities, affinities or even projective transforms, A may be poorly conditioned numerically. Thus, prior to running the estimation algorithm, data normalization should be performed. For brevity, further details have been omitted here (see [HZ03] for information on this subject). In table 2.2, a short summary of the DLT-algorithm is given.

Robust estimation using Random Sample Consensus (RANSAC). The above-described DLT-algorithm assumes, that the errors in the point-correspondences are purely Gaussian. In practice, this is not always the case because many points are mismatched, making them *outliers* to the normal distribution. These can severely affect the estimation process and eventually lead to gross errors. Thus, a mechanism is needed for the identification and removal of such outlier correspondences. This procedure is generally referred to as *robust estimation*. In the literature, there exists a multitude of techniques, mostly differing with respect to their assumptions on the expected proportion and type of outliers. An exhaustive overview can be found in [Zha97]. One method - termed RANdom SAmple Consensus (RANSAC), introduced in [FB81] - has been successfully applied to a wide variety of computer vision applications. The general idea of the algorithm is to firstly estimate a model (such as the homography estimation described in the previous section) from the minimal number of measurements necessary to determine all parameters (which is a closed-form solution). Secondly, the set of *inliers* is sought

Algorithm 2: Robust estimation of a homography using RANSAC

Objective:
Given a set of two-dimensional point correspondences of the form $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, robustly determine the set of inliers to a 2D-homography matrix H such that $\mathbf{x}'_i = H\mathbf{x}_i$.

Algorithm:

1. Select four point correspondences from the dataset and compute a closed-form solution to H.
2. Find all correspondences that comply to H, such that $\|H\mathbf{x}_i - \mathbf{x}'_i\| \leq t$, where t is an appropriate distance threshold on the euclidean distance between both points.
3. Repeat steps 1 and 2 N times, until a sufficient part of the correspondence set is compliant to H.
4. Refine the homography estimate using the DLT-algorithm from table 2.2.

Table 2.3: Robust homography estimation using RANSAC and the DLT-algorithm.

from the entire measurement set, that corresponds to the respective model (given a certain fault tolerance). This process is repeated N times and the model with the largest set of inliers is selected as the final estimate. In the case of homography estimation, the set of inliers to the RANSAC-estimate H_0 is used to obtain a refined estimation, using the DLT-algorithm described in the previous section. For more details on the number of trials N, the appropriate number of samples and a suitable distance measure, refer to section 2.5.1. There, the estimation of homographies based on the image sequences used within this work is discussed in detail. Table 2.3 summarizes the main steps of RANSAC-based homography estimation.

2.3 Measurement Setup and Image Datasets

The image sequences used within this work have been selected from a public database[1], provided by Pierre Moreels and Pietro Perona from the *California Institute of Technology* (CalTech). They are the basis for all subsequently performed experiments and evaluations. Due to the free availability of the image data, the presented results exhibit a high degree of transparency in terms of comparability and reproducibility.

The measurement setup consists of two digital cameras, an automated turn-table and

[1] `http://www.vision.caltech.edu/pmoreels/Datasets/TurntableObjects/`

Figure 2.10: Measurement setup: An example object is placed on a turn-table, illuminated by two overhead photographic lights. The scene is observed by a stereo-camera system consisting of two 3-MPixel cameras. Between every two frames, the table is rotated by $\phi = 5°$. The given information on the measurement setup and the above image have been taken from [MP07].

two photographic overhead-lights, which provide a constant and largely homogeneous scene illumination. Objects of all kinds (cheese-cracker cartons, DVDs, picture frames, etc.) have been positioned on the turn-table and automatically rotated by a fixed angle between two frames. Figure 2.10 shows a photograph of the measurement setup used by Moreels and Perona.

One of the major goals of this chapter is the assessment of region detector accuracy in terms of center position and shape of the surrounding support area. Given a single two-dimensional region within an image however, it can not be determined if either its position or the support area is in sufficient accordance with the underlying surface texture. The only way to gauge the accuracy of a region is to consider it in conjunction with its most likely correspondence in another frame. Thus, if the transformation between object surfaces in temporally adjacent frames were known, the deviation of a given region from its correspondence could be used to determine errors in both center and support area. To this purpose, only piecewise planar objects have been selected from the database, as they allow for the estimation of inter-image homographies between surfaces in neighboring frames. These may then be used to transform a set of regions from one frame to the next. In section 2.5.2, this concept is elaborated in more detail.

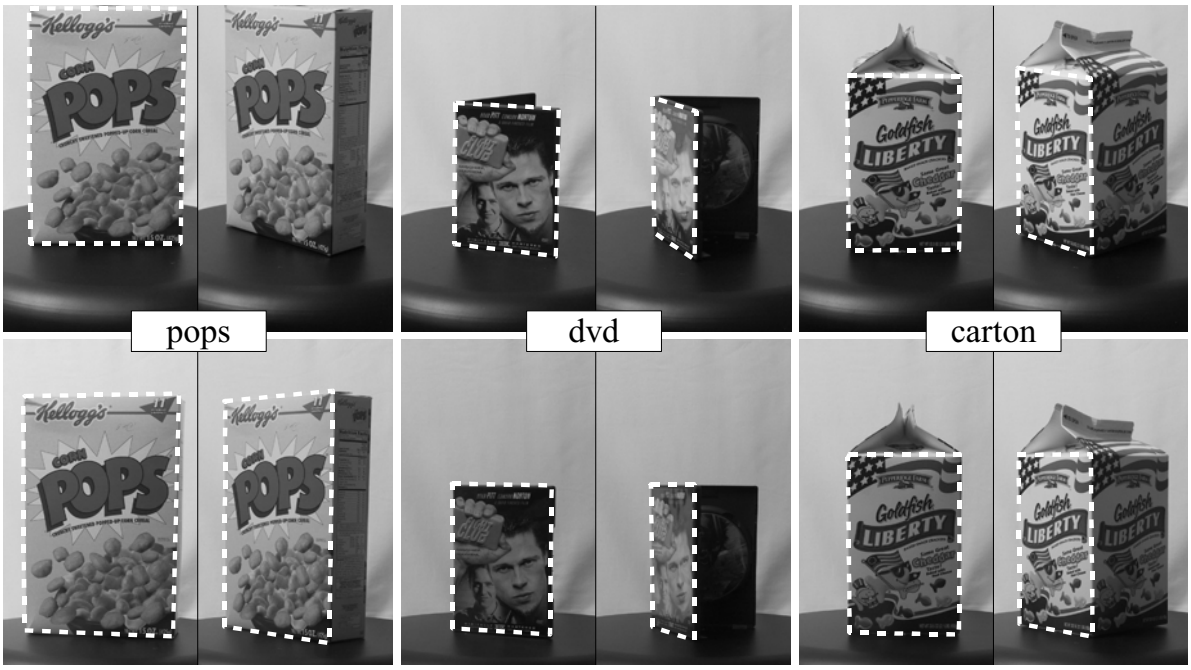


Figure 2.11: First and last (rectified) image of the 3 evaluated stereo sequences. The top row shows the *top camera* view, the bottom row shows the *bottom camera* view accordingly. The turn-table rotation angle between both images amounts to $\phi = 45°$.

As seen in figure 2.10, all available objects have been positioned on an automated turn-table with a resolution of $\frac{1}{50}°$. Between neighboring frames, the turn-table rotation angle amounts to $\phi = 5°$. According to the available setup description, two digital cameras were positioned in front of the turn-table such that the angle between object and each camera was at approximately $10°$. The chosen camera models are *Canon Powershot G1* with a resolution of ≈ 3 MPixel. The focal length has been set to the widest available setting of $f_c = 14.6\ mm$ in order to minimize pincushion distortion (which is approximately at 0.5% for this setting). The baseline width between the centers of both cameras has been estimated to $b \approx 185\ mm$, while the approximate distance to the center of the turn-table is at $\approx 1200\ mm$. Scene illumination is provided by two overhead photographic lights and (according to the measurement descriptions) with activated ceiling lights.

From the entire set of available objects in the database, three have been selected for the underlying work: In order to provide a basis for the evaluation of detector accuracy as motivated above, only objects with partly planar surfaces may be used. For such surfaces, the geometric transformation between two frames of the same sequence can be described using a homography. Within the entire database, three suitable objects with sufficiently large planar surfaces could be found: A cheese-cracker carton (*carton*), the

DVD-cover of a major motion picture (*dvd*) and a carton with breakfast cereals (*pops*). Figure 2.11 shows the first and last frame of the image sets for both *top camera* and *bottom camera*. All 3 sequences consist of 10 equal-sized images of 864 x 1444 pel for each camera. With reference to the first frame (which shows the objects in roughly frontoparallel position to the image planes), the investigated turn-table rotation is between $0° \leq \phi \leq 45°$. The estimation of the above-mentioned inter-image homographies is motivated and discussed within chapter 2.

The object surfaces mainly show homogeneous regions with distinctive edges and repeating artificial patterns, but also natural textures such as the cereals in the *pops*-sequence or the faces in the *dvd* -sequence. A categorization of textures into *e.g. structured* and *textured* as in [MTS$^+$05] or [HJA08b] has not been made due to the limited availability of suitable objects in the database.

2.4 Camera Calibration Results

In order to mitigate the effects of lens distortion and to enable the reconstruction of three-dimensional feature points, Moreels and Perona additionally provided a set of images of a planar checkerboard pattern in different poses. Based on a freely-available toolbox, both internal and external parameters of the two cameras could thus be estimated within this thesis. The necessary techniques and the calibration results are presented in the following.

The calibration of the stereo-rig shown in figure 2.10 was performed using the method described in section 2.2.1.2. The technique is based on the evaluation of images from a planar checkerboard pattern, observed under multiple orientations. Without prior knowledge on motion, either the pattern or the cameras can be moved around the scene almost arbitrarily. Figure 2.12 shows the image set for the bottom camera in different positions on the turn-table. Given the results in [Zha99], it can be expected that 20 images are sufficient to allow for a stable estimation. From the different views of the calibration pattern, it can be expected that the estimation errors at the image borders will be higher than within the center. As the calibration images have been provided 'as is', no influence on the distribution of checkerboard corners could be taken. However, the influence on the evaluation results will be negligible, as within the object observation area (on top of the turn-table) the density of corners is sufficiently large.

2.4.1 Single-Camera Calibration

Figure 2.13 shows the flow of the estimation procedure, which consists of four general steps. After the acquisition of a set of images *(a)* of the checkerboard pattern in different orientations, the exact location of the grid corners has to be determined. The pattern consists of 8 rectangles in x-direction and of 12 rectangles in y-direction, leading to a total of 117 potential corners per image. In order to successfully assign each detected

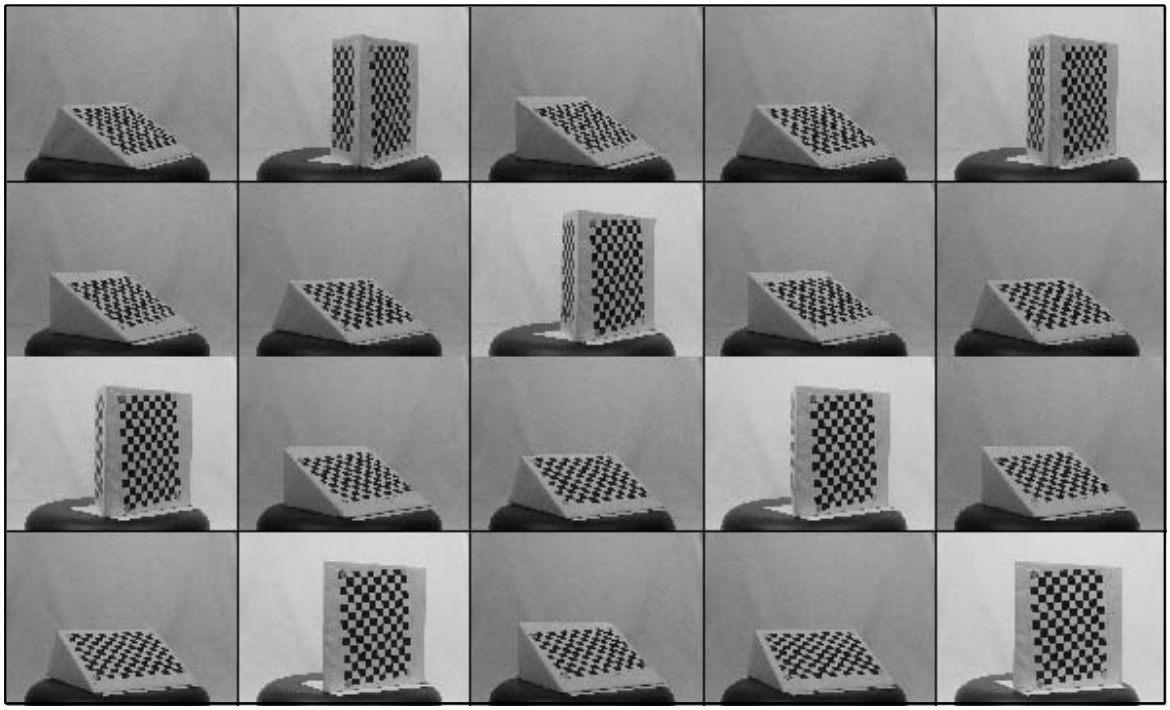


Figure 2.12: Camera calibration images (bottom camera): A checkerboard pattern is positioned on top of the turn-table in different views. From the location of the grid-corners, both external and internal camera parameters are estimated according to the method described in section 2.2.1.2.

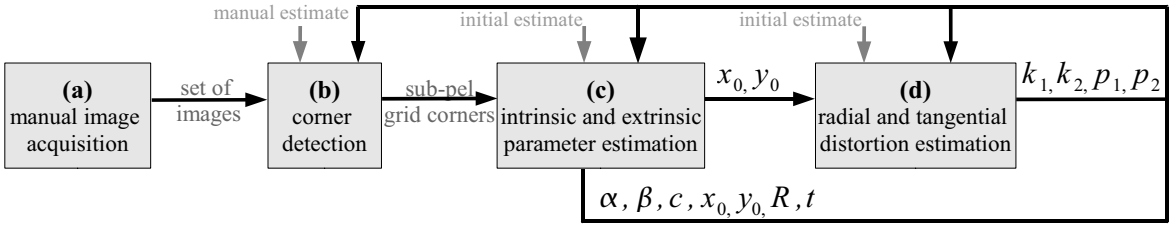


Figure 2.13: Algorithm flow: The calibration procedure described in section 2.2.1.2 consists of 4 successive steps. While image acquisition (a) is performed only once, the remaining steps (b-d) are processed iteratively, until a stable estimate of all camera parameters is reached.

corner to the correct rectangle in the local pattern coordinate system, both its origin and axes orientation are needed as illustrated in figure 2.2 (left). During the first run of the algorithm, origin and orientation of the coordinate system are found by manually marking the pattern perimeter in each image. From the known number of rectangles in each coordinate direction, a rough estimate on potential corner locations can be obtained by the above-mentioned calibration toolbox.

In a local neighborhood around each estimated location, corner detection *(b)* is then performed by evaluating the second moment matrix μ, which is computed for every pixel according to equation 2.1. A potential corner has been found, if both eigenvalues of μ are significant, indicating a two-dimensional variation of the image gradients. Based on the eigenvalues, the *cornerness* is computed according to equation 2.2 (which is the well-known Harris-detector described in [HS88]). The final corner locations are found by performing a non-maximum suppression on the response image in order to locate the integer position with locally maximal cornerness and by subsequently interpolating it to sub-pixel accuracy using a parabolic fit.

After corner extraction, parameter estimation *(c-d)* is performed in two steps. Firstly, an initial estimate on the intrinsic and extrinsic parameters is obtained by computing a closed-form solution, which does not yet include the distortion models (equations 2.3 - 2.5). Secondly, the total reprojection error (in a least-squares sense) is minimized in a non-linear optimization procedure according to equation 2.6. The skew coefficient c in equation 2.3 is set to constant zero during the entire estimation procedure, which implies an angle of $90°$ between x- and y-axis. In most practical situations, this is a justified assumption.

Due to lens distortion effects or coarse manual estimation, the predicted corner locations sometimes are not close enough to the real corners of the pattern and are thus missed during detection or mis-assigned to the wrong rectangle. Figure 2.14 shows the reprojection of the estimated corner locations onto the image plane after the first iteration of the algorithm (left). It can be seen, that for some corners a significant reprojection error in the dimension of several pixels exists. Thus, in order to improve the calibration

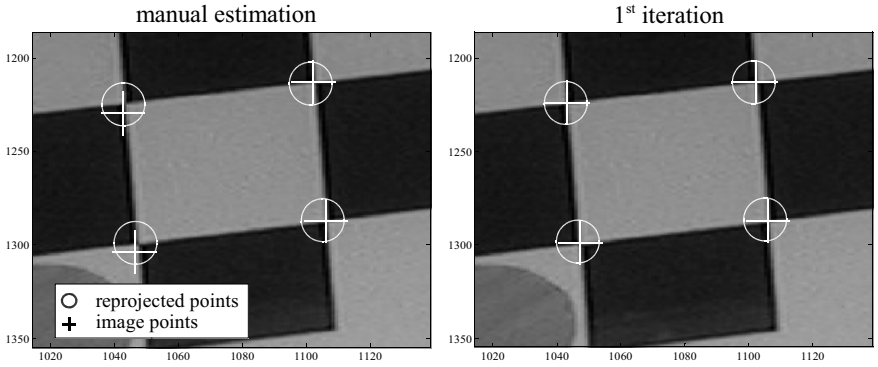


Figure 2.14: The output of the first parameter estimation run is used to obtain a refined estimate of the corner positions (right), which should be superior to the manually estimated positions (left). It can be seen, that the reprojected corners in the right image are much closer to the true corner locations than in the left image.

Table 2.4: Internal parameters for the single camera case: The uncertainty for each estimate represents three times the standard deviation.

single-camera estimation	bottom-camera	top-camera
focal length f_c [pel]	$[4254.16 \quad 4294.15] \pm [25.67 \quad 25.74]$	$[4215.01 \quad 4248.66] \pm [24.10 \quad 27.49]$
principal point x_0, y_0 [pel]	$[1094.20 \quad 610.70] \pm [66.64 \quad 57.94]$	$[1097.50 \quad 615.82] \pm [65.96 \quad 57.46]$
distortion (radial) k_1, k_2	$[-0.2201 \quad 4.8446] \pm [0.1149 \quad 4.5301]$	$[-0.1697 \quad 3.4572] \pm [0.0830 \quad 1.8808]$
distortion (tangential) p_1, p_2	$[0.0052 \quad 0.0007] \pm [0.0016 \quad 0.0017]$	$[-0.0040 \quad 0.0013] \pm [0.0017 \quad 0.0024]$
reprojection error [pel]	$[0.577 \quad 0.760]$	$[0.535 \quad 0.713]$

results, the algorithm is run iteratively. As indicated in the flow diagram, the output of the previous estimation is used as a refined estimate of the corner positions, which should be superior to the manually estimated positions. After the first iteration, the reprojected corners in figure 2.14 (right) are much closer to the real corners in the image. The final parameter estimates for both cameras (*top* + *bottom*) are shown in table 2.4.

Although the corner reprojection error is small, most internal parameters show high uncertainties. This is mainly caused by the uneven distribution of the calibration pattern in the field of view of both cameras. As seen in figure 2.12, the pattern has been positioned on the turn-table only, albeit in different poses. For the majority of the image plane, there are thus no measurements contributing to the estimation process. However, it can be expected that the calibration within the observation area is sufficiently accurate and thus does not corrupt the evaluation results in subsequent chapters.

Figure 2.15 shows both radial and tangential distortion for the bottom-camera as a quiver plot. The arrows represent the effective displacement of a pixel induced by lens distortion. The radial component (k_1, k_2) is shown on the left, while the tangential component (p_1, p_2) is given on the right. It can be observed, that points located at the

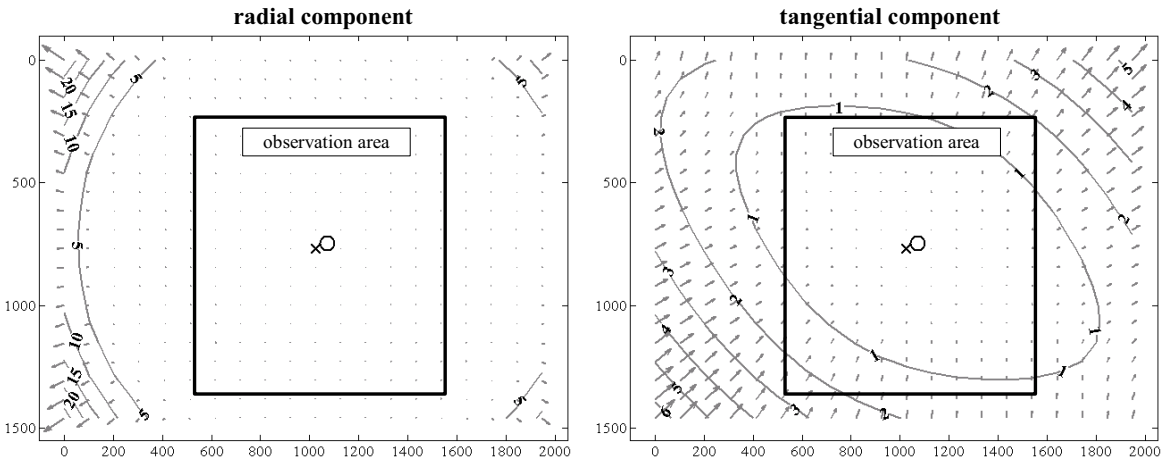


Figure 2.15: Both radial (left) and tangential (right) distortion is shown for the bottom camera. Arrows represent the effective displacement of a pixel induced by lens distortion. Points located at the corners of the image experience a larger displacement than points located in the central part. The cross denotes the center of the image, while the circle indicates the estimated principal point. The solid rectangles mark the area in which the observed objects are located.

corners of the image experience a larger displacement than points within the central part. The solid rectangles mark the area in which the observed objects are located. Within, displacements due to lens distortion stays well below 2 pel.

Once the camera distortion parameters are known, a number of corrective displacement values can be calculated for every pixel position, which transform the original image into an undistorted one. This process is referred to as *rectification* and is applied to all image sequences. All evaluation results in subsequent chapters are based on rectified image data. Figure 2.16 shows both the original and rectified version of the first frame from the *carton*-sequence. It can be seen, that the carton edges in the rectified image (right) are straight, while in the original image (left) there exists a slight distortion.

2.4.2 Stereo-Camera Calibration

Both the evaluation of detector accuracy in chapter 2 and of the tracking methods in chapter 3 are solely performed on monocular image sequences. No explicit use of the stereo camera setup as seen in figure 2.10 is made, and thus in principal the relative orientation of both cameras toward each other is not required. In chapter 4 however, inter-camera matching of region trajectories is performed in order to achieve a spatial reconstruction of the observed objects. To this end, the extrinsic orientation of both cameras is needed.

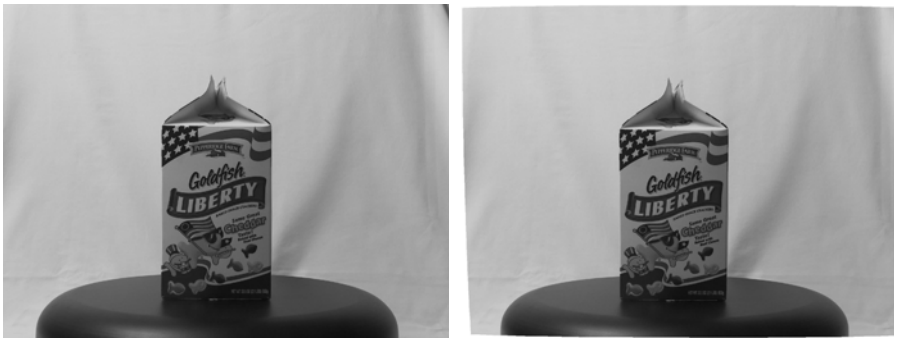


Figure 2.16: For the first frame of the carton-sequence, both the original image (left) and the rectified image (right) are shown. It can be seen, that the carton edges in the rectified image are straight, while in the original image there exists a slight distortion.

Table 2.5: Internal camera parameters after joint stereo estimation: The uncertainty for each estimate represents three times the standard deviation.

stereo estimation	top-camera	bottom-camera
focal length f_c [pel]	$[4285.76 \quad 4294.54] \pm [19.36 \quad 21.07]$	$[4207.55 \quad 4217.81] \pm [19.04 \quad 20.15]$
principal point x_0, y_0 [pel]	$[1091.31 \quad 612.42] \pm [42.24 \quad 39.91]$	$[1095.20 \quad 613.42] \pm [60.92 \quad 49.76]$
distortion (radial) k_1, k_2	$[-0.1728 \quad 3.9588] \pm [0.1080 \quad 4.3737]$	$[-0.1531 \quad 3.1839] \pm [0.0831 \quad 1.9176]$
distortion (tangential) p_1, p_2	$[0.0049 \quad 0.0005] \pm [0.0014 \quad 0.0015]$	$[-0.0045 \quad 0.0023] \pm [0.0013 \quad 0.0023]$

So far, each camera has been processed individually. Now, both of them are considered jointly in a common reference frame. By assuming the bottom-camera as origin of the world coordinate system, both spatial rotation R_t and translation $\mathbf{t_t}$ of the top-camera are estimated relative to the bottom camera. A point $\mathbf{x}_b$ in the bottom camera may be easily transformed into the top camera by applying

$$\mathbf{x}_t = R_t \mathbf{x}_b + \mathbf{t_t} \ . \tag{2.29}$$

The estimation of the intrinsic camera parameters is performed in the same way as during single camera calibration in the previous section, plus the added constraint that the top-camera observes the same scene and thus the same calibration pattern as the bottom-camera. In this way, both cameras contribute jointly to the estimation procedure, which thus has the additional advantage of an increased numerical robustness. As initial parameters, the single-camera estimates are used. Table 2.5 gives the resulting refined parameters, including the respective uncertainties. Obviously, the uncertainties of the intrinsic parameters are smaller, compared to the individual estimation in table 2.4.

Finally, figure 2.17 shows the reconstructed stereo rig based on the estimated external parameters, including the position of the checkerboard pattern for each image.

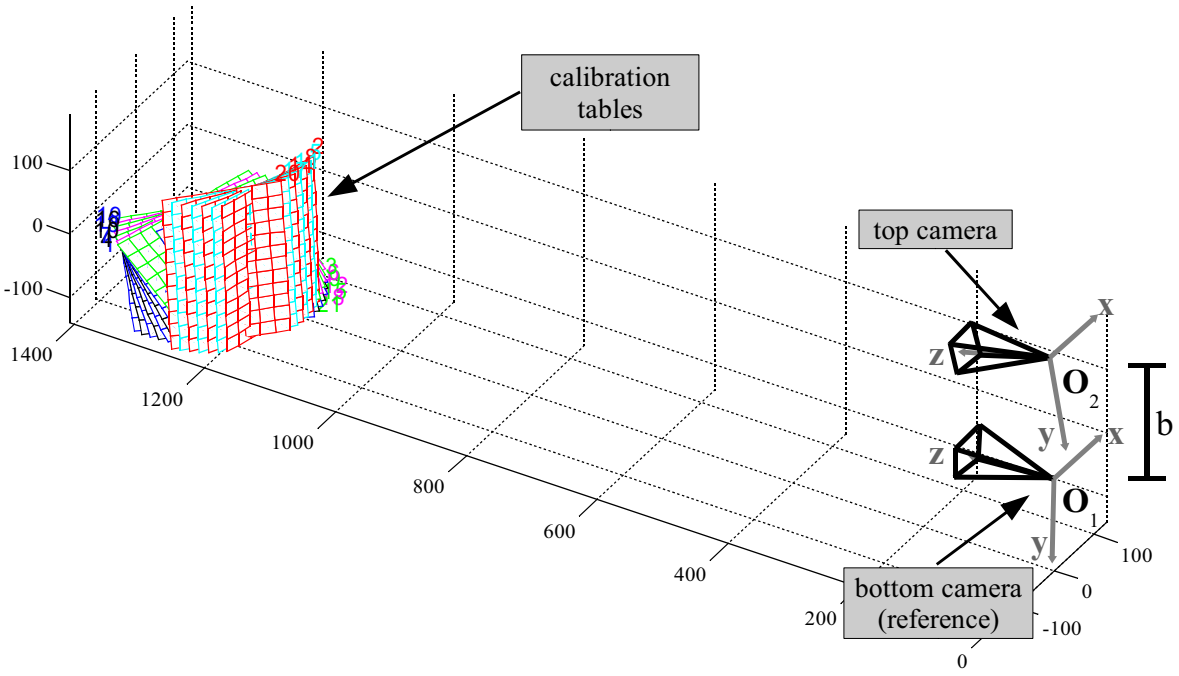


Figure 2.17: Visualization of the estimated stereo setup: Both camera positions O_1 and O_2 are shown, along with the different positions of the checkerboard calibration table. The distance between camera rig and turn-table has been estimated to $\approx$ 1200 mm while the base width of the two cameras is at $b \approx 185\ mm$.

2.5 Evaluation

2.5.1 Homography Estimation Results

As can be seen from figure 2.11, only objects consisting of mainly planar surfaces have been selected from the database. Thus, surfaces between neighboring frames of each image sequence I_i and I_{i+1} are related by a homography $H_{i,i+1}$. In order to make an assessment of region accuracy in terms of spatial localization and accuracy of the support area for each of the five detectors introduced in section 2.2.2.2, $H_{i,i+1}$ has to be estimated for every pair of adjacent frames. To this purpose, the method described in background section 2.2.4 is used. It consists of four general steps, which are illustrated in figure 2.18.

In the first step (a+b), the four corners of the bounding polygonals shown in figure 2.11 (dashed lines) are selected manually for each frame. These are used to compute a closed-form solution for an initial homography H_0. Depending on the quality of manual corner selection, the accuracy of H_0 may not suffice to be used for evaluation purposes. Therefore, a number of affine-covariant regions as well as suitable rotation-invariant descriptors are computed from each image. For this purpose, an implementation of the SIFT-descriptor as introduced in section 2.2.3.2 has been used. The software has been made available as a compiled binary by the original author, using the default parameters of the related publications[1]. In several comparative studies such as [MS05] or [MP07], the SIFT-method has proved superior to other descriptors under considerable changes in both viewpoint and illumination and should thus be well-suited to the evaluated image sequences. Between adjacent frames, region matching on the basis of the associated SIFT-descriptors is then performed, leading to a set of initial correspondences.

In the second step (c), outlier correspondences which do not comply sufficiently to H_0 are removed from the set. Based on the area of intersection of a region r_{i+1} and its transformed correspondence $r'_i = H_0 r_i$, inlier regions may be determined by evaluating the *area overlap error* as defined in [MTS$^+$05]:

$$d_o(r'_i, r_{i+1}) = |1 - \frac{r'_i \cap r_{i+1}}{r'_i \cup r_{i+1}}| \,. \tag{2.30}$$

The area of the union and intersection of the two regions is computed numerically, as a closed-form solution to this particular problem does not exist. For congruent regions, the area overlap error attains $d_o = 0$, while for completely disjunct regions it attains $d_0 = 1$. A second measure which discards all information on region shape but considers only the center position $\mathbf{x}$ is the *position error*

$$d_l(r'_i, r_{i+1}) = \|\mathbf{x}'_i - \mathbf{x}_{i+1}\| \,. \tag{2.31}$$

[1] `http://www.robots.ox.ac.uk/~vgg/research/affine`

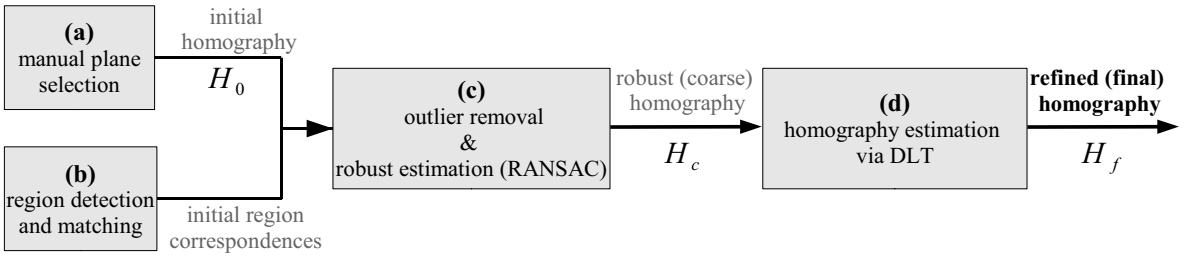


Figure 2.18: Robust estimation method of inter-frame homographies from region correspondences using DLT and RANSAC.

The disadvantage of both measures d_o and d_l is that only the error in the second frame I_{i+1} is considered while regions in frame I_i are assumed to be perfectly localized. One way of constructing a more adequate error function is to consider both forward and backward transformations $H_{i,i+1}$ and $H_{i+1,i} = H_{i,i+1}^{-1}$ and to summarize d_o and d_l for both transformations. The corresponding error measure is termed *symmetric transfer error* in [HZ03] and is defined as

$$d_s(r_i, r_{i+1}) = \sum_i d(r_i, H_{i+1,i} r_{i+1})^2 + d(r_{i+1}, H_{i,i+1} r_i)^2 \ . \tag{2.32}$$

The first term corresponds to the transfer error in the first image, while the second term refers to the transfer error in the second image. In the remainder of this work, each reference to either d_o or d_l automatically includes the symmetric transfer error according to equation 2.32. Apart from homography estimation, both measures will also be used extensively for accuracy assessments within subsequent chapters.

Because no further information on the evaluated objects in figure 2.11 is available, inter-image transformations have to be estimated from the detected regions themselves. As there exists no prior knowledge on the accuracy of each detector yet (which is the goal of this chapter, after all), estimation has been performed for all detectors individually. Also, it has to be taken into account that all subsequent results are directly dependent on the quality of homography estimation and thus on the assumption of planar object surfaces. The latter is violated to some degree by all three objects, as neither the DVD nor the carton sides will be perfectly flat. Thus, comparisons between the detectors should always be seen relative and by no means absolute. For the latter, a very accurate knowledge about the object surfaces and their position within the scene would have been necessary. For the evaluated sequences however, neither of those was available.

In table 2.6, the total number of matched regions and the percentage of compliances to the initial homography H_0 are given for all five detectors. Also, both symmetric area overlap error d_o and the symmetric position error d_l computed according to equation 2.32 are shown. Each table entry has been averaged over all frames of the *carton*-sequence. A complete overview of all sequences has been spared here for the sake

Table 2.6: Robust estimation of the initial homography H_0. Each table entry has been averaged over all frames of the *carton*-sequence.

	EBR	IBR	MSER	HARAFF	HESAFF
$\sum$ matched regions	590	171	189	209	530
% compliant to H_0	89	74	75	68	67
localization accuracy $d_{s,l}$ in [pel]	8.86	7.05	6.09	6.61	6.62
shape accuracy $d_{s,o}$ in [%]	26.24	31.73	28.64	33.67	33.64

of brevity. It can be seen, that the number of correspondences varies greatly between the five detectors. In conjunction with the percentage of compliances to H_0, the EBR-method is clearly superior, followed by HESAFF. For the remaining detectors, the number of compliant matches is similar. With regard to the position error, d_l is below 6 pel for all detectors, while the area overlap error exceeds $d_0 \geq 0.25$. Clearly, H_0 is to coarse an estimate to be used as groundtruth. Thus, a more robust but still coarse homography H_c is estimated from the compliant correspondences using the RANSAC-algorithm as described in section 2.2.4. In order to properly parameterize the algorithm, two questions need to be discussed.

The problem to be solved can be summarized as follows: Given a set of 2D measurements (region correspondences), find the homography H_c which minimizes the symmetric transfer error according to equations 2.31 and 2.32, subject to the condition that none of the region centers deviates by more than t units. *The first question* to be answered is which distance threshold t should be used. One way to choose t would be to set it empirically by looking at the data and the required accuracy of the fit. A second way described in [HZ03] is based on the assumption, that the measurement error obeys a zero-mean Gaussian distribution with standard deviation σ. In this case, d_l is a sum of squared Gaussian variables and follows a χ^2_m-distribution with m degrees of freedom (which is the co-dimension of the model). For a homography holds $m = 2$, as both errors in $x-$ and $y-$direction are measured. The probability that the value of a χ^2_m random variable is less than k^2 is given by the cumulative chi-squared distribution $F_m(k^2) = \int_0^{k^2} \chi^2_m(\xi) d\xi$. From this distribution, inliers may be identified based on the following decision scheme:

$$\begin{cases} \text{inlier} & \text{if } d_l^2 < t^2 \\ \text{outlier} & \text{if } d_l^2 \geq t^2 \end{cases} \qquad \text{with} \quad t^2 = F_m^{-1}(\xi)\sigma^2 \ . \tag{2.33}$$

If ξ were chosen as 0.95, the probability that a point was an inlier would be at 95%. This means that an inlier will only be incorrectly rejected in 5 % of the time. According to the distributions in [HZ03], the squared threshold was $t^2 = 5.99\sigma^2$. In all experiments, the threshold in normalized coordinates was set to $t = 0.005$ and thus $\sigma = 0.002$.

Table 2.7: Refined estimation of the final homography H_f. Each table entry has been averaged over all frames of the *carton*-sequence.

	EBR	IBR	MSER	HARAFF	HESAFF
% compliant to H_c and H_f	65	59	72	55	61
position error d_l in [pel]	2.84	2.28	0.83	2.48	2.62
overlap error d_o in [%]	14.74	17.04	7.45	20.56	19.64

The second question refers to the number of samples N which should be tried. Often, it is infeasible to process the entire dataset. Instead, N is chosen sufficiently high such that with probability p, at least one of the random samples is free from outliers. If w was the probability that a specific sample was an inlier and $u = 1 - w$ was the probability that it was an outlier, then at least N selections (of s points) are required, where $(1 - w^s)^N = 1 - p$, so that

$$N = \frac{\log(1-p)}{\log(1-(1-u)^s)} \; . \tag{2.34}$$

For every detector and every image pair, N is determined automatically under the assumption that $p = 0.98$ and $s_r = 4$.

From the inliers to H_c, a refined estimate based on the *Direct Linear Transform* described in section 2.2.4 is obtained. Table 2.7 gives the respective number of inliers as well as the resulting overlap and localization errors. It can be seen, that for all detectors, the accuracy has been greatly improved after removing model outliers with the RANSAC-method. However, except for MSER, the overall position error is still $d_l \geq 2\ pel$. Therefore, the MSER-detector is chosen for the computation of all inter-frame homographies, which form the basis for subsequent experiments and the entire evaluations within this work. For all sequences, d_l stays below 1 pel for this detector. As mentioned before however, a further error is introduced into all subsequent results due to slight deviations of the object surfaces from perfect planarity. Thus, the evaluation results within this work only allow for a relative comparison of the five affine-covariant detectors. Conclusions on their absolute accuracy are not admissible without further knowledge on the exact object shape and their position within the scene.

2.5.2 Region Performance Measures

Within this section, a selection of affine-covariant region detectors is evaluated and compared with regard to several performance measures. As discussed in section 2.2.2, a region generally consists of a center position $\mathbf{x} = [x, y]^T$ and an elliptic support area around $\mathbf{x}$. Ideally, a good detector should

1. provide a large number of regions, which are detected reliably as long as the corresponding object in world space is seen by the camera.

2. exhibit high accuracy in terms of center position and the shape of the support area.

3. be robust under image transformations, such as perspective distortions.

From the above list, a number of performance measures is derived, which are elaborated and discussed in the following.

Given a single two-dimensional region within an image, it can not be determined if either the center position or the elliptic support area is in sufficient accordance with the underlying surface texture. The only way to determine its accuracy is to consider it in conjunction with its most likely correspondence in another frame. Thus, if the transformation between object surfaces in temporally adjacent frames were known, the deviation of a given region from its correspondence could be used to determine errors in both center and support area. To this purpose, the inter-image homographies from the previous section 2.5.1 are used. Figure 2.19 illustrates the principle. Given a set of regions in frame I_i, transform each one of them into frame I_{i+1} using the respective homography $H_{i,i+1}$ (blue ellipses). Assign to each transformed region the most similar region in frame I_{i+1} (red ellipses) as its likely correspondence. Congruency between two regions is measured using the area overlap error d_o as introduced in equation 2.30. For two regions r_i^l and r_{i+1}^m, congruency is reached for $d_o(r_i^{l'}, r_{i+1}) = 0$, with $r_i^{l'} = H_{i,i+1} r_i^l$. Ellipses with no intersection yield $d_o(r_i^{l'}, r_{i+1}^m) = 1$ instead. Both regions are associated to each other as potential correspondences, if the respective area overlap error is below a threshold, *i.e.* $d_o \leq d_{o,max}$.

If however several regions in frame I_i should claim the same candidate region in frame I_{i+1} or vice versa, ambiguities among the correspondences have to be resolved, such that every region pair is unique and the overall area overlap error d_o is minimal for the respective frame pair. To this purpose, a combinatorial optimization technique has been used, which solves the region assignment problem in polynomial time. The selected method is generally known as the *Hungarian algorithm* and has been originally published by Harold Kuhn in 1955 [Kuh55]. The algorithm models an assignment problem as an $L \times M$ cost-matrix C, where L and M represent the number of regions in frames I_i and I_{i+1}, respectively. Each element $C(l, m)$ of the matrix represents the cost of assigning region r_i^l to region r_{i+1}^m, which is expressed in terms of the area overlap error d_o as described above. In each row and column of C, there exists exactly one element, which belongs to the optimal solution. The Hungarian method finds these optimal elements by rearranging the rows and columns such that the sum of the elements on the main diagonal is globally minimal. A detailed description of the algorithm may be found in [Fra05]. After the successful termination of the optimization procedure, the original set of correspondence candidates has been divided into two subsets. The remaining elements in the rows and columns of C represent the optimal solution in terms of

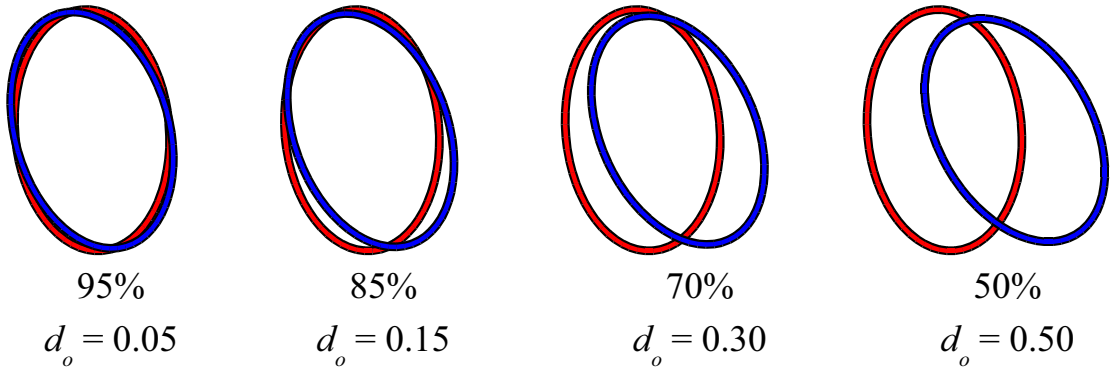


Figure 2.19: Examples of different degrees of support area overlap. The blue regions exist in frame I_{i+1}, while the red regions have been transfered from frame I_i into frame I_{i+1} by using the respective inter-frame homography $H_{i,i+1}$. The given percentages denote the amount of common image area. The corresponding area overlap error d_o according to equation 2.30 is given in the bottom line.

d_o and are denoted as accepted correspondences or *positives* in the following. The correspondences which have been removed from C during the optimization procedure are referred to as rejected correspondences or *negatives* instead. Figure 2.20 illustrates the two types. There, an initial set of five ambiguous correspondence candidates is shown (black arrows, left). As can be seen, two regions in frame I_{i+1} are claimed by several regions in frame I_i. After applying the Hungarian algorithm, the original set has been subdivided into positives and negatives. The latter are removed from the original set, while the former represent a unique assignment solution with minimal area overlap error (green arrows, right).

Although the Hungarian algorithm is comparatively fast, it still has a computational complexity of $O(N^3)$, where N is the number of rows in C. In order to reduce the runtime of the optimization procedure, the original correspondence candidates are further divided into disjunct subsets, which are processed independently of each other. Thus, the complexity of the optimization problem may be reduced significantly. The entire correspondence assignment algorithm is summarized in table 2.8.

In order to obtain a representative set of region correspondences, a specific setting for the maximally permissible area overlap $d_{o,max}$ has to be chosen. Figure 2.21 shows the relative number of positives (left) and negatives (right) as a function of this threshold. It can be seen, that the detectors respond differently to changes in $d_{o,max}$: while the number of positives converges to a clear maximum (saturation level), the number of negatives increases exponentially instead. With respect to the number of positives, the relative ordering of the detectors changes at $d_{o,max} = 0.3$, but essentially stays the same

Algorithm 3: Optimal resolving of region correspondence ambiguities

Objective:

Given a set of two-dimensional region correspondence candidates $\left\{r_i^l \leftrightarrow r_{i+1}^m\right\}$, label each candidate as either accepted (*positive*) or rejected (*negative*) correspondence, such that the resulting set of *positives* is unique and the area overlap error d_o is minimal for each frame pair.

Algorithm:

1. Given two adjacent frames I_i and I_{i+1}, identify the set of potentially corresponding regions $\mathcal{L}$, for which the area overlap error d_o according to equation 2.30 is below a threshold $d_{o,max}$, *i.e.* $\mathcal{L} = \left\{r_i^l \leftrightarrow r_{i+1}^m | d_o(r_i^{l'}, r_{i+1}^m) \leq d_{o,max}\right\}$.

2. Identify interdependent correspondences, which share a common region in the same frame. Based on this information, the set of all correspondences $\mathcal{L}$ is partitioned into subsets, which are disjunct with regard to the regions within.

3. Under the assumption that each image region within a frame I_i is a unique and unambiguous projection of an object in world space, there may exist only a single corresponding region in the next frame I_{i+1}, which is the projection of the same object at a later time instant. In order to meet this constraint, ambiguities have to be resolved in each subset of correspondences. To this purpose, a combinatorial optimization scheme is applied (the Hungarian algorithm), which resolves potential ambiguities and at the same time minimizes the overall d_o within each subset.

Table 2.8: Algorithm 3: Resolving correspondence ambiguities

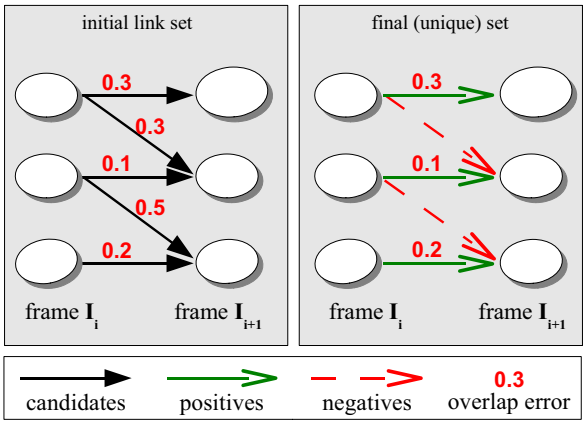


Figure 2.20: Region correspondence assignment: On an initial set of region correspondence candidates (left), a combinatorial optimization procedure is applied, which removes assignment ambiguities such that the overall area overlap error is minimal. Remaining correspondences are denoted as *positives*, while rejected correspondence candidates are called *negatives*.

afterward. For all subsequent experiments, $d_{o,max} = 0.5$ was chosen. For this threshold, the number of positives is almost saturated with every detector, while the number of negatives is still at a moderate level. Also, this choice is consistent with the relevant literature [MTS$^+$05][SMB00]. There, this particular threshold has also been motivated with regard to descriptor-based matching: For an area overlap of 50%, the probability for a correct correspondence assignment by an appropriate region descriptors (*e.g.* SIFT) is still significant.

Obviously, the HESAFF-detector provides the highest relative number of correspondences, followed with distance by HARAFF and EBR. The lowest number of correspondences is detected by MSER, which is only slightly exceeded by the IBR-detector. Table 2.9 additionally shows the absolute number of accepted and rejected correspondences for all detectors and sequences.

From the curve progressions, a first assessment of shape accuracy is possible: With MSER for example, the number of correspondences with an area overlap below $d_o \leq 0.2$ has increased to more than 95% of the respective saturation level, indicating high region accuracy. In the case of HESAFF on the contrary, slightly less than 50% of all correspondences exhibit an area overlap of $d_o \leq 0.2$. Concluding, the most accurate regions should be detected by the MSER-method while HARAFF and HESAFF will probably be least accurate. Later in this section, it will be shown that a more sophisticated evaluation of region accuracy generally coincides with these presumptions.

From the progression of rejected correspondences (right), first conclusions on the density of regions in the images are possible. In order to avoid region ambiguities, it is desirable to have only few negatives, while the number of positives should be high at

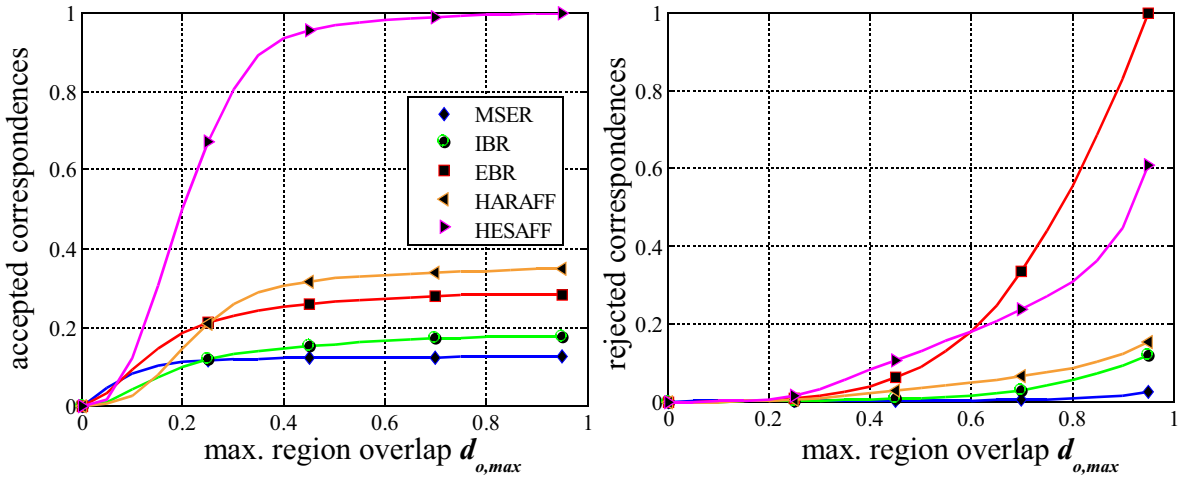


Figure 2.21: Relative number of accepted (left) and rejected (right) correspondences (*positives*). See figure 2.20 and table 2.8 for more details.

		carton	dvd	pops	carton	dvd	pops	$\sum, \varnothing$
		top camera			*bottom camera*			
MSER	$\sum pos.$	2457	1304	1189	2260	882	1091	**9183**
	$\sum neg.$	1040	601	321	746	190	186	**3084**
	$\frac{neg.}{pos.}$	0.42	0.46	0.27	0.33	0.22	0.17	**0.32**
IBR	$\sum pos.$	2296	1081	2745	2067	978	2681	**11848**
	$\sum neg.$	2310	1271	3204	2091	1158	3207	**13241**
	$\frac{neg.}{pos.}$	1.01	1.18	1.17	1.01	1.18	1.20	**1.13**
EBR	$\sum pos.$	6339	1096	3037	5822	880	2779	**19989**
	$\sum neg.$	54513	4634	8556	44088	2906	7625	**122322**
	$\frac{neg.}{pos.}$	8.60	4.23	2.78	7.57	3.30	2.74	**4.87**
HARAFF	$\sum pos.$	6875	1934	4420	6225	980	4036	**24470**
	$\sum neg.$	14191	3578	8213	13152	1724	7585	**48443**
	$\frac{neg.}{pos.}$	2.06	1.85	1.86	2.11	1.76	1.88	**1.92**
HESAFF	$\sum pos.$	15592	6350	15688	15330	4946	15165	**73071**
	$\sum neg.$	38241	16380	36633	36081	12713	34473	**174521**
	$\frac{neg.}{pos.}$	2.45	2.58	2.34	2.35	2.57	2.27	**2.43**

Table 2.9: Number of accepted correspondences (*positives*,top row) and rejected candidates (*negatives*,middle row) for all sequences and detectors at $d_{o,max} = 0.5$. Additionally, the ratio of positives and negatives is given (bottom row).

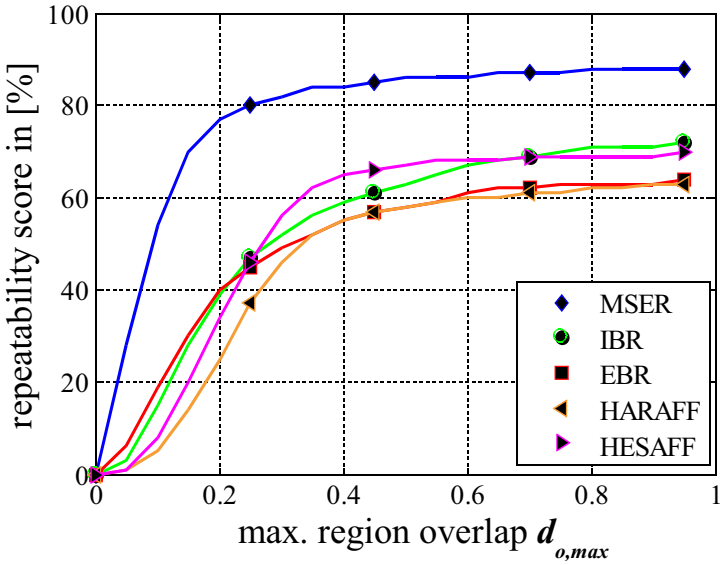


Figure 2.22: Percentage of successfully matched regions (repeatability score) as a function of $d_{o,max}$: A region r_i^l in frame i and its correspondence r_{i+1}^m in frame $i+1$ are deemed 'matched', if $d_o(r_i^{l'}, r_{i+1}^m) \leq d_{o,max}$ according to equation 2.30.

the same time. With EBR for example, the contrary is the case: While the number of positives of this detector is clearly inferior to both HESAFF and HARAFF, the amount of negatives exceeds all other detectors (except HESAFF for $d_o \leq 0.5$). Table 2.9 also gives the ratio of negatives and positives at $d_{o,max} = 0.5$, showing that with EBR, a ratio of ≈ 5 is reached while MSER is at only ≈ 0.3.

Figure 2.22 shows the percentage of successfully matched regions as a function of $d_{o,max}$, according to the definition in [MTS+05]. There, the *repeatability score* for a pair of images is computed as the ratio between the number of region-to-region matches for which holds $d_o(r_i^{l'}, r_{i+1}^m) \leq d_{o,max}$, and the smaller of the number of regions in the pair of images. Here, only regions that are located within the bounding polygonals as illustrated in figure 2.11 are taken into account. It can be seen, that the MSER-detector performs best with a repeatability score of $\approx 85\%$ at $d_{o,max} = 0.5$. The second-best candidate is the HESAFF-detector, with a repeatability score of slightly less than 70% at $d_{o,max} = 0.5$. For both EBR and HARAFF, the repeatability score stays below 65% for all settings of $d_{o,max}$. Table 2.10 additionally gives the number of regions for each detector, as well as the repeatability score for $d_{o,max} = 0.5$.

Table 2.10: Total number of detected regions for each detector on all frames of every sequence (top row) and the respective *repeatability scores* for $d_{o,max} = 0.5$ (bottom row).

	MSER	IBR	EBR	HARAFF	HESAFF
$\sum$ regions	12042	20806	37906	46987	120716
repeatability score	86 %	63 %	58 %	58 %	67 %

2.5.3 Overlap-based Region Correspondences

In this section, an analysis of region accuracy is presented with regard to the position error d_l and to the area overlap error d_o, as introduced in equations 2.30 and 2.31. Based on the estimated homographies between neighboring frames, the given results represent a lower bound on the achievable region accuracy. For real matching applications, homographies are most often not available and correspondence assignment is based on the similarity of region descriptors instead. In the next section, it is shown that the resulting accuracy is lower in that case, depending on the respective combination of detectors and descriptors. As discussed earlier in this section, unique region correspondences between a pair of adjacent frames are found on the basis of a combinatorial optimization method, which globally minimizes the area overlap error. Details on the method can be found in table 2.8.

Also, an analysis of the impact of certain region properties (such as scale, elliptic shape or density within the image) on localization accuracy is analyzed statistically. It will be shown that based on these properties, error-prone regions may be identified (and removed) reliably for most detectors. Thus, the accuracy and computational complexity of a subsequent application (*e.g.* monocular tracking) may be significantly reduced. The major ideas presented in this section have also been published in [HJA08b].

Figure 2.23 shows the distribution of both d_l (left) and d_o (right) over the n-percentile. For both diagrams, the maximally permissible area overlap error has been set to $d_{o,max} = 0.5$ as discussed previously. With regard to d_o, MSER-regions perform best with more than 80 % of all correspondences below $d_o \leq 0.15$. With considerable distance, EBR performs second-best, closely followed by IBR. Notably, differences between HARAFF and HESAFF are very small. For both detectors, almost 60 % of all correspondences show an area overlap error higher than $d_o \geq 0.2$. With regard to the position error d_l, MSER again performs best with almost 95 % of all correspondences below $d_l \leq 2\ pel$. Between HARAFF and HESAFF, differences are negligible: Both curves are almost entirely congruent. Contrary to d_o, their performance now exceeds IBR and EBR. For the latter, position accuracy is above $d_l \geq 5\ pel$ for more than 20 % of all correspondences. Summarizing, the MSER-detector clearly shows the highest accuracy for both d_l and d_o with significant distance to all other detectors.

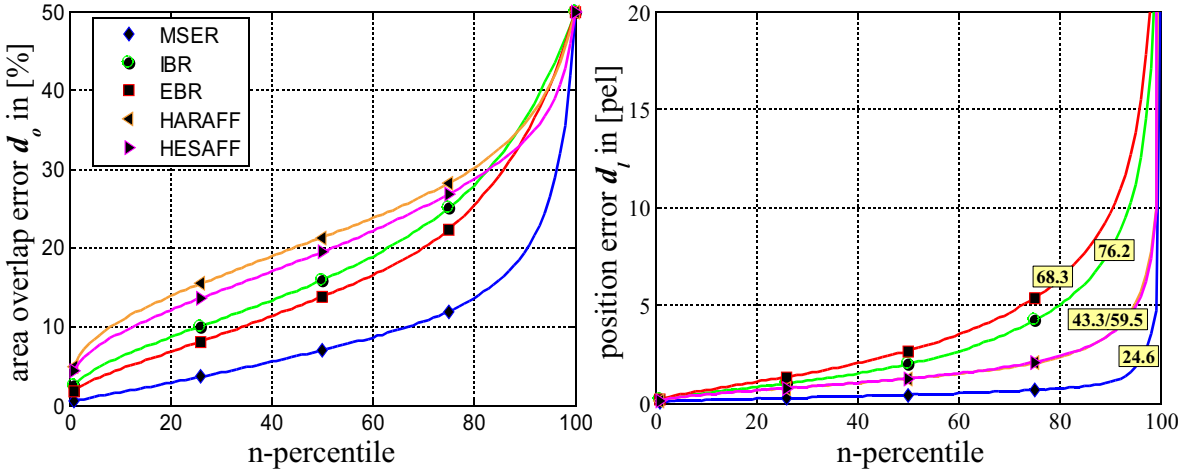


Figure 2.23: Distribution of region accuracy in terms of area overlap error d_o (left) and position error d_l (right). Boxed numbers denote the maximum d_l at the 100-percentile. The 50-percentile is equivalent to the median of the respective distributions.

While the area overlap error is a measure which is independent of region scale, this is not the case for the position error. Considering a small circular region of scale $s_r = 10\ pel$ as illustrated in figure 2.24, the position error for a pre-defined overlap error of $d_o = 0.5$ is at $d_l = 5\ pel$, while for $s_r = 40\ pel$, it reaches up to $d_l = 20\ pel$. Thus, a detector that produces mainly large regions is in principle more prone to larger position errors. For better comparability among different detectors, d_l could also be normalized on regions scale. However, this has not been done in the context of this evaluation, as information on the *true* error can be used more effectively as pre-selection criterion with regard to a specific target application. In figure 2.25 (left), the distribution of region scales is shown for each detector. Again, differences between HARAFF and HESAFF are negligible: For both detectors, 80 % of all regions are smaller than $s_r = 20\ pel$, closely followed by the MSER-detector. The largest regions are detected by EBR, with 50 % of all regions exceeding $s_r \geq 30\ pel$. As expected, the relative ordering of EBR and IBR coincides with the distribution of position accuracy in figure 2.23. However, this is not always the case: Although it detects larger regions, the position error of MSER is significantly lower than for HESAFF. Obviously, the distribution of region scales alone is not a sufficient criterion in order to explain differences in position accuracy among the detectors.

Therefore, a statistical analysis of both position error and area overlap error is given: For every detector, the range of region scales between the 5 %-percentile and the 95 %-percentile has been divided into 20 equally-spaced bins. Within each bin, the median and the data spread in terms of 25 %- and 75 %-percentiles (solid lines) as well as 5 %- and

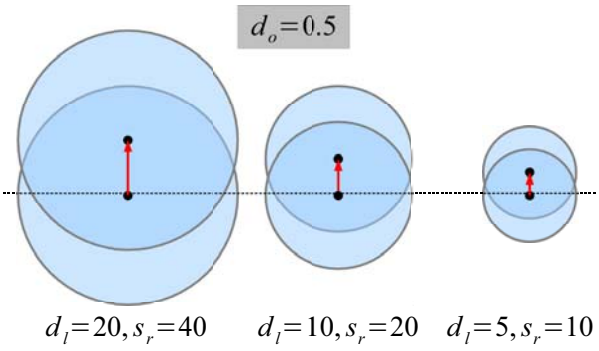


Figure 2.24: Influence of region scale s_r: For the same area overlap error d_o, position errors d_l depend on region scale. Thus, detectors which provide larger regions are principally more prone to higher position errors.

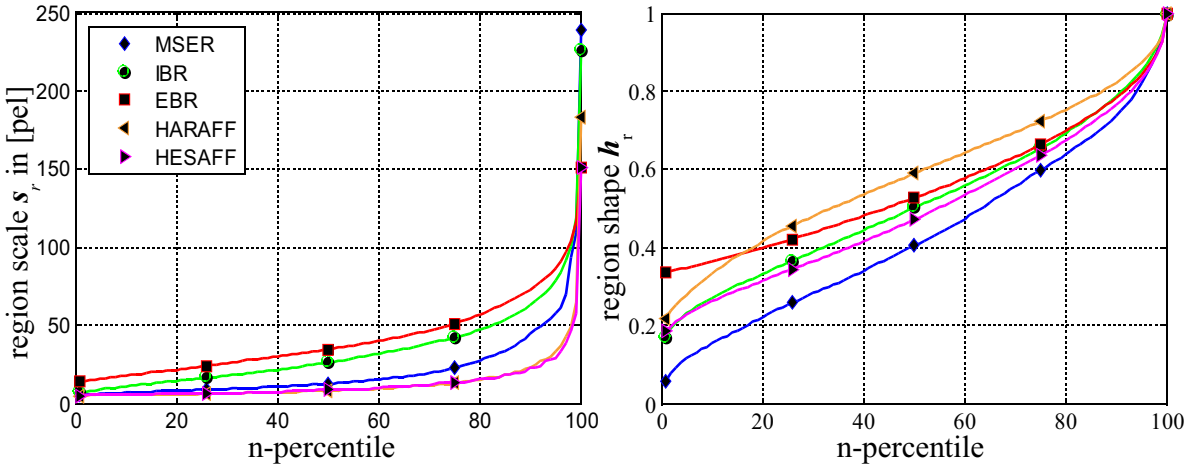


Figure 2.25: Distribution of region scale s_r and region shape h_r for each detector, based on the entire sequence set. Region scale is defined as $s_r = \sqrt{ab}$, where a and b is the length of the major and minor axis respectively. Region shape is defined as $h_r = \frac{b}{a}$.

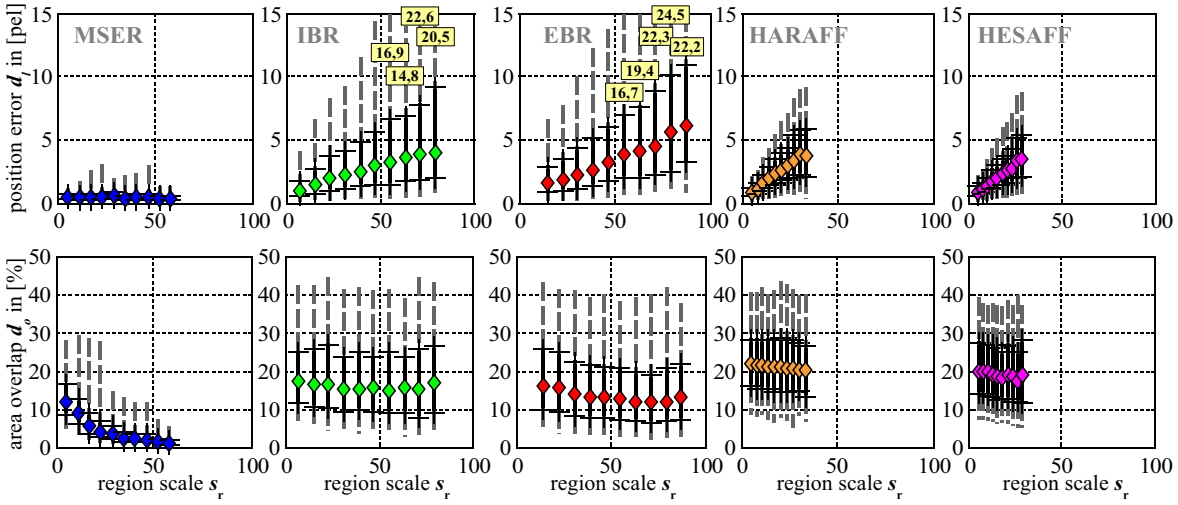


Figure 2.26: Dependency of region accuracy in terms of area overlap error d_o (bottom) and position error d_l (top) on region scale s_r. The diagrams show the median error (diamond markers), the 25- and 75-percentiles (solid lines) and the 5- and 95-percentiles (dashed lines), estimated from all sequences. Boxed numbers denote the 95-percentiles of d_l.

95 %-percentiles (dashed lines) have been computed for both d_l (upper diagram) and d_o (lower diagram). It can easily be seen, that for every detector except MSER, a linear dependecy of d_l on the region scale s_r exists. Especially for IBR and EBR, the spread of d_l increases strongly for $s_r \geq 50\ pel$. Both HARAFF and HESAFF detect mainly small regions below $s_r \leq 40\ pel$ while the 75 %-percentiles do not exceed $d_l \leq 6\ pel$.

As expected, d_o is largely independent of s_r. For MSER however, it decreases significantly with increasing scale: For $s_r \geq 25\ pel$, the median area overlap error is below $d_o \leq 0.5$. Using the distribution in figure 2.25 (left) in conjunction with the dependency plots in figure 2.26, potentially error-prone regions can be removed based on their scales, if high accuracy is sought. With EBR for example, removing all regions with $s_r \geq 50\ pel$ would reduce their numbers by approximately 25 % while at the same time, both median and spread were lowered significantly (to $d_{l,50} = 3.5\ pel$ and $d_{l,75} = 5.5\ pel / d_{l,95} = 13.5\ pel$ respectively).

Additionally, figure 2.25 (right) also shows the distribution of region shape, which is defined as the ratio between minor and major axis of the associated ellipse. For a circular region, shape would attain $h_r = 1$ while for an elongated ellipse, it would instead approach $h_r = 0$. As with region scale, dependencies between shape and localization accuracy have been statistically analyzed. Figure 2.27 shows the results for each detector. Obviously, dependencies are much less pronounced than with region scale. Only for MSER and IBR, a slight decrease in both median and spread with increasing shape can be found for d_l (upper diagram). In the case of area overlap d_o (lower diagram), a

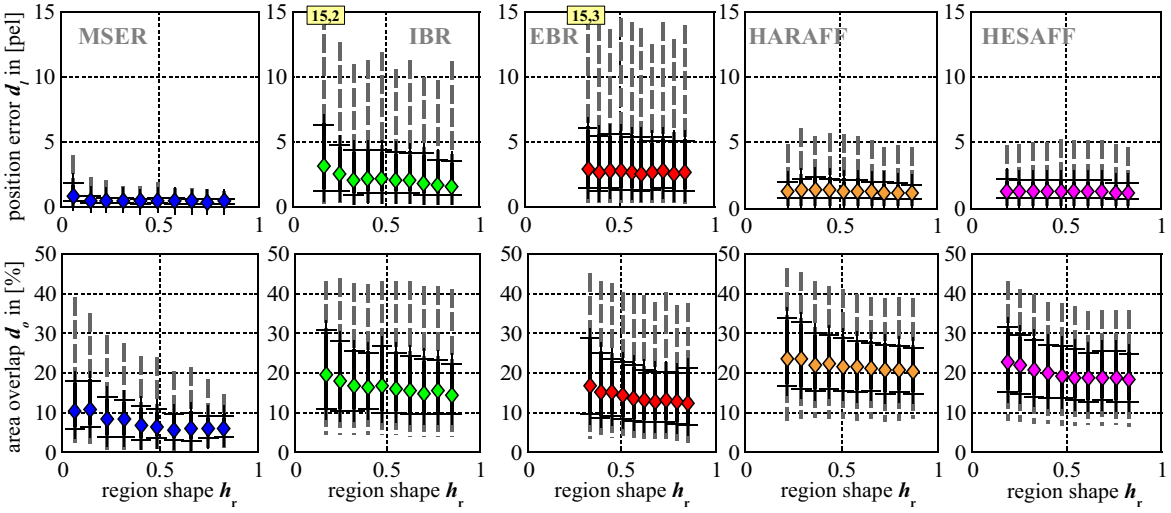


Figure 2.27: Dependency of region accuracy in terms of area overlap error d_o (bottom) and position error d_l (top) on region shape h_r. The diagrams show the median error (diamond markers), the 25- and 75-percentiles (solid lines) and the 5- and 95-percentiles (dashed lines), estimated from all sequences. Boxed numbers denote the 95-percentiles of d_l.

minor decrease in d_o can instead be found with every detector. Especially for MSER, the spread of d_o is significantly lower for circular regions. Concluding, the region shape h_r may as well be used as an adequate pre-selection property in addition to scale.

Thirdly, the density of regions in the image plane has been analyzed in figure 2.28. In the left diagram, the distribution of the number of neighbors for every region within the same image is given in accordance with equation 2.30 as

$$n_l = \left\{ r_i^m | d_o(r_i^l, r_i^m) \leq 0.5 \; ; \; m \in [1...N] \right\} \; , \tag{2.35}$$

where r_i^l is a specific region in frame I_i and r_i^m is taken from the set of all N regions in the same frame, less r_i^l. Intuitively, one would assume that the more regions existed within a local neighborhood around r_i^l, the greater were the probability of a mismatch during correspondence assignment. Judging from figure 2.28, EBR-regions exhibit by far the highest number of neighbors with almost $40\ \%$ above $n_l \geq 4$, followed with considerable distance by HESAFF and HARAFF. For MSER-regions, only approximately $20\ \%$ have at least a single neighbor.

However, if only the mere number of neighboring regions were considered, information on the respective degree of area overlap were lost. Therefore, in the right diagram, the *mean area overlap* between a region r_i^l and all its n_l neighbors is additionally shown,

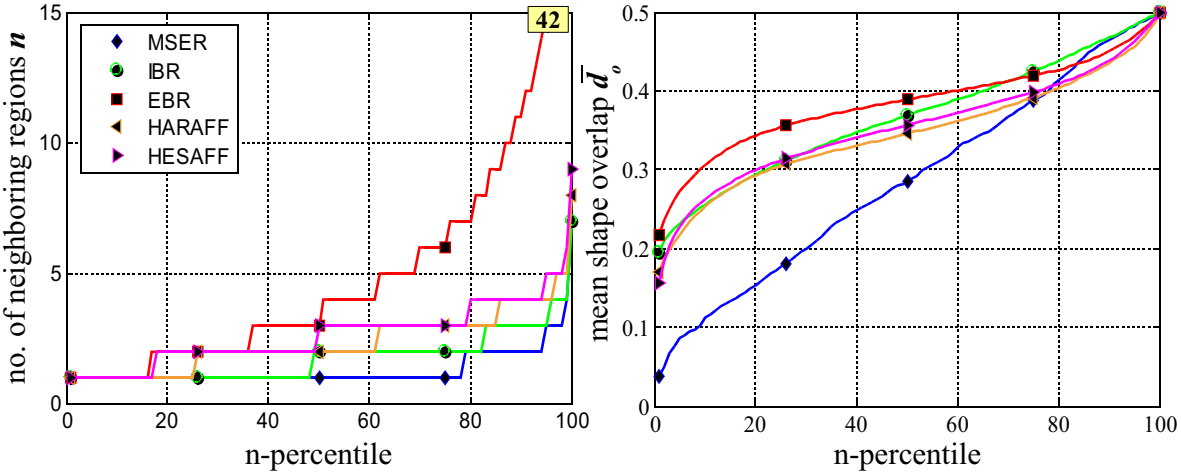


Figure 2.28: Distribution of the number of neighboring regions n (left) and the mean area overlap $\overline{d_o}$ for each detector, based on the entire sequence set. The boxed number denotes the maximum n at the 100-percentile.

given as

$$\overline{d_o}^{l} = \frac{1}{n_l} \sum_{k=1}^{n_l} d_o(r_i^l, r_i^k) \; . \tag{2.36}$$

Naturally, $\overline{d_o}$ can only be computed for regions with at least one neighbor. Thus, single regions with no neighbors have been omitted for the computation of figure 2.28 (right). It can be seen that for MSER-regions, the mean area overlap is smallest among all detectors: Although only 20 % of all regions have at least one neighbor, the similarity among neighboring regions is high with ≈ 50 % below $\overline{d_o} \leq 0.3$. For EBR, the mean area overlap is highest, closely followed by the remaining detectors: It can be presumed however, that the given results are at least partially influenced by region density: If the number of regions in a local neighborhood were high (as with EBR), the probability that several neighbors existed with considerable overlap would also be elevated, biasing $\overline{d_o}$ as a consequence.

As with both scale and shape, the dependency of region accuracy on the mean area overlap $\overline{d_o}$ has been evaluated statistically in figure 2.29. However, there is no clear dependency for any of the detectors to be observed (except with d_o for MSER and IBR). Considering the general data spread, this should not be deemed significant. Thus, although significant differences with regard to mean area overlap exist between detectors, this property clearly has no significant impact on region accuracy and is therefore not recommendable as an effective pre-selection property.

Lastly, in order to compensate for the said density-related bias, a new measure is proposed which uses the definition of the area overlap error from equation 2.30: Shape

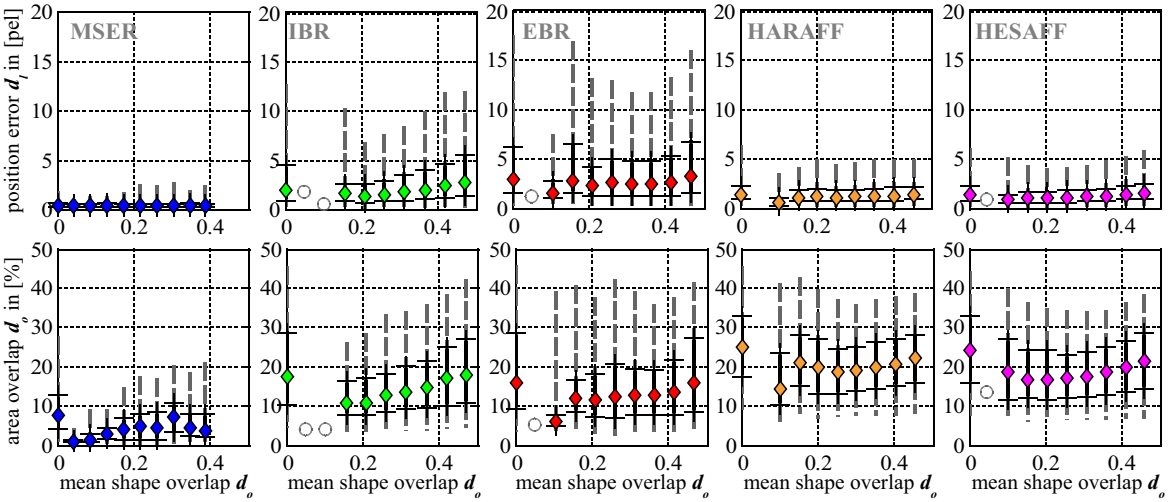


Figure 2.29: Dependency of region accuracy in terms of area overlap d_o (bottom) and position error d_l (top) on the mean area overlap $\overline{d_o}$. The diagrams show the median error (diamond markers), the 25- and 75-percentiles (solid lines) and the 5- and 95-percentiles (dashed lines), estimated from all sequences.

uniqueness s_u is defined as the minimum overlap between a region r_i^l and its second-closest neighbor within the same frame:

$$s_u = 1 - \min\{d_o(r_i^l, r_i^m) | d_o(r_i^l, r_i^m) \leq d_{o,max}, \quad l, m \in \{1...N\}\} , \qquad (2.37)$$

where N is the total number of regions in frame I_i.

In the case of congruent regions, this ratio will attain $s_u = 1$. Naturally, it is desirable for a detector to produce regions with low s_u as these are more likely to be successfully discriminated by an appropriate descriptor in the matching step. In the remainder of this work, the measure introduced in equation 2.37 is termed *shape uniqueness*, as it expresses in how far a specific region differs from its most similar neighbor.

In figure 2.30, the distribution of shape uniqueness ratios is shown for each of the five detectors. Among all methods, MSER shows the lowest s_u. Notably, the percentage of regions with more than one neighbor attains only $\approx 5\ \%$ for this detector. The remaining detectors can be partitioned into two groups: HARAFF/HESAFF exhibit the highest shape uniqueness, with significant distance to IBR/EBR.

As before, the dependency of region accuracy in terms of d_l and d_o on the newly introduced measure s_u has been evaluated statistically. The results are given in figure 2.31. Contrary to the mean area overlap, there exists a significant linear dependency: the higher s_u and thus similarity between neighboring regions, the larger are d_l and d_o. This relation holds for all five detectors, although most significant for IBR and EBR. For example: By discarding all IBR-regions with $s_u \geq 0.5$, $d_{l,75}$ can be lowered to $\approx 7\ pel$, while $d_{o,75}$ is at only 20 % (compared to 35 % with $s_u = 0.8$). Judging from these

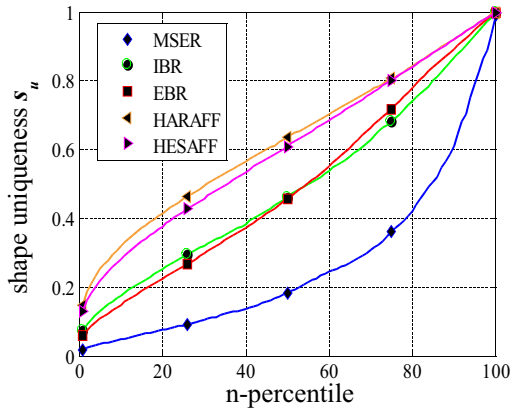


Figure 2.30: Distribution of shape uniqueness s_u for each detector, based on the entire sequence set.

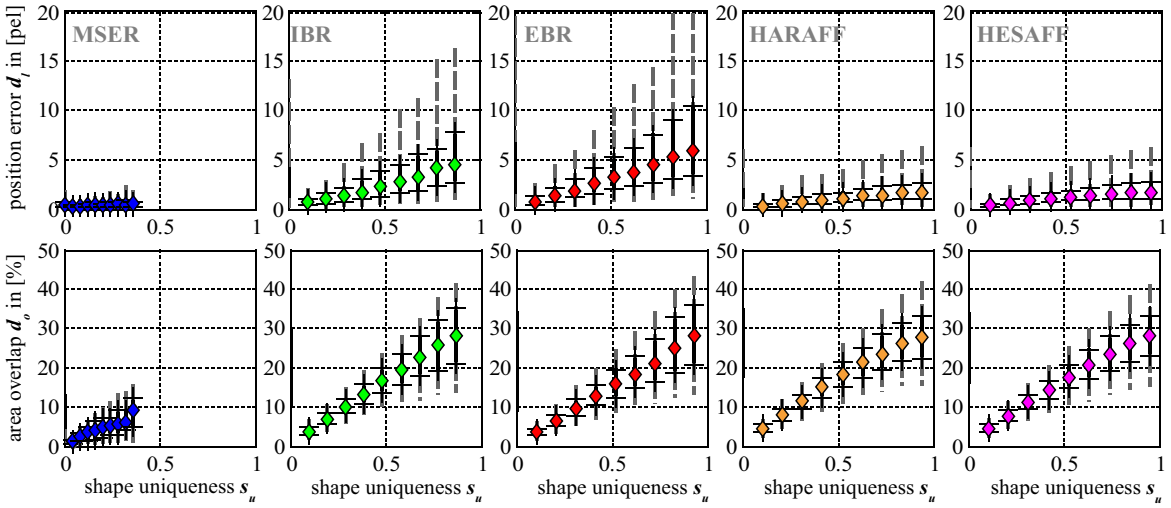


Figure 2.31: Dependency of region accuracy in terms of area overlap d_o (bottom) and position error d_l (top) on shape uniqueness s_u. The diagrams show the median error (diamond markers), the 25- and 75-percentiles (solid lines) and the 5- and 95-percentiles (dashed lines), estimated from all sequences.

results, shape uniqueness can be effectively used to reduce the number of error-prone regions in addition to region scale and shape.

2.5.4 Descriptor-based Region Correspondences

In the previous section, region accuracy has been assessed by assigning correspondences based on the area overlap d_o. The latter has been computed using the inter-image homographies from section 2.5.1. As these are usually not available in real applications, region descriptors such as SIFT or SPIN are used instead. Both methods compute a distinctive multi-dimensional vector from the image signal within the support area of a region. Given two frames, correspondences are then found by measuring the euclidean distance in descriptor space. The latter is defined as

$$d_d(r_i^{l'}, r_{i+1}^m) = \|d_l - d_m\| \,, \tag{2.38}$$

where d_l and d_m are multi-dimensional descriptors associated to the respective regions. Both are assigned to each other as a candidate correspondence, if the euclidean distance between them is below a pre-defined threshold, *i.e.* $d_d \leq d_{d,max}$. It will be shown in this section, that the resulting set of region correspondences using descriptor distance differs from the set using inter-image homographies. Within the latter, both the position error and the area overlap error are usually lower. It is investigated, to what extent the five affine-covariant detectors are affected, and which of the two region descriptors provides the better matching performance.

In the following, the set of descriptor-based region correspondences is referred to as c_d, whereas the set of overlap-based correspondences is termed c_o instead. Both may be partitioned into several subsets, as illustrated in figure 2.32. There, c_o (red) refers to the set of overlap-based correspondences with $d_{o,max} \leq 0.5$. The set $c_d = c_{d,o} \cup c_{d,i}$ denotes the set of descriptor-based correspondences instead. It may be further partitioned into *outliers* $c_{d,o}$ and *inliers* $c_{d,i}$, whereas only for the latter holds $d_o \leq 0.5$. If a specific region has only a single candidate during correspondence assignment, overlap-based and descriptor-based decisions will naturally be identical. If, however, multiple candidates exist, the two methods do not necessarily produce the same results, *i.e.* the correspondence with lowest overlap error does not also have to be the correspondence with lowest descriptor distance. These set differences are termed $c_{diff,d}$ and $c_{diff,o}$ respectively, according to figure 2.32. It can be expected, that the area overlap within the set $c_{diff,d}$ will be higher than with $c_{diff,o}$. Later in this section, supporting evidence for this assumption will be given.

Processing the set of all correspondence candidates according to the algorithm described in table 2.8 produces two disjoint sets, *positives* and *negatives*. While it is desirable to obtain many positives, a low number of negatives is sought at the same time. In the previous section, this issue has been already discussed in the context of finding an acceptable max. overlap threshold $d_{o,max}$ in figure 2.21. It could be seen, that the

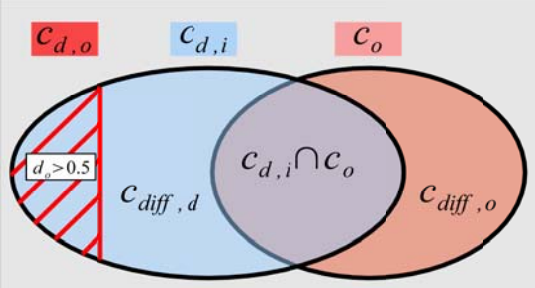


Figure 2.32: Set differences: overlap-based correspondences c_o (*right*) are based on $d_{o,max}$, while descriptor-based correspondences c_d (*left*) are based on $d_{d,max}$ instead. The latter can be subdivided into the subset of *inliers* $c_{d,i}$ for which additionally holds $d_o \leq 0.5$ and the subset of *outliers* $c_{d,o}$. For the latter, the area overlap error exceeds $d_o > 0.5$.

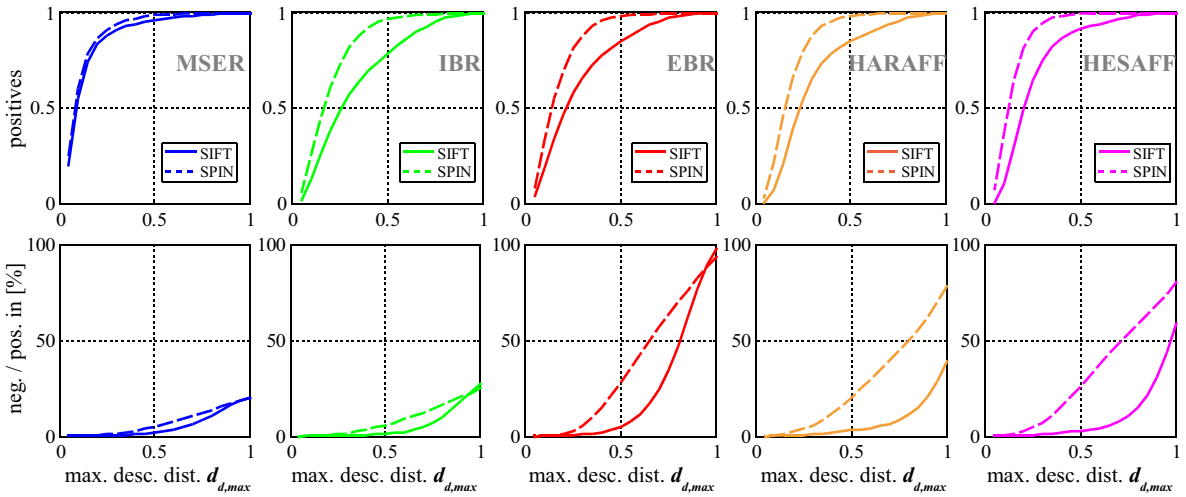


Figure 2.33: Dependency of the number of positives (*top*) and the ratio of negatives and positives (*bottom*) on the maximally permissible descriptor distance $d_{d,max}$.

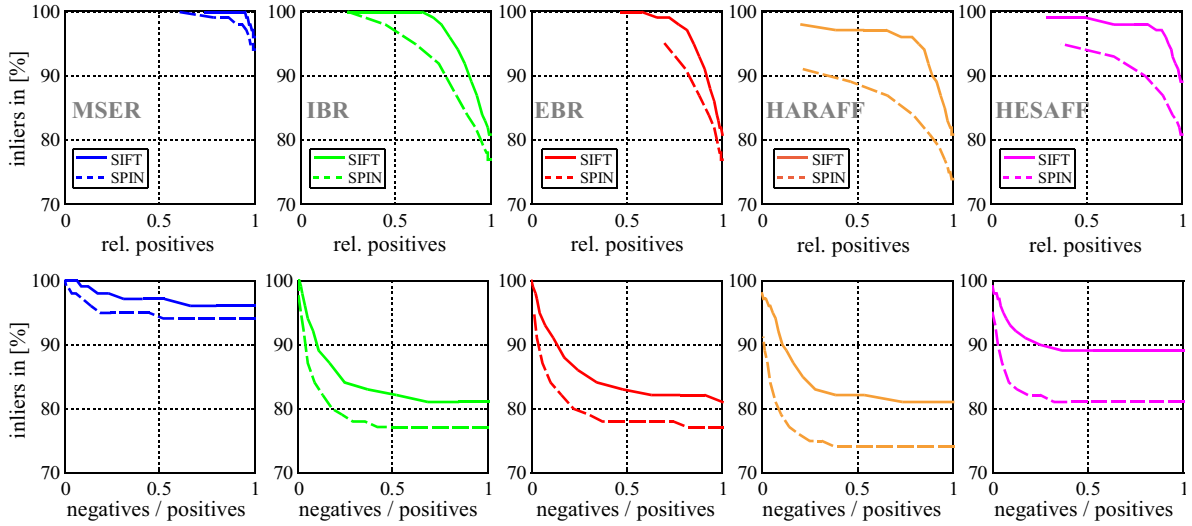


Figure 2.34: Dependency of the percentage of inlier correspondences (*i.e.* $c_{d,i}/c_d$) on the number of positives (*top*) and on the ratio of positives and negatives (*bottom*) for both SIFT (*solid line*) and SPIN (*dashed line*).

detectors behave differently with regard to both the absolute number of positives as well as to the ratio of negatives and positives. In this section, correspondence assignment has been performed using two different region descriptors - SIFT and SPIN. In figure 2.34, the behavior of both methods with each of the five detectors has been tested as a function of $d_{d,max}$. In the upper row, the relative number of positives is shown. It can be seen, that the curve progressions for SIFT and for SPIN differ from each other: With the former, the increase of positives is more shallow than with SPIN, where saturation is reached already for $d_{d,max} = 0.5$. However, a direct comparison of both descriptors is hardly possible from these characteristics alone. They are rather intended to provide information on the necessary descriptor threshold for a given number of positives. From the lower diagram, the ratio of negatives and positives for a specific choice of $d_{d,max}$ can be looked up additionally.

In figure 2.34, the percentage of inlier correspondences (*i.e.* $c_{d,i}/c_d$) is shown for each detector and both descriptors as a function of the relative number of positives (top) and the ratio of negatives and positives (bottom). It can be seen, that for a specific number of positives, the inlier percentage for the SIFT-descriptor is significantly higher than for SPIN. While for MSER the differences between both descriptors are minimal, HARAFF and HESAFF show a significant improvement for SIFT. In the lower diagram, SIFT-based correspondences also achieve a higher inlier percentage than SPIN at a much lower rate of negatives per positives. Again, HARAFF and HESAFF exhibit the greatest differences between both descriptors. Concluding, SIFT achieves a higher percentage of inlier correspondences than SPIN as well as a higher number of positives for the same

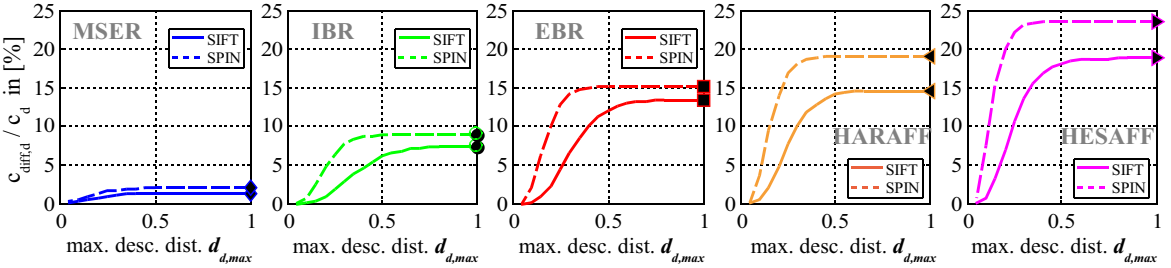


Figure 2.35: Dependency of the amount of set differences $c_{diff,d}/c_d$ on the maximally permissible descriptor threshold $d_{d,max}$. As the error within the set $c_{diff,d}$ is higher than within c_o, a low ratio is desirable. It can be seen, that SIFT-descriptors (*dashed line*) generally perform better than SPIN (dashed line), albeit at the expense of an increased computational complexity.

percentage of inliers. Also, the ratio of negatives and positives is much more favorable for SIFT.

Figure 2.35 shows the percentage of set differences $c_{diff,d}$ and $c_{diff,o}$ as a function of $d_{d,max}$, *i.e.* descriptor-based correspondences with $d_o \leq 0.5$ that can not be found in the set of overlap-based correspondences c_o and vice versa. As discussed previously, the area overlap in the set $c_{diff,o}$ will be lower than in $c_{diff,d}$ by principle. Thus, a low ratio of $c_{diff,d}$ and the size of the set c_d is desirable. For MSER-based correspondences, this ratio is smallest for all settings of $d_{d,max}$ for both SIFT and SPIN. With all detectors, $c_{diff,d}/c_d$ is higher for the latter, especially with HARAFF and HESAFF. For the latter, as much as 25 % of all SPIN-correspondences are different from the optimal set c_o at $d_{d,max} = 0.5$, while for MSER differences amount to only 2.5 %.

Finally, figure 2.36 shows the set differences in terms of area overlap d_o (bottom) and position error d_l (top). The solid line always denotes the respective error in the set $c_{diff,d}$, while the dashed line shows the error in the overlap-based set $c_{diff,o}$. In every case, the latter is smaller for both d_o and d_l. Notably, differences in d_l between both sets are very small for HARAFF and HESAFF. This alleviates the significant set differences seen in figure 2.35. With regard to d_o however, differences are more pronounced. Between SIFT and SPIN, no differences in the distributions could be found. For this reason, figure 2.36 contains only the results for the SIFT-descriptor. Concluding, differences between both sets are most significant for EBR and generally manifest more in d_o than in d_l.

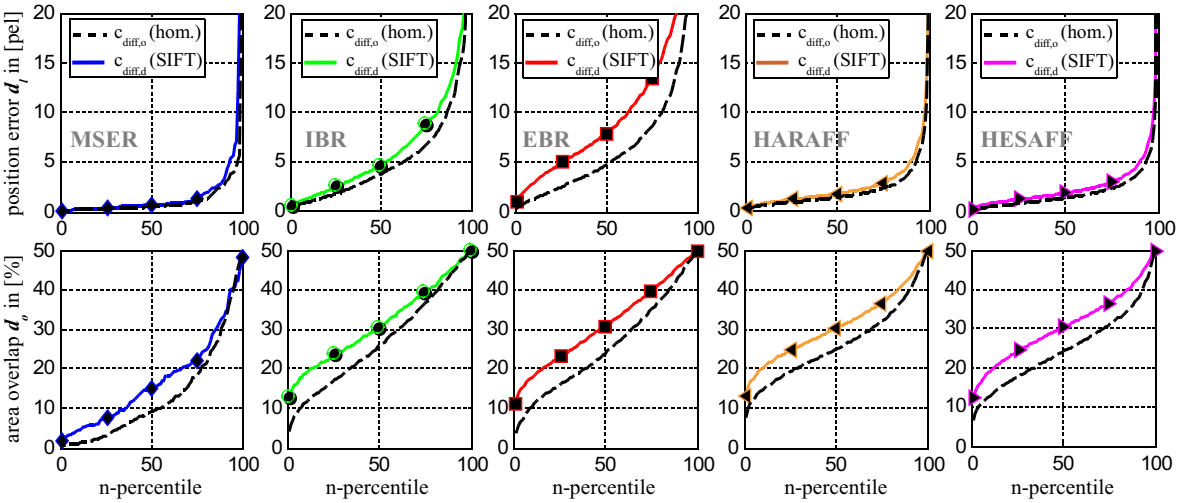


Figure 2.36: Distribution of region accuracy in terms of area overlap d_o (bottom) and position error d_l (top) over the n-percentile. The solid lines denote the error within the set of descriptor-based correspondences $c_{diff,d}$, while the dashed lines show the error in the complementary set of homography-based correspondences $c_{diff,o}$. Set terminology is illustrated in figure 2.32.

2.6 Chapter Conclusion

In this chapter, five state-of-the-art affine-covariant region detectors have been compared with regard to the number of region correspondences, the percentage of successful matches and the accuracy of region localization. The evaluated image sequences have been taken from a publicly available database, so that a validation of the given results and a comparison to other methods is possible. The used stereo camera setup has been carefully calibrated, so that lens distortion effects are minimal. In order to enable a sound assessment of detector accuracy, only piecewise planar objects have been selected, such that a set of homographies can be estimated which relates the object position between adjacent frames and which is thus able to predict for each region the presumed shape and position of its most likely correspondence. The estimation of these homographies has been performed based on MSER-region correspondences: It has been shown, that these exhibit the smallest residuals among all five detectors.

With regard to the number of correspondences, the HESAFF-detector is clearly superior, followed with distance by both HARAFF and EBR. The smallest number of region correspondences was provided by the MSER-detector. However, the latter exhibited the best performance with respect to the percentage of successfully matched features (the *repeatability score*): Given the number of region-to-region matches for a pair of images, repeatability was computed by normalization on the smaller of the number of regions in both images. For a maximally permissible overlap of 50 % during correspondence as-

signment, the MSER detector achieved a repeatability score of $\approx 85\ \%$, while all other detectors were well below 70%.

Region localization accuracy was evaluated using the euclidean error of the region centers d_l and the area overlap error of the associated ellipses d_o, given the ground truth homographies. Concluding, the best detector with regard to d_l on all 6 sequences is MSER, followed with distance by HESAFF. Both IBR and EBR clearly performed worst, with $>\ 20\ \%$ of all correspondences above $d_l\ \geq\ 5\ pel$. With regard to d_o, MSER-regions repeatedly showed the smallest errors. Notably, the area overlap error of both IBR and EBR is considerably smaller than with HARAFF and HESAFF, although the latter two exhibited a smaller d_l instead.

In order to identify and remove error-prone regions before correspondence assignment, the dependency of several region properties such as scale, shape and the density in a local neighborhood on both d_l and d_o have been evaluated statistically. *Firstly*, with regard to d_l, each detector except MSER shows a clear dependency on region scale. The area overlap error d_o on the other side has proved largely independent of scale, except for MSER-regions which show a decrease in d_o with increasing scale. *Secondly*, for region shape (which is defined as the ratio between minor and major axis length of the associated ellipse), dependencies are less pronounced. With regard to d_l, a slight decrease with increasing shape could be found for MSER and IBR. For all other detectors, d_l showed constant over shape. With regard to area overlap however, a decrease in d_o could be shown for each detector. Obviously, circular regions are more accurately localized than elongated regions. *Thirdly*, the dependency of localization accuracy on the density of regions has been analyzed. The latter has been defined as the number of neighboring regions in a local neighborhood with an area overlap of $\leq 50\ \%$. Intuitively, one would assume that the more regions existed within a local neighborhood, the greater were the probability of a mismatch during correspondence assignment. However, there is no clear dependency for any of the detectors to be observed (except with d_o for MSER and IBR). Thus, region density can not be used effectively for the pre-selection of regions in order to improve localization accuracy. *Lastly*, a new measure closely related to region density has been evaluated, which considers the overlap between a region and its closest neighbor within the same image. This new measure - termed *shape uniqueness* s_u- expresses in how far a specific region differs from its neighbors with regard to the image signal enclosed within the area of support. Thus, it also indicates how well a region can be (presumably) discriminated by a region descriptor during matching. It has been shown, that a significant and approximately linear dependency between shape uniqueness and both d_l and d_o exists. The higher s_u and thus the similarity between neighboring regions, the larger are the localization errors. Based on s_u, error-prone regions can be removed before the actual matching step. Thus, the computational load of descriptor-computation and matching can be significantly reduced. This is especially beneficial in applications, where the available hardware is limited. For all region properties, the distributions have been additionally given so that the number of affected regions for a

specific pre-selection setting can be easily found. With EBR for example, removing all regions with scale $s_r \geq 50\ pel$ would reduce their numbers by approximately $25\ \%$ while at the same time, the median of d_l were lowered significantly to $d_{l,50} = 3.5\ pel$.

Usually, homographies between frames are unavailable in typical applications. In this case, region descriptors such as SIFT or SPIN are used for correspondence assignment between frames. Compared to the correspondences based on area overlap (which has been used solely for evaluation purposes), a different set with lower accuracy results. Based on the type of descriptor used for matching, the set differences are of different magnitudes. For the SIFT descriptor, compliances (and thus accuracy) are generally higher. Further, it has been shown that SIFT- and SPIN-descriptors respond differently with regard to variations of the maximally permissible distance threshold $d_{d,max}$: Usually, the increase of both the number of correspondences and candidates is faster for SPIN. Also, the percentage of inlier correspondences (with an area overlap of at least $50\ \%$ given the groundtruth homographies) is lower and thus worse for the latter. For a specific inlier percentage, the ratio of correspondences and actual candidates is much lower (and thus more favourable) for the SIFT-descriptor. Concluding, the performance of the SIFT-descriptor with regard to uniqueness, the number of successful matches and localization accuracy is superior to SPIN, albeit at the expense of a higher computational effort.

The evaluation results within this chapter serve as a supplement to existing comparative studies and facilitate the selection of appropriate detectors for a given application (such as monocular tracking). The major contributions have also been published in [HJA08b]. In the next chapter, both descriptors will be further compared to each other in the context of tracking.

3 Monocular Region Tracking

3.1 Chapter Introduction

There are two major contributions contained within this chapter: Firstly, the assessment of region detectors is extended from mere localization in single images to the task of tracking through monocular image sequences. Secondly, a novel graph-based tracking method is proposed, which is in many ways superior to existing standard techniques from the literature.

In the previous chapter, a selection of affine-covariant detectors has been evaluated and compared, especially with regard to errors in region position and shape of the surrounding support area. Within this chapter, the assessment of the very same detectors will be continued in terms of their suitability for descriptor-based tracking in monocular image sequences. While in the previous chapter, region descriptors were used for the purpose of finding correspondences for accuracy assessment, they will now be used for the task of tracking the same region through an entire sequence of images.

Ideally, a region detector should provide salient locations within an image, whose surrounding neighborhoods showed strong variations of the intensity signal in both coordinate directions to allow for a unique and accurate localization. At the same time, it should also contain sufficiently discriminatory information for a subsequent region descriptor. The latter should ideally derive a unique and highly distinctive measure, which captured even small region differences but was simultaneously robust against camera noise and other visual disturbances. At the same time, the computational complexity of both region detectors and descriptors should be small. In practice however, a trade-off between distinctiveness and complexity usually has to be made. To this end, two histogram-based methods have been selected from the variety of available region descriptors - SIFT and SPIN. Both have been introduced already in section 2.2.3 of the previous chapter and it has been shown in section 2.5.4, that there exist significant differences in terms of the percentage of successfully matched regions and in the number of ambiguous correspondence candidates. With regard to distinctiveness, SIFT is clearly superior to SPIN, albeit at the cost of increased computational complexity. The decisive advantage of both methods however is the possibility to directly determine a euclidean distance measure in multi-dimensional descriptor space. Other methods such as the *local jet* [SM97] additionally require the computation of a covariance matrix from a representative set of sample images for this purpose. In this case, descriptor similarity is expressed using the *Mahalanobis* distance instead [Cox93]. For reasons of transparency,

such methods have not been used within this thesis, as the choice of sample images directly influences the evaluation results. For histogram-based descriptors, this influence can be avoided entirely. Also, the latter have shown superior performance in comparative evaluations, such as [MS05].

In addition to the performance measures from the previous chapter, three additional criteria are introduced here: trajectory length, the number of outliers and robustness against increasing perspective distortion of the observed objects. An *outlier* in the context of tracking is defined as a region correspondence within a trajectory, for which the area overlap error exceeds a predefined threshold. In order to compute this measure, the groundtruth homographies introduced in section 2.5.1 are used. In the case of high region density and low descriptor distinctiveness, the number of outliers is usually very high. One of the major contributions of this chapter is a novel tracking technique, which greatly reduces the number of outliers, especially with less distinctive region descriptors such as SPIN.

Basically, the tracking task may be subdivided into two major stages: *region detection* and *correspondence assignment*. While the first stage has been extensively discussed in the previous chapter already, the current chapter is concerned with the second stage instead. Correspondence assignment refers to the process of associating regions in neighboring frames to each other on the basis of their descriptor similarity. In this stage, a region from the first frame may be assigned to multiple candidate regions in the second frame and vice versa. Hence, the resulting set of correspondences contains potentially ambiguous elements. In order to resolve assignment conflicts, uniqueness is enforced such that each region belongs to exactly one trajectory in the end. Two methods from the literature have been implemented for this purpose and are compared against each other: While the first (and most simple) method searches for region-to-region correspondences within a circular gating region of constant size and decides on the pair with highest descriptor similarity (nearest-neighbor approach), the second method performs a model-based state prediction of the presumed feature location in the next frame (Bayesian-filtering approach) to narrow the search area and thus reduce the number of correspondence candidates. Both methods employ a combinatorial optimization scheme for the resolution of ambiguities among the candidates of the two most recent frames. Although the results of the Bayesian filtering approach are superior to the nearest-neighbor method, both still exhibit a comparatively large number of outliers.

In order to compensate for this drawback, a new method for an improved treatment of correspondence ambiguities is introduced in this chapter. Instead of deciding on unique correspondences for every frame pair (as with both the nearest-neighbor approach and Bayesian filtering), the proposed method keeps all candidate correspondences until the end of the sequence (or for a predefined number of frames) in so-called *track graphs*, which model the relations between regions based on descriptor distance and an additional motion smoothness (or *path coherence*) model from the literature. The latter is further extended such that the additional shape information provided by the region detectors

is appropriately exploited. Instead of prematurely selecting the locally most probable candidate, decisions on specific correspondences are postponed until sufficient evidence has been gathered that allows for an improved extraction of trajectories by means of weighted graph-traversal.

This chapter is organized as follows: Firstly, a brief taxonomy of methods for tracking and motion analysis in computer vision is given in section 3.2, as well as a short introduction into the concepts of Bayesian filtering (especially Kalman filters) and into the well-known KLT-tracker. Secondly, three concepts for descriptor-based tracking are presented in section 3.3. Thirdly, in section 3.4, all three methods are compared to each other with regard to the above-mentioned performance measures. Finally, in the same section, the results of descriptor-based region tracking are compared against the KLT-tracker [TK91], which is a widely-used and well-known method for feature tracking in computer vision and thus serves as an adequate standard reference.

The presented tracking concepts and evaluation results serve as a prerequisite for chapter 4 but may also be used in a self-contained way. The major aspects have been additionally published in [HJA08a].

3.2 Background

3.2.1 An Introduction to Tracking

In computer vision, tracking means to maintain correspondence of an image structure over multiple frames. Generally, such a structure belongs to a three-dimensional object, that moves within the field of view of the observing camera. Both position and appearance of the object in the images change, as either the object itself or the camera move in world space, depending on the camera and lens type, on the scene illumination parameters (ambient or directed light source, constant or changing intensity), on potential occlusions and on measurement noise. If no prior assumptions on the nature of both the object and of the world in which it exists are made, the success of a tracking algorithm will be limited. Generally, the suitability of such assumptions to the existing conditions define the *robustness* of a tracking method. If *e.g.* a constant-velocity motion model is assumed for an accelerating object, correspondence maintenance is likely to fail. In [TH96], robustness is further defined as the ability of a tracking system to track accurately and precisely during or after visual circumstances that are less than ideal. Also, the authors broadly divide the existing literature on (robust) tracking into two major categories:

Pre-failure methods aim at avoiding tracking failures by anticipating visual disturbances and attempt to track despite them. They usually employ robust statistics, temporal filtering or *ad-hoc* methods for handling specific types of visual perturbations (*e.g.* [Vin96]). Other methods take advantage of known (or assumed) dynamics and noise models, such as [BSFC90] or [Rei79]. Generally, different disturbances have different

solutions, such as modeling changes in ambient lighting conditions by concentrating on color cues [RTH96], edge-based tracking [Low92], or by explicitly modeling illumination parameters [HB98]. In the case of fast and/or unpredictable motion, probabilistic models have also been used, *e.g*. in [IB96].

Post-failure methods however are designed to recover from failure when it has already happened. Robustness is generally achieved by some form of high-level processing, which incorporates explicit knowledge about the tracked object. As noted in [TH96], much of the existing work on this topic has been inspired by the concept of *focus of attention* in biological systems. As suggested by cognitive science research, biological vision systems are broadly organized in pre-attentive and post-attentive stages. While the task of the former stage is to locate regions of interest at high speed but with limited accuracy, the post-attentive stages' purpose is to examine these regions more closely, with a focus on robustness and accuracy [Nei67] [Wol94]. This notion has been integrated into tracking systems, where one algorithm is concerned with rapidly identifying relevant candidate regions in a pre-attentive (low-level) stage while another one post-attentively tries to perform the actual (high-level) tracking task [IB98].

In this work, an alternative classification of tracking methods is presented. On the top level, a division into dense and sparse trackers is made. In the spirit of Toyama and Hager, all of the subsequently presented approaches can be attributed to the group of pre-failure methods as introduced above.

Dense trackers try to provide information on the motion of every pixel within an image. Without claiming completeness, two major techniques have been attributed to this group, namely optical flow methods and correlation methods, which rank among the earliest tracking approaches. Assuming that appearance changes between adjacent frames are small, a window of pre-defined size around an image location in the first frame is used to determine the corresponding location in the second frame by shifting an equally-sized window in the proximity of the original location and by maximizing the signal-correlation between both. In [SP72], such a method has been applied in the context of cloud-tracking from ATS-images. Usually, if the appearance change between frames is too large, tracking failure occurs. Therefore, and because of their comparatively high computational load, correlation-based methods are seldom used nowadays in the context of tracking. A concept that is closely related to signal correlation is the estimation of the *optical flow*. In computer vision, the latter denotes a vector field, that provides information on the 2D-direction and velocity (*i.e*. the motion) for every image point within an image sequence. In the context of tracking, the optical flow can be used for the prediction of the presumed location of an image point in subsequent images. Contrary to correlation-based methods, flow-based techniques are able to incorporate a more sophisticated model of structure appearance into the estimation process, such as affine shape changes. However, they also require inter-frame motion to be sufficiently small. In order to successfully compute the optical flow on sequences with larger disparities, pyramidal approaches are often used that start on a coarse (down-sampled) representa-

tion of the original image and successively increase the resolution in order to improve estimation accuracy. A significant drawback of flow-based methods is their inherent sensitivity to illumination changes. To this concern, a method that addresses the issue of computing the optical flow under illumination changes has been proposed in [BFY98] and in [Neg98]. Also, robustness and accuracy are highly dependent on the method of flow computation. An extensive overview of currently available estimation techniques is given in [Jäh05]. It should be noted that, although both correlation- and flow-based tracking techniques originally rank among the dense trackers, they can also be used in a sparse environment, where only specific salient image points are selected for tracking. In conjunction with Bayesian state estimators (*e.g.* a Kalman-filter), they may well serve as an auxiliary guiding function for state estimation in a probabilistic tracking framework, such as in [Wue04].

Sparse trackers on the other hand select only subsets of all pixels in an image. In the following, such a subset is referred to as a *feature*. The latter may consist of a single pixel or of a group of pixels (which is termed a *region*). As not every pixel in an image provides sufficiently discriminatory information for a unique and accurate relocalization in neighboring frames, a pre-selection of suitable candidates by an appropriate detection method may be advantageous over dense trackers, as unstable candidates can be sorted out prior to the actual tracking task. Depending on the selection method, single features are generally more robust to appearance variations, caused by view or illumination changes. A well-known example for a robust feature detector is the Harris-method introduced in section 2.2.1: From the image signal in a support area around a specific position $\mathbf{x}$, the second-order moment matrix $\mu(\mathbf{x})$ is computed by Gaussian convolution and successive differentiation of the smoothed image signal with respect to the main image axes. A salient feature is defined as one, which yields a strong signal variation in both coordinate directions, indicated by two sufficiently large eigenvalues of μ. Edges and line crossings are examples of feature types, to which the Harris-detector responds very well. Typically, they are well-localized and robust against small perspective and illumination changes [SMB00]. However, in the case of major perspective changes, methods with an adaptive (*i.e.* covariant) support area are preferable instead. A selection of popular representatives of this type of detectors has already been introduced and evaluated in the previous chapter.

Sparse trackers typically consist of two major stages, *feature detection* and *data association*. While the former is responsible for localizing stable feature locations, the task of data association is to identify and assign these to each other between frames, depending on some kind of similarity measure (*e.g.* region descriptor distance or the degree of deviation from a predicted position). According to [BSFC90], there exist two fundamentally different models for data association. The first one is a *deterministic model* where the most likely of several candidate correspondences is selected for each feature while the rest is discarded, based on local information and (in the case of a Bayesian approach) the track-history of a feature. In a global sense however (*i.e.* considering several frames

or even the entire sequence), the winning candidate does not necessarily have to be the correct one and might eventually entail an entire series of erroneous assignments in the remaining frames of the sequence. An advantage of deterministic methods however is the ability to provide a set of (incrementally built) trajectories with each new frame. In a time-critical online tracking system this is most-often an indispensable property.

Without claiming completeness, the abundance of available sparse tracking methods may be further divided into Bayesian and non-Bayesian algorithms. Among the Bayesian approaches, the Kalman filter (KF) is probably the most well-known. It provides a means of statistically estimating the state of a dynamic system (*i.e.* for example the presumed location of a feature in the next frame) from a sequence of noisy measurements. Preconditions for the use of a KF are a linear system model (*e.g.* a constant-velocity motion model) and a Gaussian-distributed measurement error. Under these circumstances, the KF provides an optimal solution to the Bayesian filtering problem [WB01]. In the case of a non-linear model, the extended Kalman filter (EKF) employs further linearization by means of a Taylor-series expansion as an approximation. However, this can introduce large errors into the state estimates, which may lead to sub-optimal performance and sometimes even divergence of the filter. A better solution is provided by the unscented Kalman filter (UKF). Here, the state distribution is represented using a minimal set of carefully chosen sample points. These sample points completely capture the true mean and covariance of the system state and avoid the linearization of the system model. Notably, the computational complexity of the UKF is of the same order as that of the EKF. In situations where the measurement errors are not Gaussian-distributed, a particle filter provides a more adequate solution. Within this work however, the necessary preconditions for the use of an ordinary KF are largely fulfilled and neither EKF/UKF nor particle filters are needed. For further information on the latter, a good overview of available techniques can be found in [AMG$^+$02]. In sections 3.2.2 and 3.2.3 of this chapter, the general Bayesian filtering framework and the KF-equations are introduced and discussed.

In the case of non-Bayesian tracking, data association is performed on the basis of an appropriate similarity measure between features (*e.g.* descriptor distance) of the current and previous frame alone. Therein lies the decisive difference to Bayesian filtering, where the history of a feature is captured in form of a state covariance matrix which evolves with each new assignment and contributes to the decision process for the current frame. Most often, a nearest-neighbor approach is used which assigns to each feature the most likely candidate. In the case of ambiguities (when several features claim the same candidate or vice-versa), an optimization procedure is performed to globally find the optimal set of correspondences which maximizes *e.g.* the overall descriptor similarity. Instead of using region descriptors for feature assignment, the well-known *Kanade-Lucas-Tomasi-* or KLT-tracker [ST94] aims at minimizing the dissimilarity between image regions under an affine shape model instead. Although the assignment step of this method does not employ combinatorial optimization, it is attributed to the general group

of deterministic non-Bayesian tracking methods. The second category of approaches to sparse tracking employs a *probabilistic model*. Contrary to deterministic approaches, such methods aim at incorporating multiple measurements (*i.e*. features) into a joint estimation process in order to find the best correspondences. In the case of multiple competing tracks, a solution is desirable that keeps track of all possible assignments instead of concentrating on the locally most feasible correspondence. One of the most widely applied methods for handling such 'probabilistic assignment ambiguities' is the *joint probabilistic data association filter* (JPDAF), which is a strategy for multi-feature tracking given uncertainties in data association [VGP05][BSFC90]. Usually, the JPDAF is used in conjunction with a Kalman filter (KF/EKF/UKF) or particle filter. Its idea is to weigh all measurements with all existing tracks, where a weight represents the probability that a certain measurement originated from a specific feature, hence the term *probabilistic data association*.

A second method for multi-target tracking is the *Multiple Hypothesis Filter* (MHT) for multiple targets introduced in [BSFC90]. While the Bayesian filtering framework is target-oriented (*i.e*. the probability that each measurement belongs to an established feature is evaluated), the MHT is measurement-oriented instead. This means, that the probability that each existing feature or a new feature gave rise to a certain measurement sequence is obtained. However, the high computational complexity of both JPDAF and MHT [STVL04] prohibits their use in the context of this work. For further information, a well-structured taxonomy of currently available multiple target tracking methods is provided in [Pul05], which also contains a short overview of advantages, specialties and limitations for each method.

One of the major contributions of this chapter is the introduction of a method for non-Bayesian tracking. Instead of modeling the relations between features based on the covariance estimates of a Bayesian filter, the proposed *graph-based method* integrates all correspondence candidates for each feature into a directed multi-edge track graph by gradually traversing through an image sequence. Relations (edges) between corresponding regions (nodes) are weighted using a combination of descriptor distance and a path-coherence model, which also evaluates and penalizes the relative changes of region scale. Single trajectories (paths) are iteratively extracted from each graph by using Dijkstra's method for graph traversal [Dij59]. The major advantage of this method is the preservation of all possible assignment combinations for a pre-defined number of frames so that decisions can be reached in a (temporally) global sense. Instead of prematurely selecting the locally most probable candidate, decisions on specific correspondences are postponed until sufficient evidence has been gathered that allows for an improved extraction of trajectories by means of weighted graph-traversal. If all ambiguities have been finally resolved, trajectories may easily be assembled by following each region from its first frame of appearance to the last.

3.2.2 Generic Bayesian Filtering Framework

Bayesian filtering is a general probabilistic approach to the sequential estimation of an unknown probability density in a dynamic system, using a series of sensor measurements and a mathematical process model. In the following, the framework for a generic model is described, which is parameterized by a state $\mathbf{s}_i$, where i denotes a discrete time or frame index. The process of tracking in such a framework is defined as the recursive estimation of a sequence of states $\mathbf{s}_{1:i} = \{\mathbf{s}_1, ..., \mathbf{s}_i\}$ based on a set of measurements $\mathbf{z}_{1:i} = \{z_1 ... z_i\}$ up to the current time step i. In order to obtain a degree of belief in the current state estimate $\mathbf{s}_i$, the *a-posteriori* probability distribution $p(\mathbf{s}_i|\mathbf{z}_{1:i})$ is sought, since it embodies all available statistical information and is thus the complete solution to the estimation problem, including a measure of accuracy for the estimated state.

A preliminary condition for the application of a Bayesian approach is the validity of the *Markov*-condition

$$p(\mathbf{s}_i|\mathbf{s}_{1:i-1}, \mathbf{z}_{1:i-1}) = p(\mathbf{s}_i|\mathbf{s}_{i-1}) \ , \tag{3.1}$$

which is fulfilled if the current state s_i solely depends on the previous state s_{i-1} and is at the same time independent of all previous measurements $\mathbf{z}_{1:i-1}$ [VGP05]. State estimation in the Bayesian sequential framework is performed in a two-step recursion, which consists of a prediction step and a filtering step.

In the *prediction step*, the state density is estimated using the Chapman-Kolmogorov equation and the Markov condition in equation 3.1, which provide the *a-priori probability density*

$$p(\mathbf{s}_i|\mathbf{z}_{1:i-1}) = \int p(\mathbf{s}_i|\mathbf{s}_{i-1}) p(\mathbf{s}_{i-1}|\mathbf{z}_{1:i-1}) d\mathbf{s}_{i-1} \ . \tag{3.2}$$

It is assumed here, that $p(\mathbf{s}_{i-1}|\mathbf{z}_{1:i-1})$ from the prior time step is available.

In the *filtering step*, the *a-posteriori* density is computed from the *a-priori* density as

$$p(\mathbf{s}_i|\mathbf{z}_{1:i}) = \frac{p(\mathbf{z}_i|\mathbf{s}_i) p(\mathbf{s}_i|\mathbf{z}_{1:i-1})}{p(\mathbf{z}_i|\mathbf{z}_{1:i-1})} \ , \tag{3.3}$$

using Bayes' rule as shown in [Wit08].

A recursive filter provides an estimate of the current state each time a new measurement is received. This approach opposes to batch-methods, which require the existence of all measurements $\mathbf{z}_{1:i}$ in order to provide a solution. Thus, a Bayesian approach is ideally suited for online-tracking, where a current flow of state estimates at each time step is needed. Also, the need of storing the complete data set is rendered unnecessary as well as a repeated processing of the existing data.

Recursion requires the definition of a dynamic model which describes the evolution of system states $p(\mathbf{s}_i|\mathbf{s}_{i-1})$ - also defined as the *state transition function* - and the *a-posteriori* state density $p(\mathbf{s}_{i-1}|\mathbf{z}_{1:i-1})$ from the previous time step $i-1$ in order to

predict the *a-priori* density $p(\mathbf{s}_i|\mathbf{z}_{1:i-1})$ for the current time step i. Given a model for the state likelihood in the light of the current measurement $p(\mathbf{s}_i|\mathbf{z}_i)$, the *a-priori* density can be updated by multiplication, which yields the *a-posteriori* density for the current time step as defined in equation 3.3. The denominator $p(\mathbf{z}_i|\mathbf{z}_{1:i-1})$ is further computed as

$$p(\mathbf{z}_i|\mathbf{z}_{1:i-1}) = \int p(\mathbf{z}_i|\mathbf{s}_{1:i-1})p(\mathbf{s}_i|\mathbf{z}_{1:i-1})d\mathbf{s}_i \ , \tag{3.4}$$

which serves as a normalization term that incorporates the likelihood function $p(\mathbf{z}_i|\mathbf{s}_i)$ defined by the measurement model and the known (or assumed) statistics of the measurement noise.

The recursion only yields a closed-form solution in a small number of cases. For linear and Gaussian dynamic and likelihood models, the well-known Kalman filter (KF) provides such a solution. For non-linear and/or non-Gaussian models, approximation techniques are required such as the extended Kalman filter (EKF), the unscented Kalman filter (UKF) or particle filters.

3.2.3 The Kalman Filter

The original Kalman filter [Kal60] is basically a set of mathematical equations, that provides a recursive means of estimating the state of a process, in a way that minimizes the mean of the squared error. The filter supports the estimation of past, present and future states, even if the precise nature of the modeled system is unknown [WB01]. In the following, an introduction into the nature of the filter is given.

The Kalman filter addresses the general problem of estimating a state vector $\mathbf{s} \in \mathbb{R}$ of a system or an object. In a tracking application, $\mathbf{s}$ could contain both position and velocity of a feature point. The corresponding discrete-time controlled process is governed by the linear stochastic difference equation

$$\mathbf{s}_i = E\mathbf{s}_{i-1} + F\mathbf{u}_{i-1} + \mathbf{w}_{i-1} \ , \tag{3.5}$$

where the $n \times n$-matrix E relates the process state at time $i-1$ to the new state at time i. The random variable $\mathbf{w}_{i-1}$ represents the process noise, while the (optional) $n \times l$-matrix F further relates the control input vector $\mathbf{u} \in \mathbb{R}$ to the state $\mathbf{s}$. In the following, F and $\mathbf{u}$ are discarded, as there exists no external control which governs the motion of feature points in the observed scenes.

Further, the measurement $\mathbf{z} \in \mathbb{R}$ of an object is defined as

$$\mathbf{z}_i = G\mathbf{s}_i + \mathbf{v}_i \ , \tag{3.6}$$

where the $m \times n$-matrix G relates the current state $\mathbf{s}$ to the current measurement $\mathbf{z}$ at the same time step i. The dimension m of the measurement does not necessarily have to be equal to the dimension n of the state vector: While the latter might consist

of both position and velocity ($n = 2$), the former could contain position information only ($m = 1$). The vector $\mathbf{v}$ represents the measurement noise, which is assumed to be Gaussian-white with zero-mean. Both process noise and measurement noise may be expressed as

$$p(\mathbf{w}) \sim N(0, Q) \tag{3.7}$$
$$p(\mathbf{v}) \sim N(0, S)\ , \tag{3.8}$$

where Q and S are the respective covariance matrices. In order to initialize the Kalman filter, both Q and S have to be known. In practice, they are either supplied manually by a human expert or they have to be estimated in a process called *system identification* [WB01].

Let $\hat{s}_i^- \in \mathbb{R}^n$ be the *a-priori* estimate of the process state at time step i (assuming knowledge on the process prior to step i). Further, let $\hat{s}_i \in \mathbb{R}^n$ be the *a-posteriori* state estimate at the same time step, given the measurement z_i. The main task of the Kalman filter is to estimate $\hat{s}_i$. Both the *a-priori* and *a-posteriori* estimation errors are defined as

$$e_i^- = s_i - \hat{s}_i^- \tag{3.9}$$
$$e_i = s_i - \hat{s}_i\ . \tag{3.10}$$

The respective error covariance matrices are defined as

$$P_i^- = E[e_i^- e_i^{-T}] \tag{3.11}$$
$$P_i = E[e_i e_i^T]\ . \tag{3.12}$$

The *a-posteriori* state estimate $\hat{s}_i$ can be modeled as a linear combination of the a *priori estimate* $\hat{s}_i^-$ and a weighted difference between the measurement at the current time step z_i according to equation 3.6 and the predicted measurement $G\hat{s}_i^-$:

$$\begin{aligned} \hat{s}_i &= \hat{s}_i^- + K\nu_i \\ &= K(z_i - G\hat{s}_i^-)\ . \end{aligned} \tag{3.13}$$

The $n \times m$-matrix K is called the *gain* or *blending factor* of the filter, whereas ν_i represents the *measurement innovation*, which reflects the difference between actual and predicted measurement. In order to find an optimal solution for equation 3.13, the gain K has to be chosen such that P_i is minimized.

According to [WB01], minimization can be accomplished by substituting equation 3.13 into 3.10, inserting the resulting form into equation 3.12, performing the indicated expectations, taking the derivative of the trace of the result with respect to K, setting the result equal to zero and then solving for K. More details can be found in [May82], [BH92] and [Jac96]. One possible solution for K which minimizes P_i is given by

$$K_i = \frac{P_i^- G^T}{G P_i^- G^T + S}\ . \tag{3.14}$$

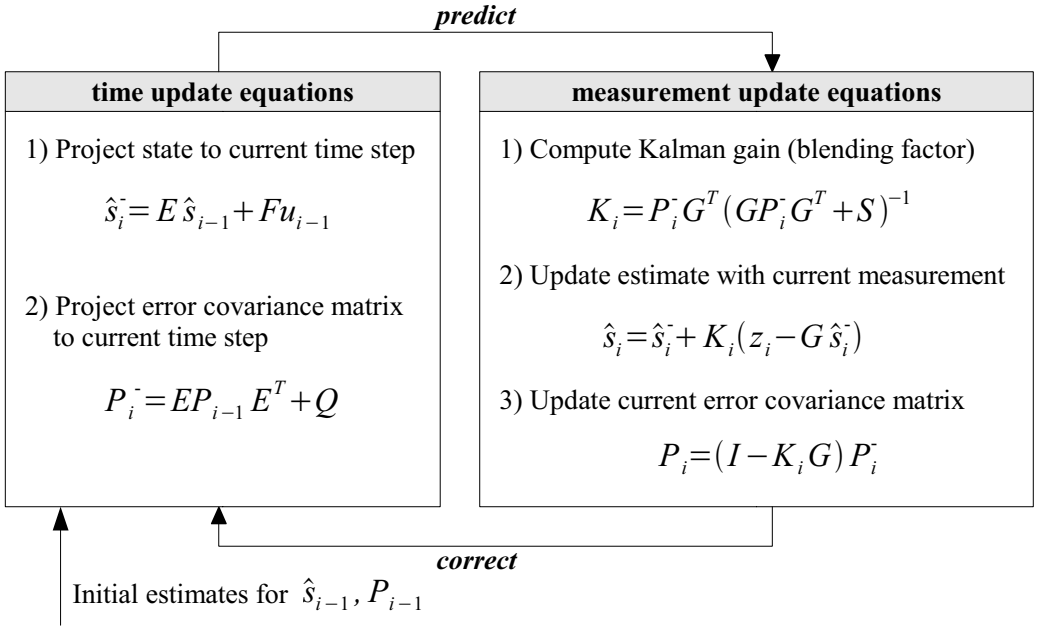


Figure 3.1: *Predictor-corrector* cycle of the Kalman filter. The time update (*left*) predicts both state and error covariance for the current time step i on the basis of the previous state, while the measurement update (*right*) integrates the current measurement into the estimate.

As noted in [WB01], all of the Kalman filter equations can be algebraically transformed into numerous forms. The given equations merely represent one possible (and popular) form. If the measurement error covariance S is small, the gain K weights the residual more heavily. If S approaches zero,

$$\lim_{S_i \to 0} K_i = G^{-1} \,, \tag{3.15}$$

meaning that the actual measurement z_i is trusted more and more, while the prediction $G\hat{s}_i^-$ is trusted less and less.

If, on the other hand, P_i^- is smaller, the gain K weighs the residual less heavily. If P_i^- approaches zero,

$$\lim_{P_i^- \to 0} K_i = 0 \,, \tag{3.16}$$

meaning that the actual measurement z_i is trusted less and less, while the predicted measurement $G\hat{s}_i^-$ is trusted more and more.

Analogously to state prediction and update, the same can be formulated for the error covariance matrix P_i. The *a-priori* estimate P_i^- (which is the prediction of P_i) may be computed by transforming the estimate covariance of the previous time step and adding

the process noise covariance Q, such that

$$P_i^- = EP_{i-1}E^T + Q \; . \tag{3.17}$$

The update for P_i can be formulated as

$$P_i = (I - K_i G)P_i^- \; , \tag{3.18}$$

where I represents the identity matrix.

Generally, the Kalman filter equations may be divided into two groups: *time update* and *measurement update* equations. The former are responsible for the forward projection in time of the current process state and error covariance estimate in order to obtain the *a-priori* estimates of the next time step. The latter are responsible for providing feedback in the form of a new measurement to the estimation, in order to obtain an improved *a-posteriori* estimate. The Kalman filter can also be seen as a *predictor-corrector algorithm*, where the time update equations provide a prediction of the state and the measurement update equations enable a correction of the predicted state. Figure 3.1 illustrates the *predict-correct* cycle of the filter and shows the corresponding update equations. An overview of filter extensions such as the extended or unscented Kalman filter can be found in [WB01].

3.2.4 Kanade-Lucas-Tomasi Tracker

The well-known Kanade-Lucas-Tomasi (KLT) tracker is widely used in computer vision applications. The origins of the tracker are found in the early work of Lucas and Kanade [LK81], where an image registration method was proposed, based on the computation of the optical flow between two frames. Later, the method has been extended to the problem of tracking single feature points through image sequences [TK91]. In [ST94], an extension to the original method was proposed with regard to motion modeling: Instead of a purely translational model, an affine extension was used, leading to significantly improved results.

The goal of the algorithm is to align a template window $I(\mathbf{x})$ to a target window $T(\mathbf{x})$ such that the dissimilarity between both is minimized. Figure 3.2 (left) shows an exemplary target window (solid rectangle), superimposed on the first image of the *carton*-sequence. For each shift of the template window around the position of the target window (dashed rectangle), a different error ε results (right). In this example, a simple translation model has been used, *i.e.* the shape and size of the window remain constant. Naturally, more complex models can be used as well (*e.g.* an affine transformation).

In the following, the major components of the KLT tracking algorithm are introduced and explained, mainly based on [ST94]. Note that the major equations are presented in continuous form. Thus, the discrete time index i is replaced by t and τ respectively.

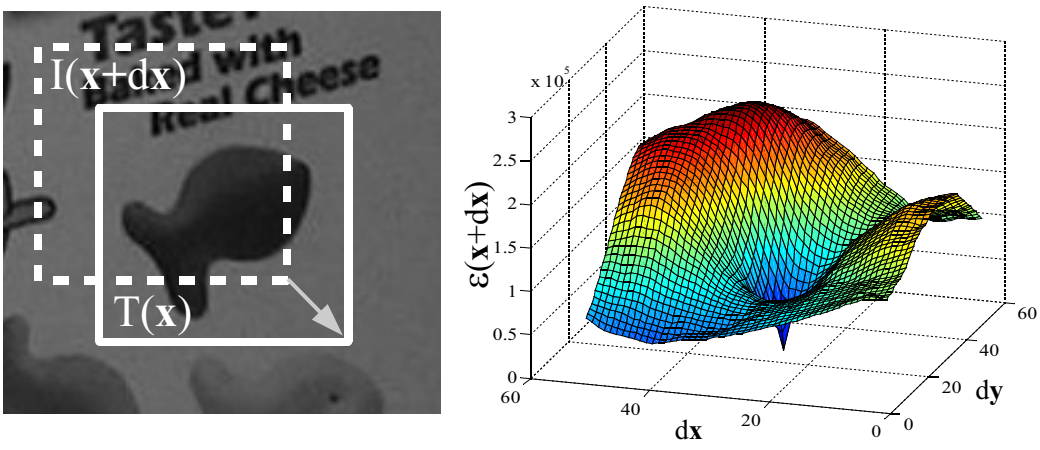


Figure 3.2: On the left, an example target window $T(\mathbf{x})$ (solid rectangle) is shown, superimposed on the first image of the *carton*-sequence. For each shift of the template window $I(\mathbf{x} + d\mathbf{x})$ around the position of the target window, a different error ε results (right). The respective $d\mathbf{x} = [dx, dy]^T$ gives the feature translation in both coordinate directions. In the shown example, a pronounced minimum of the error function exists at $d\mathbf{x} = [25, 25]^T$.

KLT-Algorithm. In the case of camera motion and/or moving objects within the observed scene, changes of the image intensity I occur. According to

$$I(x, y, t + \tau) = I(x - \xi(x, y, t, \tau), y - \eta(x, y, t, \tau)) \,, \tag{3.19}$$

an image $I(t)$ taken at time t can be transformed into another image $I(t+\tau)$ by shifting every point $\mathbf{x} = [x, y]^T$ by a certain amount in both coordinate directions. The latter is referred to as the displacement $\sigma = (\xi, \eta)$ of the point. However, if σ is different for every pixel location, a better representation is an *affine motion field* of the form

$$\sigma = D\mathbf{x} + \mathbf{d} \,, \tag{3.20}$$

where D is a 2×2-deformation matrix and $\mathbf{d}$ describes the translation of the window center. The motion of a point in image $I(t)$ into another image $I(t + \tau)$ can then be described by

$$I(A\mathbf{x} + \mathbf{d}, t + \tau) = I(\mathbf{x}, t) \,, \tag{3.21}$$

where $A = \mathbf{1} + D$ and $\mathbf{1}$ denotes the 2×2-identity matrix. In this case, tracking can be defined as the task of determining the six parameters in D and $\mathbf{d}$. Naturally, the quality of the estimate depends on the size of the support window W around a position $\mathbf{x}$: For a small window, parameter estimation is less accurate in most cases than for a large window due to the smaller number of measurements and the smaller amount of variation. Thus, with regard to parameter estimation, a large window size is preferable.

In the case of depth discontinuities (*e.g.* at object borders) however, a small window is more adequate. For this reason, Shi and Tomasi propose to use a pure translational model of the form $\sigma = \mathbf{d}$ for tracking, where the deformation matrix D is assumed to be zero. The two parameters of such a model are significantly easier to estimate. An affine model according to equations 3.20 and 3.21 is used additionally between the first frame of appearance of a feature and the current frame in order to monitor its quality over time.

Due to noise and deviations of the real scene from the affine model, equation 3.21 is usually not exactly satisfied. Thus, the problem of finding the correct model parameter can be expressed as minimizing the dissimilarity

$$\varepsilon = \int \int_W [I(A\mathbf{x} + \mathbf{d}, t + \tau) - I(\mathbf{x}, t)]^2 w(\mathbf{x}) d\mathbf{x} \,, \tag{3.22}$$

where $w(x)$ is an optional weighting function, *e.g.* a Gaussian in order to emphasize the central area of the support window W. If only the translation $\mathbf{d}$ is sought, A is constrained to equal the identity matrix. In order to find the set of parameters that minimizes ε, equation 3.22 has to be differentiated with respect to all six unknown parameters and setting the result to zero. The system is then linearized by a truncated Taylor expansion. For full detail on the resulting equations, the reader is referred to [ST94].

Between adjacent frames, the affine deformation is most likely to be small. In this case, the matrix D is set to zero, as the estimation of these parameters can lead to a poor solution of the displacement vector $\mathbf{d}$ as well. This is also a significant limitation of the KLT-tracker: It can only provide acceptable results, if the motion of features between neighboring frames can be described sufficiently by a pure translation. If this was not the case, the algorithm would most likely fail. Between the first frame of appearance of a feature and the current frame, all six parameters are estimated instead, in order to monitor the quality of the respective track. In this case, motion changes are too large in order to be represented by a pure translational model.

In order to identify suitable features which can be reliably used for tracking, Shi and Tomasi rely on the well-known structure tensor according to equation 2.1. They show, that a feature can be tracked well if the matrix $\mu(\mathbf{x}, \sigma_i, \sigma_d)$ exhibits two large eigenvalues λ_1 and λ_2. The variables σ_i and σ_d define the size of a Gaussian smoothing kernel and thus the support area W. A feature is accepted for tracking, if it fulfills the condition

$$\min(\lambda_1, \lambda_2) > \lambda \,, \tag{3.23}$$

where λ is a pre-defined threshold. However, not every feature which shows a high texture variation within its support area is suited for tracking. If two objects were to occlude each other in the three-dimensional world (*e.g.* two twigs of a tree), a camera would observe a two-dimensional intersection with no information on the corresponding depth. Such a feature would be selected for tracking according to the demands of equation 3.23. If the tree branches were to move slightly due to wind, the corresponding

feature motion in the image sequence would not correspond to the changes in the world scene. If, however, the affine deformation parameters are estimated as well, they can be used to monitor the quality of the feature over time. In this case, the residual error in equation 3.22 would indicate a growing dissimilarity and thus exclude the feature from further tracking.

In order to extend the range of the tracker beyond the size of the support area, a *pyramidal implementation* may be used, according to [Bou00]. For a generic image $I_{(0)}$ of size $n_x \times n_y$ (which denotes the original image), it may be build in a recursive fashion: The second layer of the pyramid $I_{(1)}$ is computed from $I_{(0)}$, the third layer $I_{(2)}$ is computed from $I_{(1)}$ and so on. Given several layers of I, *pyramidal feature tracking* can be performed by processing the top-most layer $I_{(L)}$ first. The resulting features are then passed to the next layer $I_{(L+1)}$, where a refined estimation is performed. Features, which did not exist on layer $I_{(L)}$ yet due to insufficient size, are newly added to the list of features. The results are propagated further down the pyramid, until the deepest layer $I_{(0)}$ is reached. In this manner, features of different sizes and with large displacement may be tracked as well.

3.3 Concepts for Descriptor-based Region Tracking

In this section, three concepts for the descriptor-based tracking of affine-covariant regions are presented. As already discussed in section 2.5.4, the euclidean distance d_d of two descriptors according to equation 2.38 is used as a measure of similarity between the regions of a pair of images. The first two algorithms presented here provide a set of trajectories (or *tracks*) which are further extended in length with every new frame (which is a favorable property in the case of time-critical applications). While the first tracking method is based on finding region correspondences within an equidistant gating region around the location of the most recently added element of a track (the *track head*), the second algorithm is based on a set of Kalman filters (KF), which estimate the presumed location of features in future frames based on the history of the associated trajectories. Additionally, a measure of reliability in form of a covariance matrix is provided. By exploiting information from an eigen-analysis of this matrix, the gating region for correspondence assignment can be significantly narrowed compared to the first method. Both methods enforce uniqueness among the feature correspondences, *i.e.* potential ambiguities are resolved for each new frame pair such that every region in the first frame is assigned to exactly one region in the second frame.

Instead of resolving ambiguities on the basis of two frames only, the third proposed tracking method keeps *all* potential correspondences for a predefined number of frames and models these into a directed and weighted graph. Uniqueness is enforced by means of shortest-path search between start-nodes and end-nodes of a graph. The advantage of this method toward the other two is the availability of all the information contained in the

sequence during the decision process and thus an increased reliability when it comes to extracting the final trajectories. A clear disadvantage however is the belated availability of usable tracks, especially in cases where updates on feature positions are needed with each new frame.

Although all three methods are used for tracking affine-covariant regions, the latter are mostly termed *features* in the following in order to avoid confusions when it comes to describing the *gating regions* during correspondence assignment, which define the maximally permissible disparity between two frames.

3.3.1 Multi-Region Tracking Using a Constant Gating Region

The algorithm presented in this section consists of three major steps, which are illustrated in figure 3.3. Firstly, *data association* is performed in order to find initial (and potentially ambiguous) correspondence candidates for each neighboring frame pair of a sequence. Secondly, a subset of these candidates is selected such that only unique correspondences exist and that the overall descriptor distance of this subset is minimal. These two goals are achieved by means of *combinatorial optimization*. Thirdly, the resulting set of unique feature correspondences is appended to already existing tracks or - if no *track affiliation* exists yet - are used to initialize new ones. In the following, each step of the algorithm is explained in detail. A significant advantage of this comparably simple tracking algorithm lies in the availability of ready-to-use trajectories, which are further extended with each new frame.

Data Association. In order to find all potential correspondences for a pair of images, features within a gating distance d_G of each other are compared with regard to their descriptor distance d_d. All pairings for which the relation $d_d \leq d_{d,max}$ holds are added to a list of correspondence candidates $\mathcal{L}$. The size of d_G bounds the maximally permissible feature velocity, but does neither contain any further assumption on the expected type of motion nor information on the track history of a feature, which might be used to predict the presumed new position and thus to narrow the search area. In figure 3.4, a typical example for data association is shown: The foremost elements of two trajectories - termed *track heads* in the following - are initially assigned to all potential feature candidates (black circles) within the respective gating region. The two features lying within the overlap area may belong to either trajectory, given a sufficiently small descriptor distance. As indicated in figure 3.3, the set of all correspondence candidates $\mathcal{L}$ may well contain such ambiguous assignments. In the data association step however, dependencies between competing trajectories and candidate features are not yet resolved. This is done in the subsequent optimization step. In the end, each track head may only be assigned to a single new feature.

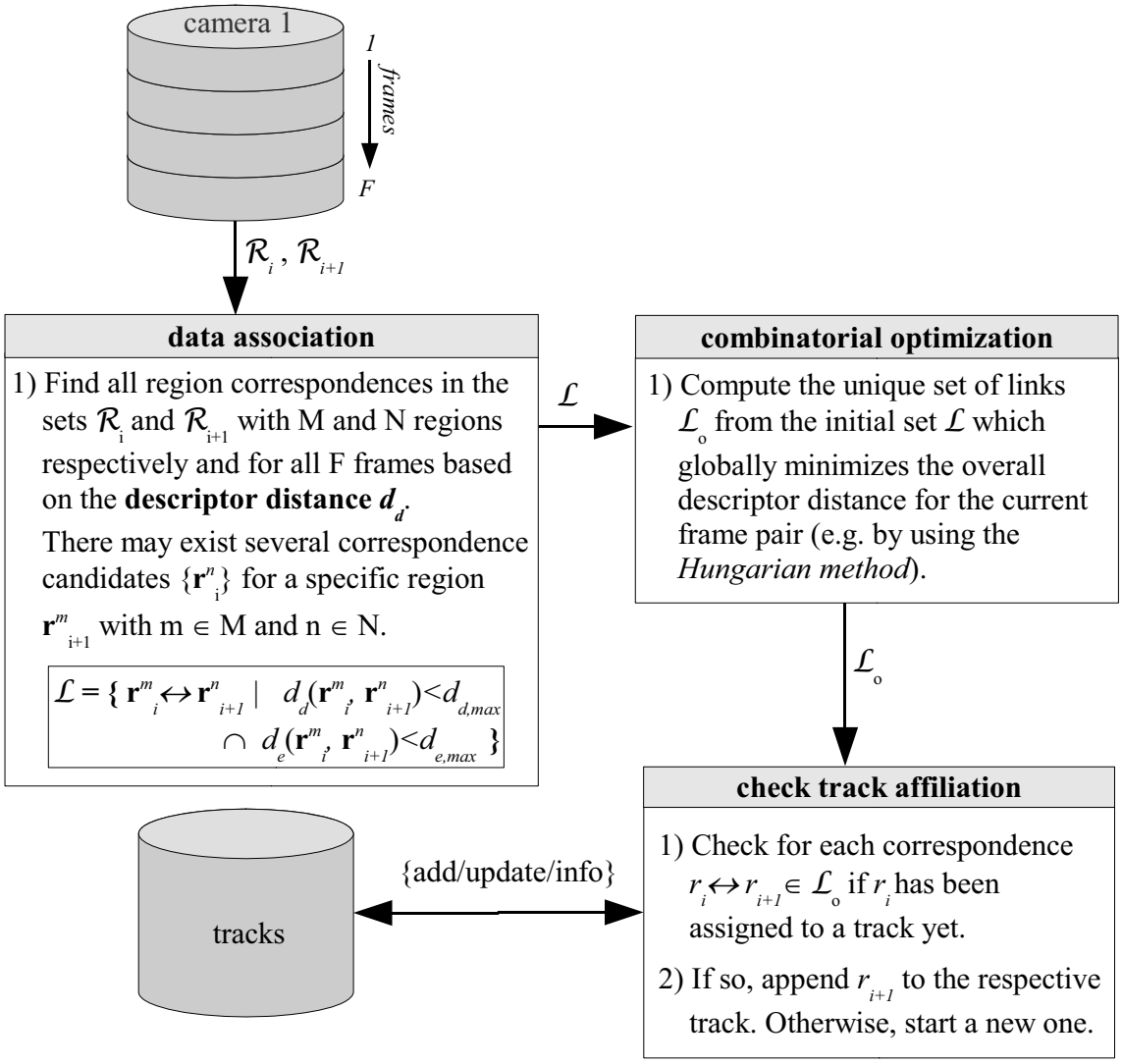


Figure 3.3: Algorithmic flow of multi-region tracking using a constant gating region: Firstly, data association within a gating region of circular shape and constant size is performed in order to find a set of initial (and potentially ambiguous) correspondence candidates $\mathcal{L}$. Secondly, a subset $\mathcal{L}_0$ of these candidates is extracted such that only unique correspondences exist and the overall descriptor distance of $\mathcal{L}_0$ is minimal. These two goals are achieved by means of combinatorial optimization using the Hungarian method. Thirdly, the resulting correspondences are appended to already existing tracks or, if no track affiliation exists yet, are used to initialize new ones.

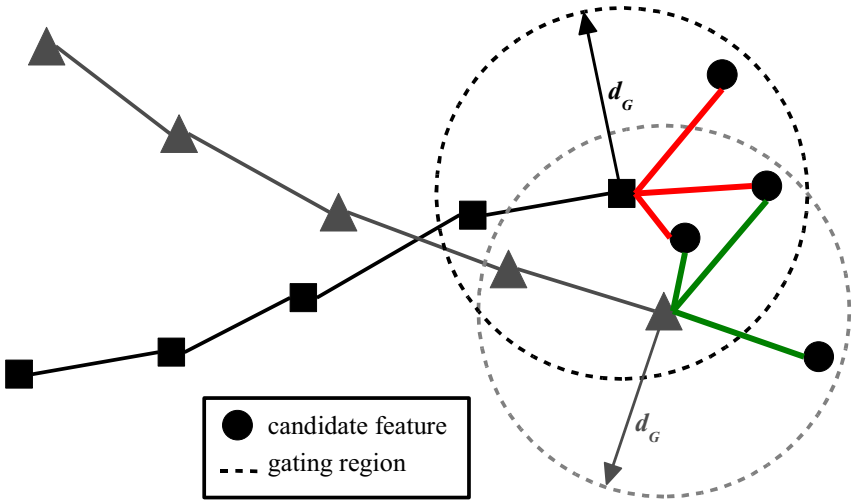


Figure 3.4: Illustration of data association with circular gating regions of predefined radius d_G: Two tracks (indicated by squares and triangles) are initially assigned to a set of candidate features (circles). Within the overlapping area of both gating regions, assignment ambiguities may arise, especially in the case of descriptors with low distinctiveness.

Combinatorial Optimization. As illustrated in figure 3.4, assignment ambiguities between trajectories and feature candidates may easily arise, if descriptors within the gating region are too similar. There may occur two cases: *Firstly*, an already assigned feature in the current frame may find several potential correspondences in the next frame. In this case, the most similar candidate (with regard to d_d) is to be chosen while the remainder of dependent candidates is discarded. This approach is generally referred to as *nearest-neighbor assignment*, because the final feature correspondences are nearest to each other in terms of their euclidean distance in descriptor space. *Secondly*, a feature in the next frame may additionally be claimed by several features in the current frame. In this case, the simple nearest-neighbor assignment is no longer guaranteed to provide the best possible results. Depending on the processing order of trajectories and feature candidates and on the dependencies among them, it is well possible that the local decision on the optimal candidate for a specific trajectory might entail several sub-optimal assignments with other trajectories in the processing queue. As will be seen later, this is especially a problem with high feature densities in a local neighborhood. Thus, it would be more favorable to find a global solution to the trajectory-feature assignment problem for each pair of frames, which considers all potential pairings at the same time. To this end, a combinatorial optimization approach is needed, which analyzes the interconnections between the features of two frames and at the same time provides *the* subset of correspondences which globally minimizes the overall d_d. In the previous chapter, such

a method has already been used in the context of finding feature correspondences with minimal geometrical overlap, based on the well-known *Hungarian algorithm*. The basic idea of the algorithm is to model an assignment problem as a cost matrix C, whose entries represent weighted interconnections between the elements of two sets (*e.g.* the features in two frames). By iteratively processing the elements of C, an optimal assignment solution can be reached in polynomial time $O(n^3)$, where n is the number of rows in C. Due to the relatively high computational complexity, it is essential to divide the entire set of features into independent subgroups with a smaller n, which can be processed independently (and thus more effectively). A detailed description of the algorithm can be found in section 2.5.3. The result of the combinatorial optimization procedure is a reduced set of unique correspondences $\mathcal{L}_0$ with minimal overall descriptor distance.

Track Affiliations. For each newly detected feature correspondence, the affiliation to existing tracks must be checked. If a feature in the first frame is already part of a track, the corresponding feature in the second frame may then be easily appended to it as the new track head. If no affiliation exists, a new track is initialized with the corresponding feature pair as its first two elements. Following from this concept, tracks may start (and end) at arbitrary points during the sequence. As indicated in figure 3.3, all tracks are stored in a central database, to which either request or modifications are made by the track affiliation stage. At each time step, the set of existing tracks may be freely accessed by subsequent algorithms in order to obtain information on the current and previous motion of features in the scene. It is in the responsibility of the track affiliation stage to update and modify this database with each new frame.

Two significant advantages of the presented method are its comparatively low computational complexity and the ability to provide a ready-to-use set of trajectories at every time instant. With each new frame, the set is further extended by associating all newly detected regions on the basis of their similarity to the already existing track heads. However, based on the distinctiveness of the used region descriptors, assignment errors may occur, leading to potentially gross position errors. Based on the expected object motion (and thus on the size of the gating region), the probability for the occurrence of such outliers may differ significantly, depending on the application. This disadvantage is compensated for by the subsequently presented method, which allows for a prediction of the presumed position of a feature in the subsequent frame. Depending on the accuracy of this prediction, the gating region size and thus the probability for the occurrence of outliers may be significantly reduced.

3.3.2 Multi-Region Tracking Using Kalman-Filtering

In the previous section, a method has been presented for feature tracking which considers all correspondence candidates within a circular gating region with radius d_G around each track head for data association and decides on *the* pair of features whose associated de-

scriptors are nearest neighbors with regard to their euclidean distance in descriptor space. In this section, a more refined method is proposed, which uses a Bayesian framework for state-filtering as described in background section 3.2.2 to predict the presumed location of a feature in future frames. Thus, the size of the gating region can be lowered significantly, assuming a sufficiently accurate estimate. Also, the method provides a measure for the uncertainty of its state estimates in form of a covariance matrix, which may be used in conjunction with descriptor distance to additionally weigh potential feature correspondences, based on their deviation from the predicted position. In the proposed implementation, a set of standard Kalman filters has been used. Their use is justified by both a linear process model and a near-Gaussian distribution of position errors for the evaluated affine-covariant detectors. Preconditions for the use of specific representatives of Bayesian filtering approaches (*e.g.* KF/EKF/UKF or particle filters) have been previously discussed in section 3.2.2. Given a matrix of error covariances supplied by each KF, a suitable distance measure which weighs the deviation of a feature from a predicted position is the *Mahalanobis distance* (introduced later in this section). Figure 3.6 shows two tracks (indicated by solid squares and triangles), the predicted feature locations resulting from the respective track histories (framed square and triangle) and several new candidate features (circles). The dashed ellipses illustrate the uncertainty of the respective position estimates, based on the state covariance information. Axis direction and length have been obtained from an eigenvalue analysis of the latter. Compared to figure 3.4 in the previous section, the gating region and thus the number of potential correspondences has been significantly narrowed in this example. In the following, the different steps of the algorithm according to the illustration in figure 3.5 are explained in detail. Firstly, the state transition model which contains *a-priori* information on the expected object motion is introduced. Secondly, issues of filter initialization are discussed, including the distribution of position errors for each of the five detectors from the previous chapter. Thirdly, the filter update methodology is introduced, including an appropriate distance measure based on the Mahalanobis distance for assessing the deviation of candidate features from the predicted filter states.

Constant-Velocity State Transition Model. Based on the linear process model defined in equation 3.5, the system state $\mathbf{s}_i$ in frame i of the Kalman filter described in section 3.2.3 is defined as

$$\mathbf{s}_i = \begin{bmatrix} p_{x,i} \\ p_{y,i} \\ v_{x,i} \\ v_{y,i} \end{bmatrix} = \begin{bmatrix} p_{x,i-1} + v_{x,i-1}\Delta i + \frac{1}{2}a_x \Delta i^2 \\ p_{y,i-1} + v_{y,i-1}\Delta i + \frac{1}{2}a_y \Delta i^2 \\ v_{x,i-1} + a_x \Delta i \\ v_{y,i-1} + a_y \Delta i \end{bmatrix}, \tag{3.24}$$

where the first two entries $p_{x,i}$ and $p_{y,i}$ represent the predicted feature position in the image plane while $v_{x,i}$ and $v_{y,i}$ denote its velocity in both coordinate directions.

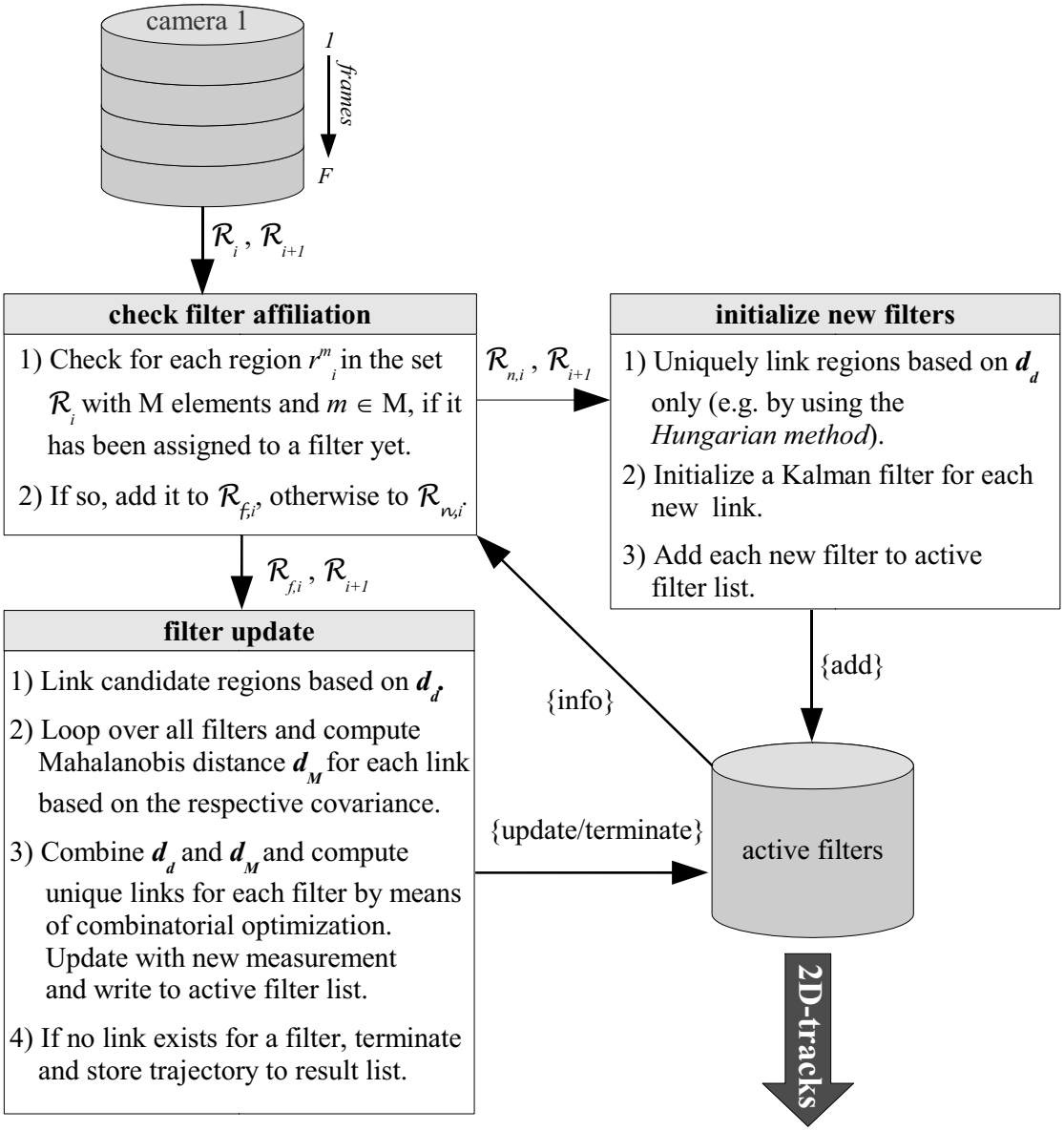


Figure 3.5: Algorithmic flow of multi-region tracking using Bayesian filtering: The basic idea is to initialize for each newly detected region a proprietary Kalman-filter, which is used for the prediction of the presumed position within the next frame. The reliability of the prediction is expressed in terms of the Mahalanobis distance, which uses a covariance matrix to compute a non-euclidean distance measure. The latter is combined with descriptor similarity as a joint measure for correspondence assignment. As with the previously introduced tracking method in section 3.3.1, uniqueness is enforced using a global optimization technique, given the previous and current frame. The diagram shows the components of the proposed tracking method.

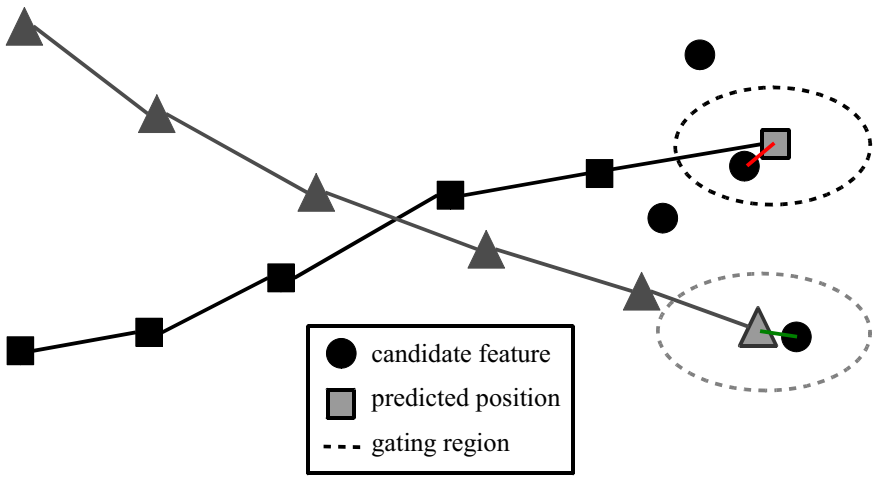


Figure 3.6: Illustration of data association using Bayesian filtering: For both tracks (indicated by solid squares and triangles), the presumed location of the next feature (framed square and triangle) is predicted using a Bayesian filtering framework. The deviation from the estimated position is used to obtain a further distance measure in addition to descriptor distance. The uncertainty of the estimate is modeled using both eigenvectors and eigenvalues of the state covariance matrix, which provides an ellipsoidal gating region of varying shape and size.

The state transition matrix E which relates the process state $\mathbf{s}_{i-1}$ to the new state $\mathbf{s}_i$ and the process noise $\mathbf{w}_i$ thus take the form

$$\mathbf{s}_i = \underbrace{\begin{bmatrix} 1 & 0 & \Delta i & 0 \\ 0 & 1 & 0 & \Delta i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{E} \mathbf{s}_{i-1} + \underbrace{\begin{bmatrix} \frac{1}{2} a_x \Delta i^2 \\ \frac{1}{2} a_y \Delta i^2 \\ a_x \Delta i \\ a_y \Delta i \end{bmatrix}}_{\mathbf{w}_i} . \tag{3.25}$$

Wile the velocity is assumed constant, the acceleration a is modeled as an additional noise term.

The state covariance matrix P_i according to equation 3.12, which models the uncertainty of the state estimates is defined as

$$P_i = \begin{bmatrix} \sigma_{p_x p_x} & \sigma_{p_x p_y} & \sigma_{p_x v_x} & \sigma_{p_x v_y} \\ \sigma_{p_y p_x} & \sigma_{p_y p_y} & \sigma_{p_y v_x} & \sigma_{p_y v_y} \\ \sigma_{v_x p_x} & \sigma_{v_x p_y} & \sigma_{v_x v_x} & \sigma_{v_x v_y} \\ \sigma_{v_y p_x} & \sigma_{v_y p_y} & \sigma_{v_y v_x} & \sigma_{v_y v_y} \end{bmatrix} . \tag{3.26}$$

Here, the frame subscript i has been omitted for the sake of readability.

Let further P_a be the covariance of the acceleration $\mathbf{a} = [a_x, a_y]^T$, *i.e.*

$$P_a = \begin{bmatrix} \sigma_{a_x}^2 & 0 \\ 0 & \sigma_{a_y}^2 \end{bmatrix} . \tag{3.27}$$

Projecting P_a into the domain of the process noise $\mathbf{w}_i$ according to equation 3.8 with covariance Q_i yields

$$\begin{aligned} Q_i &= \frac{\partial \mathbf{w}_i}{\partial \mathbf{a}} P_a \frac{\partial \mathbf{w}_i}{\partial \mathbf{a}}^T \\ &= \begin{bmatrix} \frac{1}{4}\Delta i^4 \sigma_{a_x} & 0 & \frac{1}{2}\Delta i^3 \sigma_{a_x} & 0 \\ 0 & \frac{1}{4}\Delta i^4 \sigma_{a_y} & 0 & \frac{1}{2}\Delta i^3 \sigma_{a_y} \\ \frac{1}{2}\Delta i^3 \sigma_{a_x} & 0 & \Delta i^2 \sigma_{a_x} & 0 \\ 0 & \frac{1}{2}\Delta i^3 \sigma_{a_y} & 0 & \Delta i^2 \sigma_{a_y} \end{bmatrix} . \end{aligned} \tag{3.28}$$

(3.29)

Finally, the measurement vector is defined as

$$\mathbf{z}_i = \begin{bmatrix} m_{x,i} \\ m_{y,i} \end{bmatrix} , \tag{3.30}$$

where $m_{x,i}$ and $m_{y,i}$ denote the measured (noisy) feature position in frame i. As the feature velocity is not determined directly, the matrix G in equation 3.6 which models the relation between states and measurements is set to

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} . \tag{3.31}$$

Filter initialization. In its original formulation, the Kalman filter (KF) is capable of estimation the state of a *single* process from noisy measurements by following the predictor-corrector cycle shown in figure 3.1. With each new arriving measurement, the system state is updated and used to refine the next state prediction. In the case of multiple measurements, the one which is closest to the prediction is assigned to the filter. If no prior track history for two corresponding regions exists, a new KF is initialized as follows:

- A unique set $\mathcal{L}_o$ of feature correspondences is generated on the basis of the combinatorial optimization algorithm also used in the previous section 3.3.1.

- For each feature pair $r_i^m \leftrightarrow r_{i+1}^n$ in the set $\mathcal{L}_o$ (where $m \in M$ and $n \in N$ with M, N as the number of features in the respective frames), an initial state vector is computed as

$$\mathbf{s}_1 = \begin{bmatrix} p_{x,2} \\ p_{y,2} \\ p_{x,2} - p_{x,1} \\ p_{y,2} - p_{y,1} \end{bmatrix} , \tag{3.32}$$

  where $\mathbf{p}_i = [p_{x,i},\ p_{y,i}]^T$ denotes the location of both features of a specific correspondence.

- The initial measurement vector is accordingly set to

$$\mathbf{z}_1 = \begin{bmatrix} p_{x,2} \\ p_{y,2} \end{bmatrix} . \tag{3.33}$$

- The measurement noise S according to equation 3.8 is estimated from the distribution of position errors d_l for each detector individually. Estimates are obtained by using the homography-based groundtruth correspondences as described in section 2.5 of the previous chapter. In figure 3.7, the respective distributions of d_l are shown. For each detector, both mean and variance have been additionally superimposed (red curves) in order to illustrate the near-Gaussian shape of each distribution, which justifies the use of a KF in its original form. The respective standard deviations σ are used as an estimate of S.

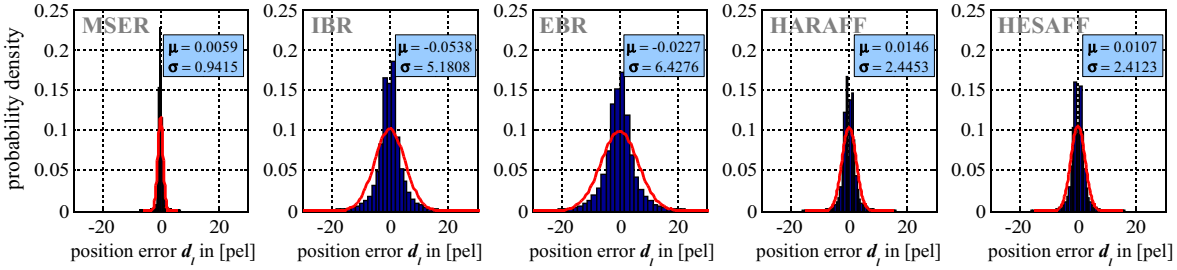


Figure 3.7: Distribution of position errors d_l for all five detectors. Normal distributions with equivalent mean and standard deviation have been superimposed on the histograms (red curves).

- As noted in [WB01], the estimation of the process noise covariance Q is generally more difficult compared to S, as there is no possibility to directly observe the estimated process. In practice, Q is often tuned manually by a human expert or estimated in a separate process called *system identification* [Bal87]. In [Wue04], the influence of different settings of Q has been analyzed in the context of tracking. In his experiments, the author could show that a small process noise generally lead to good state estimates in the case of a stable true state. If the latter changes during the sequence, a too small setting for Q will introduce an undesired latency into the filter. A too large setting of Q however, will lead to a better response time at the cost of noisier state estimates. The author recommends that in cluttered environments which contain multiple measurements, a small process noise covariance should be chosen so that the filter does not accidentally switch to a wrong track. In practice, a setting of $\sigma_{a_x} = \sigma_{a_y} = 0.1$ has been used in equation 3.29 as a reasonable compromise, which has proved sufficient in the performed experiments.

- For the initial error covariance P_1 according to equation 3.12, a large setting is generally desirable over a small one. If P_1 is not large enough, the resulting Kalman gain i in equation 3.14 will remain very small during the update process and thus will the influence of the actual measurements on state estimation. If P_1 is chosen too large instead, convergence to an acceptable level usually occurs after a few iterations of the predictor-corrector cycle. In the underlying implementation, P_1 has been set according to the standard deviations of the distributions of d_l, as shown in figure 3.7.

Filter update. Once a filter has been initialized, the algorithm tries to update its state with a new measurement with every incoming frame. In the first step of the update stage, correspondence candidates between the current and next frame are generated by

means of descriptor-based data association as described in the previous section. As explained in section 3.2.3, the KF provides a means of estimating the error covariance P_i of the estimated state, *i.e.* the uncertainty or trustworthiness of the predicted position of the next feature. Instead of comparing the descriptor distance d_d to all features within a gating region around the *current* position (as with the algorithm introduced in the previous section), the proposed method uses the *a-priori* state $\hat{\mathbf{s}}_i^-$ to reach an estimate of the presumed feature location. As given in figure 3.1, this estimate is based on the *a-posteriori* state vector of the prior time step $\hat{\mathbf{s}}_{i-1}$ and the state transition matrix E. Correspondence assignment may then be performed based on both d_d and the *a-priori* error covariance P_i^-. An appropriate distance measure between the predicted position and a measured position $\mathbf{z}_i$, which models the uncertainty of the prediction, can be defined as

$$d_M = \sqrt{(\mathbf{z_i} - G\hat{\mathbf{s}}_i^-)^T S_i^{-1} (\mathbf{z_i} - G\hat{\mathbf{s}}_i^-)}, \tag{3.34}$$

where the difference $\mathbf{z_i} - G\hat{\mathbf{s}}_i^-$ is the *measurement innovation* and the matrix S_i represents the innovation covariance, which is given according to [BSFC90] as

$$S_i = G P_i^- G^T + S\ . \tag{3.35}$$

The expression in equation 3.34 is also referred to as *Mahalanobis distance* in the literature [Cox93]. It models the uncertainty in the given covariance matrix to compute a probabilistic distance measure (which can be represented as an ellipse as seen in figure 3.6). If the uncertainty of the state estimate in the direction of x was higher than in y-direction, surely a feature with a small deviation in y would be more trustworthy than one with an equally small deviation in x. Accordingly, the Mahalanobis distance were larger for the second case, while a euclidean distance measure would provide identical (and thus inappropriate) results.

The euclidean distance however can be seen as a special case of the Mahalanobis distance where the covariance matrix equals the identity matrix. The expression d_M in equation 3.34 can be converted into euclidean space through normalization with the help of an eigen-decomposition of the inverse covariance: If S_i^{-1} is symmetric positive definite, it can be decomposed into the form $S_i^{-1} = V^T D V$. In this case, D is a diagonal matrix with the eigenvalues of S_i^{-1} on its main diagonal and V is an orthogonal matrix with its eigenvectors as columns. Drawing the square root of the inverted covariance matrix delivers $S_i^{-\frac{1}{2}} = \sqrt{D}V$. Inserting this term into equation 3.34 yields

$$d_M = \left\| \sqrt{D}V\mathbf{z_i} - \sqrt{D}VG\hat{\mathbf{s}}_i^- \right\|\ , \tag{3.36}$$

which corresponds to the euclidean distance of the normalized vectors where $P_i^{-1} = V^T D V$. In order to combine the deviation from the predicted position with the descriptor distance d_d between both features, a combined measure is computed as

$$d_c = d_M d_d\ . \tag{3.37}$$

For each existing KF, a list of candidate measurements is created on the basis of d_c. At this point however, the list may still contain ambiguous assignments which have to be resolved before performing the actual measurement update. By means of combinatorial optimization on the basis of the *Hungarian method* (as described in section 2.5.2), a unique set of correspondences between filters and feature correspondences in the current frame is computed. After the elimination of all ambiguities, each filter is assigned a new measurement in order to update its state according to figure 3.1. If no feature with sufficiently low d_c can be found for a KF or if no candidate is left after the optimization procedure, the respective filter is terminated. From its history of measurements until the current frame, the associated trajectory is extracted and written to the result list as seen in figure 3.5. Alternatively, the filter states could be used as well to obtain a smoothed version of the same trajectory. However, this has not been done in the presented implementation of the algorithm.

3.3.3 A Graph-based Approach to Multi-Region Tracking

Both previously introduced methods decide on unique feature-to-feature correspondences based on descriptor distance and (in the case of KF-based tracking) the deviation from a predicted position on the grounds of the so-far available track history. Every feature has exactly one incoming and/or one outgoing link which can be followed from the first frame of appearance to the last frame in order to assemble all correspondences into a trajectory. Although the number of potential correspondence candidates within the gating region will be lower for KF-based tracking, ambiguities may still occur. In the case of feature disappearances or insufficiently discriminatory descriptors, wrong correspondences might be selected, eventually leading to invalid transitions between neighboring trajectories (deemed *outliers* in the following). Thus, it would be more favorable to preserve all correspondence candidates until the final frame of the image sequence (or for a predefined number of frames) in order to decide on the best possible set of unique correspondences on the basis of all the available information. In this section, such a method is proposed which integrates for every feature all assignment possibilities into a set of *track graphs*, gradually constructed by traversing through the image sequence. Relations (edges) between corresponding features (nodes) are weighted using a combination of descriptor distance and the response of a *path coherence model*, which weighs the motion history of a feature based on a well-known model from the literature. In this work, the latter is further extended by a term that evaluates and penalizes the relative changes of feature scales in order to incorporate the additional information supplied by affine-covariant detectors. Unique trajectories (paths) will be iteratively extracted from each graph by using Dijkstra's method for shortest-path graph traversal [Dij59]. It will be shown in section 3.4.2, that the tracking performance of the presented approach is superior to both other methods with regard to a number of appropriate measures defined in section 3.4.1. Figure 3.8 illustrates the concept of graph-based tracking and the different

steps of the algorithm. In the following, the major components are explained in detail.

Data Association and Graph Affiliation. In order to find all potential correspondences between two adjacent frames, features within a gating distance d_G of each other are compared with regard to their descriptor distances. The data association step is identical to the tracking method as introduced in section 3.3.1. For each frame pair, all potential feature correspondences are kept in a list $\mathcal{L}$, which is passed to the graph affiliation stage of the algorithm. There, each feature in $\mathcal{L}$ is compared against the track graph storage $\mathcal{G}$ in order to determine, if it has been attributed to one of the existing graphs (if any) so far. If so, the correspondence is inserted into the respective graph. Otherwise, a new graph is initialized with the associated features as its first two nodes and written to the storage $\mathcal{G}$. If a feature has been assigned to a different graph already, all elements in both graphs are recursively assigned a common identification number and merged to a single graph. If the end of the image sequence is reached (or if a pre-defined number of frames has been processed), the contents of $\mathcal{G}$ are transferred to the graph traversal stage. Note, that a major difference to the previously introduced tracking methods lies in the data processing order: Instead of providing a valid set of trajectories with each new frame (as with nearest-neighbor and KF-based tracking), the correspondences of the entire frame set are accumulated, so that the resulting trajectories are only available after processing the last frame. Thus, this method is only of limited use for time-critical online tracking systems. It will be shown in the section 3.4 however, that this drawback is countervailed by a number of benefits, such as an improved localization accuracy and a reduced number of tracking outliers.

Track Graph Construction and Traversal. A track graph g is defined as a set of nodes (features) $\mathcal{N}$ and edges (relations) $\mathcal{E}$ between corresponding features. It is represented as an adjacency list A where each entry $A(m)$ for every feature r_i^m in frame I_i contains the respective list of candidate correspondences r_{i+1}^n:

$$A(m) = \{n \in \mathcal{N} | (m, n) \in \mathcal{E}\} \tag{3.38}$$

Once all frames of a sequence have been processed, a set of track graphs is available that holds all feasible relations between features within a distance d_G of each other. If several similar features exist within the gating region bounded by d_G as shown in figure 3.9 (note that for feature 3 in the second frame, there exist 2 corresponding features in the first frame), all correspondence relations are modeled into g.

After construction, each track graph g may contain several trajectories which have to be extracted under the assumption, that each feature may only possess one unique match. A trajectory is defined as a path between a start-node and an end-node of g (i.e. nodes that have neither incoming nor outgoing edges). A transition between two nodes

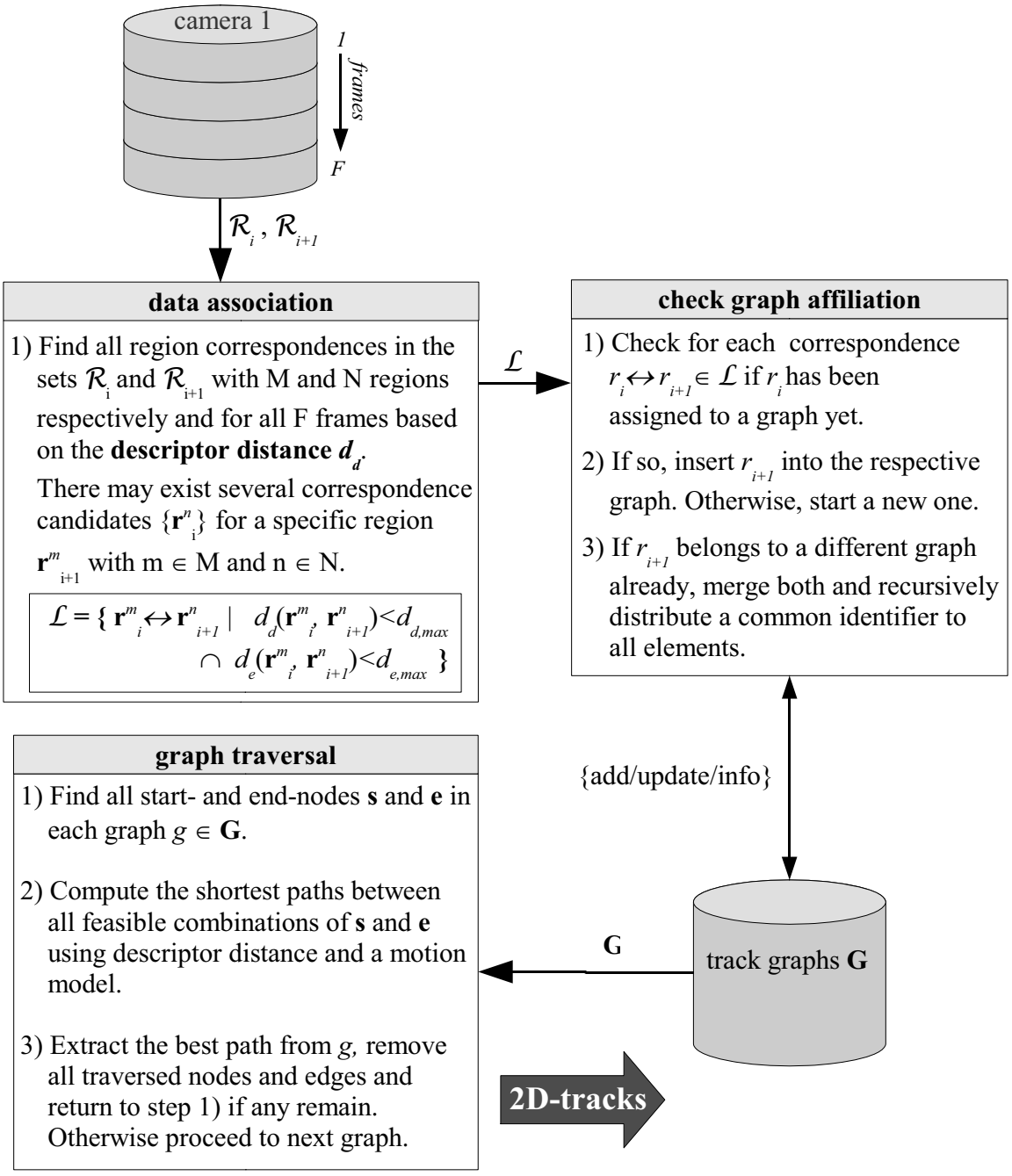


Figure 3.8: Algorithmic flow of graph-based multi-region tracking: Instead of enforcing uniqueness among the set of region correspondences on the basis of two frames alone (as with both prior methods), a set of weighted and directed graphs is constructed instead, which contains all possible (and potentially ambiguous) correspondences from all frames of a sequence (or for a predefined number of frames). Unique trajectories are extracted from these graphs by means of shortest-path search based on both descriptor similarity and a motion model.

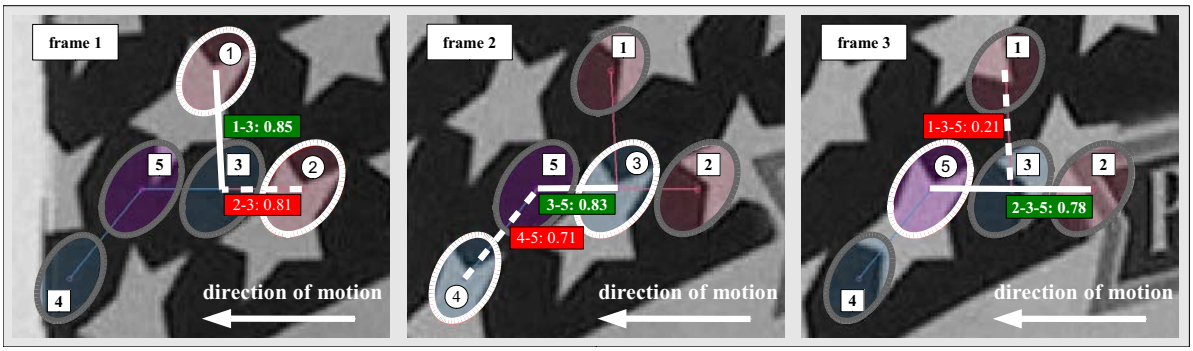


Figure 3.9: Example of a *track graph* existing in three frames. Regions not active in the current frame are grayed, rejected correspondences have a dotted line. *Left+Middle*: With regard to the initial cost function c from equation 3.39, the correspondences $(1, 3)$ and $(3, 5)$ are preferred over $(2, 3)$ and $(4, 5)$ respectively, based solely on descriptor similarity. *Right*: Using the extended cost function c_e from equation 3.44, the (correct) path $(2, 3, 5)$ is chosen as final trajectory.

m and n from frame I_i to frame I_{i+1} is further related with a cost function which, in its simplest form, reflects only the respective descriptor distance:

$$c(m_i, n_{i+1}) = 1 - d_{m,n} \tag{3.39}$$

During graph traversal, the cost of all edge transitions is accumulated and the path with the highest weight is selected as the optimal and thus most likely path. If a graph contains several start- and end-nodes, traversal is performed for all plausible combinations. From the set of all successful traversals, the one with the highest accumulated cost is chosen as the final trajectory.

Path search is performed using *Dijkstra's algorithm* [Dij59], which solves the single-source optimal path problem for directed graphs with non-negative edge weights. Its idea is to store for each node m in a cost table C the maximal cost path found so far originating from a start node s, and in a lookup-table P the indexes of the previously visited nodes. The basic operation of the algorithm is *edge relaxation*: If there is an edge from node m_i in frame I_i to node n_{i+1} in frame I_{i+1} which has the highest weight among all other edges, then the best known path from s to m_i can be extended to a path from s to n_{i+1}. In C, node n_{i+1} will be assigned the sum of the cost of the old path and the edge weight $c(m_i, n_{i+1})$ for the new transition. Additionally, m_i will be stored as its predecessor in P.

$$C[n_{i+1}] = C[m_i] + c(m_i, n_{i+1}) \tag{3.40}$$

$$P[n_{i+1}] = m_i \tag{3.41}$$

If an end-node e is reached, the optimal path can be easily reconstructed by following the entries of P back to s. The resulting overall cost is found from a simple table lookup in $C[e]$. Under the assumption that a specific feature can be assigned to a single trajectory only, all nodes contained in the extracted path are finally removed from g. Among the remaining nodes and edges in g, dependencies are then analyzed anew in order to identify independent sub-graphs. These are processed again iteratively until all nodes have either been assigned to trajectories or have become isolated.

Experiments have shown that, in cases of high feature density and low discriminatory power of the descriptors, the number of nodes in a single graph quickly increases, leading to complex and thus costly traversals. As will be seen in section 3.4.2, this is especially a problem for the EBR-detector. Also, spurious matches may occur as illustrated in figure 3.9 (*left*), where $c(1_1, 3_2) > c(2_1, 3_2)$. A solution to these problems is to extend the cost function in equation 3.39 by a term which exploits the fact that, due to inertia, the motion of observed objects cannot change instantaneously, given a sufficient frame rate. This assumption, often referred to as *path coherence*, has previously been used effectively in tracking applications such as in [SJ87], [VRB01] or [CV99].

According to [SJ87], the guiding principles for modeling a suitable path coherence function are:

- The function should not be negative.
- The function should consider the amount of deviation in the direction of motion, not its sense (*i.e.* left or right). Thus, the sign of the angle of deviation should not factor into its computation.
- The function should equally respond to increases and decreases in velocity.
- If there is no change in motion characteristics, the function should attain its maximum.

Obeying these principles, the following expression has been modeled, which penalizes changes in both velocity and direction:

Based on the center positions p of three affine-covariant regions r_{i-1}^{l}, r_{i}^{m} and r_{i+1}^{n} in consecutive frames, path coherence is computed as

$$\begin{aligned} m &= 1 - w_1 \left[1 - \frac{(p_{l,i-1} - p_{m,i}) \cdot (p_{m,i} - p_{n,i+1})}{\|p_{l,i-1} - p_{m,i}\| \|p_{m,i} - p_{n,i+1}\|} \right] \\ &\quad - w_2 \left[1 - 2 \frac{\sqrt{\|p_{l,i-1} - p_{m,i}\| \|p_{m,i} - p_{n,i+1}\|}}{\|p_{l,i-1} - p_{m,i}\| + \|p_{m,i} - p_{n,i+1}\|} \right] \end{aligned} \tag{3.42}$$

where w_1 and w_2 are weights for controlling the contributions of direction and velocity changes respectively. Generally, p is defined as the center of the associated ellipse. The

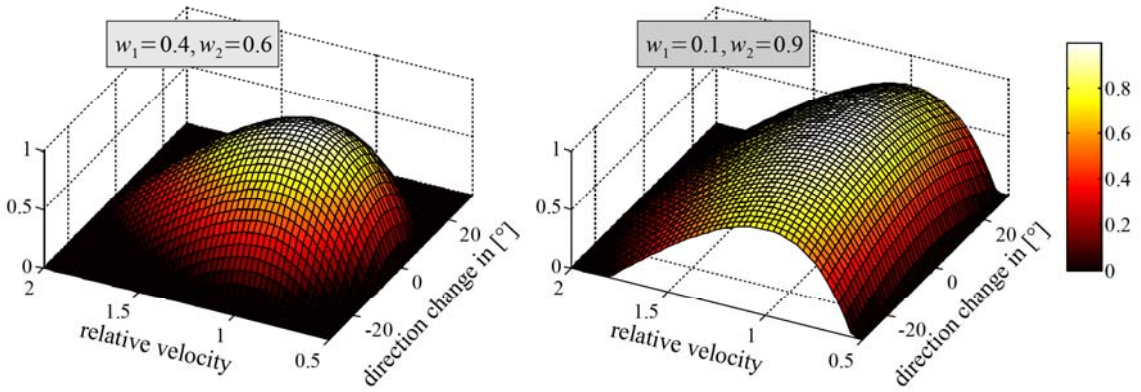


Figure 3.10: Path coherence function with two settings for *directional coherence* (w_1) and *velocity coherence* (w_2). During graph traversal, each feature transition is weighed with a coherence response value in addition to descriptor distance. In the case of more restrictive weights (left), motion jitter and thus the number of outliers within the tracks can be reduced significantly.

first term of the above expression is termed *directional coherence* and represents the dot product of the feature displacement vectors. The second term in equation 3.42 is referred to as *velocity coherence* and considers the geometric and arithmetic mean of the magnitude of both vectors. In [SJ87], the authors note that an additional acceleration term could be easily added to the given model. However, three consecutive frames would be necessary for its computation which is the reason why such an extension is generally discarded in both literature and practical applications. In the original publication, the model weights in equation 3.42 were chosen as $w_1 = 0.1$ and $w_2 = 0.9$ while Hwang proposed to use the setting $w_1 = 0.4$ and $w_2 = 0.6$ in [Hwa89]. According to the authors, the respective values were mainly determined experimentally according to the characteristics of the respective application. In figure 3.10, the effects of both settings on the model response are shown. It can be seen, that the latter setting penalizes direction changes more severely. In all experiments within this publication, it has proved more effective in avoiding motion jitter and thus in reducing the number of outliers.

In the following, the original function in equation 3.42 is extended to additionally respond to changes of feature scale in order to make use of additional information supplied by the affine-covariant detectors. Similar to the velocity term controlled by w_2, relative changes in scale s (which is the square root of the product of major and minor axis of the associated ellipse) are penalized proportionally to the associated region area. This leads to the *extended path coherence function*

$$m_e = m - w_3 \left[1 - 2 \frac{s_{l,i-1} s_{m,i}^2 s_{n,i+1}}{s_{l,i-1}^2 s_{n,i+1}^2 + s_{m,i}^4} \right]. \tag{3.43}$$

This particular expression has been reached by replacing the difference norms in the

second term of equation 3.42 through the corresponding scale ratios, *e.g.* $\|p_{l,i-1} - p_{m,i}\| \hat{=} \frac{s^2_{m,i}}{s^2_{l,i-1}}$. The feature scales have been additionally squared so that the expression is proportional to the feature support area (represented by the associated ellipses).

The major benefit of the new expression is a reduction of the number of graph nodes and thus of computational complexity. In section 3.4.2, this matter will be investigated in more detail.

Using equation 3.43, the *extended cost function* is defined as

$$c_e(l_{i-1}, m_i, n_{i+1}) = \left[1 - \frac{d_{l,m} + d_{m,n}}{2}\right] m_e. \tag{3.44}$$

If a node has several incoming and outgoing edges, the response of the coherence function m_e is computed for all possible combinations, favoring the transition with the smallest deviation in velocity, direction and scale.

The example in figure 3.9 shows a track graph existing in 3 consecutive frames: regions 1 and 2 are located in the first frame while regions 3, 4 and 5 exist in the second and third frame respectively. Based on the previously discussed groundtruth homographies from section 2.5.1, the correct trajectory can be determined as the sequence of features $(2, 3, 5)$. If the simple cost function from equation 3.39 was used, the wrong path $(1, 3, 5)$ would be chosen, due to $d_d(1, 3) < d_d(2, 3)$. If the cost function from equation 3.44 is alternatively used, the transition $(2, 3, 5)$ would be (correctly) preferred over $(1, 3, 5)$ due to its higher path coherence response.

In the following, a short discussion of the computational complexity of Dijkstra's optimal path search method is given. The most complex part of the algorithm is the process of finding the best node during graph traversal. The outer loop of the relaxation step takes up $O(\mathcal{N})$ time as each node is extracted once (where $\mathcal{N}$ is the number of nodes in a graph). Every time a node is extracted, its outgoing edges must be processed as well. During each iteration of the loop, each edge will be visited once in total, which gives the complexity $O(\mathcal{E} + \mathcal{N})$, where $\mathcal{E}$ is the number of edges in a graph. The cost for the extraction of the best node depends strongly on the data structure which is used for storage. In the most simple (linear) case, it would take $O(\mathcal{N})$ time to find the best node, giving a total cost of $O(\mathcal{N}^2)$ for all nodes. For sparse graphs, where $\mathcal{E} << \mathcal{N}^2$, the algorithm can be implemented more efficiently using an adjacency list as introduced in equation 3.38 in addition with a Fibonacci heap for data storage. In this case, the complexity can be reduced to $O(\mathcal{E} + \mathcal{N} \log \mathcal{N})$.

In the next section, the performance of all three methods is evaluated with regard to a series of appropriate measures, which are specially devised according to the requirements of a feature tracking application. These are presented in the following.

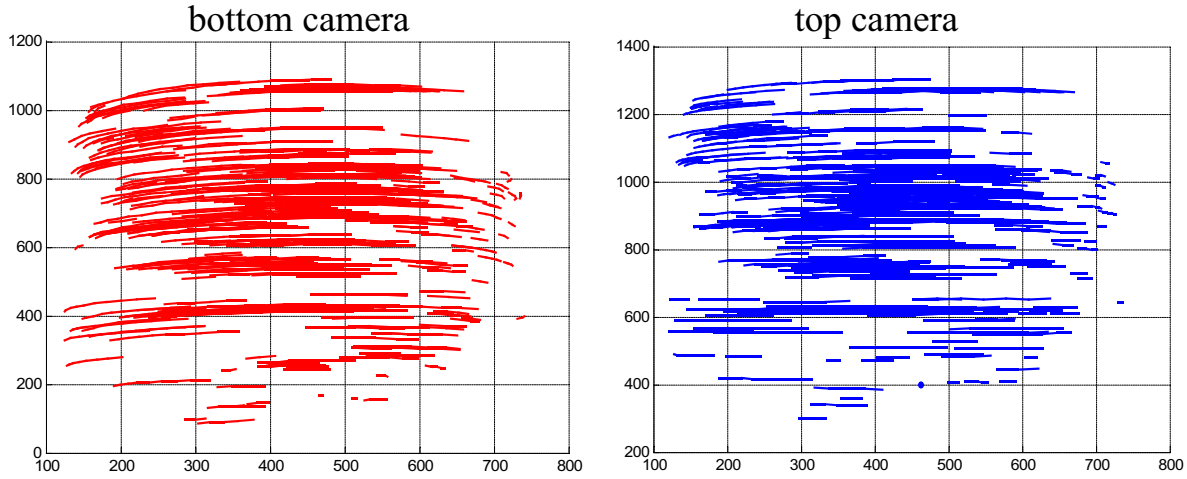


Figure 3.11: Two sets of trajectories for both the bottom- and top-camera of the measurement setup are shown. The results have been obtained by applying the previously described graph-based tracking method in conjunction with the MSER-detector to the *carton*-sequence from chapter 2.

3.4 Evaluation

3.4.1 Tracking Performance Measures

In order to assess the performance of a tracking algorithm, a set of adequate measures has to be defined. Ideally, a good tracking algorithm should

1. provide a large number of trajectories.

2. produce long trajectories, whose length represents the time of appearance of the associated feature within the observed scene.

3. exhibit a high localization accuracy in terms of the measures defined in the previous chapter.

From the above list, a number of tracking performance measures is derived, which are introduced in the following. Figure 3.11 shows two sets of trajectories for both the bottom- and top-camera of the measurement setup. The results have been obtained by applying the previously described graph-based tracking method in conjunction with the MSER-detector to the *carton*-sequence from chapter 2.

Inlier score In the previous chapter, a measure for the percentage of successfully matched regions has been used in order to assess the ability of an affine-covariant region

detector to detect the same feature repeatedly under varying viewpoints. In this section, a similar measure is employed for tracking applications. Given a homography $H_{i,i+1}$ which relates corresponding object surfaces in two frames, region correspondences may be identified based on the area overlap error d_o as introduced in equation 2.30. In section 2.5.3, two features (or regions) r_i^l and r_{i+1}^m were deemed 'corresponding', if the common area overlap after applying the transformation $r_i^{l'} = H_{i,i+1} r_i^l$ was below 50 %, *i.e.* $d_o(r_i^{l'}, r_{i+1}) \leq 0.5$. This threshold has been chosen in accordance with the relevant literature in the field and has been discussed previously in this work. In [MTS+05] the authors state that an overlap of $d_o \leq 0.5$ is still sufficient in most cases for a robust descriptor to match two regions succesfully.

Based on d_o, the *inlier score* is introduced in this section as a measure for the percentage of correspondences taken from the set of all trajectories with an area overlap error of $d_o \leq 0.5$, *i.e.* they are inliers to the respective groundtruth transformation $H_{i,i+1}$. The *inlier score* p_i is defined as

$$p_i = \sum_{i=1}^{F-1} \frac{\{(r_i \leftrightarrow r_{i+1})_l | d_o(r'_i, r_{i+1}) \leq 0.5,\ l \in \{1...L_i\}\}}{L_i}\ , \tag{3.45}$$

where the frame index i runs from the first to the penultimate frame $F-1$. The subscript l denotes the $l-th$ region correspondence from the set of *all* L_i correspondences for the current frame pair $\{I_i, I_{i+1}\}$.

Equivalently, the *outlier score* is defined as

$$p_o = 1 - p_i. \tag{3.46}$$

For subsequent applications, such outliers pose a significant problem, if for example the motion of an object is estimated based on a number of feature trajectories and an appropriately parameterized model. If the latter contained a high number of outlier correspondences, the estimation procedure would be more likely to fail depending on its robustness. Figure 3.12 illustrates the problem: There, two trajectories compete for the same feature candidate (gray circle). While the latter originally belongs to the lower trajectory (boxes), it is assigned to the upper one (triangles) instead. The resulting outlier correspondence contributes negatively to the inlier score p_i according to equation 3.45. Thus, the latter can also be seen as a measure for the suitability of a tracking method for the use in parameter estimation.

Trajectory lengths and number While the above-defined inlier score p_i only reflects if a region has been successfully tracked (with respect to its area overlap error after transformation), it does not take into account the number of features within the trajectory of which it is part. To this end, the *length of a trajectory* p_l is additionally evaluated. In order to avoid that trajectories containing outlier correspondences contribute positively

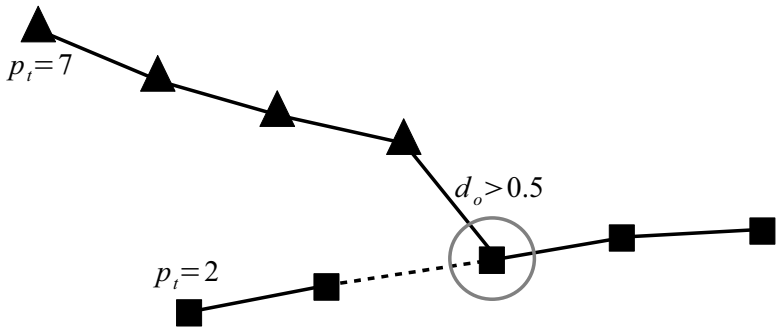


Figure 3.12: Two trajectories compete for the same feature (gray circle). While it originally belongs to the lower trajectory (boxes), it is falsely assigned to the upper one (triangles). The resulting *outlier correspondence* contributes negatively to the inlier score p_i.

to p_l, they are split up at positions where $d_o > 0.5$. Figure 3.12 illustrates a situation, where a wrong assignment leads to a misleadingly long trajectory ($p_l = 7$ instead of $p_l = 4$) due to a single outlier correspondence.

An additional indicator for tracking performance is given by the *number of trajectories* p_n. As with p_l, trajectories containing elements with an area overlap error of $d_o \geq 0.5$ are split up. If many such outliers exist however, the resulting number of trajectories naturally increases, as long tracks are split up into several shorter ones. Thus, this measure should always be seen in conjunction with p_l. Ideally, a tracking algorithm produces trajectories that are both long and numerous.

Trajectory accuracy. Finally, trajectories are evaluated with regard to the localization accuracy of the contained features. For every correspondence $r_i \leftrightarrow r_{i+1}$ within a trajectory, both area overlap error d_o and position error d_l according to equations 2.30 and 2.31 are computed. Similar to the detector evaluation in section 2.5, both measures will be analyzed statistically for every tracking method presented here.

3.4.2 Descriptor-based Region Trackers

In the following, the three tracking methods introduced in section 3.3 are evaluated with regard to the above-defined performance measures. For each of the five affine-covariant detectors from the previous chapter, the results of the respective algorithms are discussed and compared against each other. For region description, both SIFT and SPIN are used. To facilitate the flow of the argument, the following abbreviations are introduced: For the multi-region tracking algorithm with constant gating region size from section 3.3.1, the term *local tracking* is used, as decisions on a specific correspondence are always made

Table 3.1: Investigated range of the maximally permissible descriptor threshold $d_{d,max}$ for graph-based tracking.

	MSER	IBR	EBR	HARAFF	HESAFF
SIFT	$0.1 \leq d_{d,max} \leq 0.5$	$0.1 \leq d_{d,max} \leq 0.5$	$0.1 \leq d_{d,max} \leq 0.35$	$0.1 \leq d_{d,max} \leq 0.5$	$0.1 \leq d_{d,max} \leq 0.5$
SPIN	$0.1 \leq d_{d,max} \leq 0.35$	$0.1 \leq d_{d,max} \leq 0.35$	$0.1 \leq d_{d,max} \leq 0.2$	$0.1 \leq d_{d,max} \leq 0.35$	$0.1 \leq d_{d,max} \leq 0.25$

within the local context of the respective gating region. Further, the abbreviation *KF-based tracking* is chosen for the multi-region tracking algorithm using Kalman-filtering from section 3.3.2, while the graph-based multi-region tracker from section 3.3.3 is referred to as *graph-based tracking*.

A note on matching complexity. As noted in section 2.2.3, the descriptive power of SIFT is generally superior to SPIN, albeit at the expense of higher computational complexity. The maximum number of correspondences for a given image sequence and a specific detector is naturally bounded by the number of features within the gating region used in the matching procedure. As observed in figure 2.33, an increase in the maximally permissible descriptor distance $d_{d,max}$ during matching also increases the number of resulting region correspondences. For SPIN-descriptors, this upper bound of accepted correspondences (*positives*) is reached for a much lower setting of $d_{d,max}$ than with SIFT-descriptors. At the same time, the number of rejected correspondences (*negatives*) increases accordingly as seen in the same diagram. Based on these observations, a one-to-one comparison of both descriptors by simply selecting the same descriptor threshold is hardly possible.

In figure 2.34 of the same section, the dependency of the percentage of inliers was additionally tested against the number of correspondences. It has been shown, that for the same number of matched regions, the percentage of inlier correspondences is significantly lower for SPIN (with regard to the groundtruth homographies in section 2.2.4). Also, the amount of rejected candidates in form of the ratio $\frac{negatives}{positives}$ as seen in the lower diagram of the same figure, is more favorable in the case of SIFT-based correspondences. Thus, there exists no linear transformation of d_d that would relate these descriptors. As a consequence, each comparison between both is more of a qualitative than of a quantitative nature.

For local and KF-based tracking, the complexity of the matching problem is directly controlled by the size of the gating regions. In the case of weak descriptors, all candidates within them are selected as potential match candidates. Accordingly, the complexity of the combinatorial optimization procedure grows exponentially with $O(n^3)$. To this end, figure 2.28 in the previous chapter is a good indicator for the complexity of each affine-covariant detector. As can be seen, region density (and thus n) is especially high for the EBR-detector.

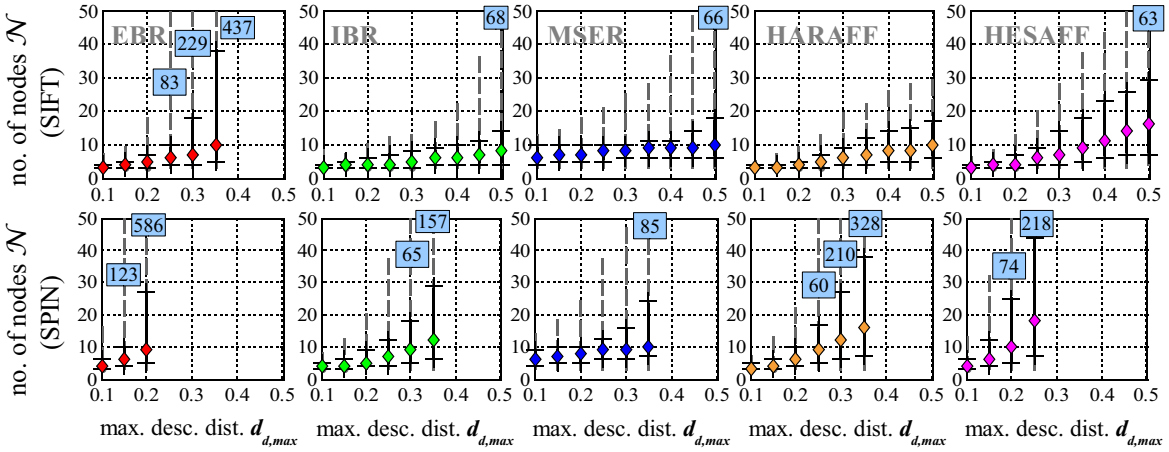
For the graph-based tracking approach, the additional load of traversing the con-

structed track-graphs has to be considered. In section 3.3.3, the complexity of traversal has been estimated as $O(\mathcal{E} + \mathcal{N} \log \mathcal{N})$. To this end, the dependency of the number of nodes $\mathcal{N}$ and edges $\mathcal{E}$ on $d_{d,max}$ has been evaluated for all five detectors and both descriptors in figure 3.13. While for IBR and HARAFF, $\mathcal{N}$ and $\mathcal{E}$ are comparably low even for $d_{d,max} = 0.5$, EBR suffers from increasing complexity: For $d_{d,max} = 0.35$, almost 25% of all EBR-graphs contain more than 40 nodes while in the upper 5-percentile, their number exceeds 400 with up to 5000 edges. For higher settings of $d_{d,max}$, path extraction in acceptable time is no longer possible. Notably, the complexity of HESAFF-based graphs is much lower than for EBR, although according to figure 2.21, the number of region correspondences is significantly higher. Based on these results, the maximally permissible descriptor thresholds have been limited according to table 3.1.

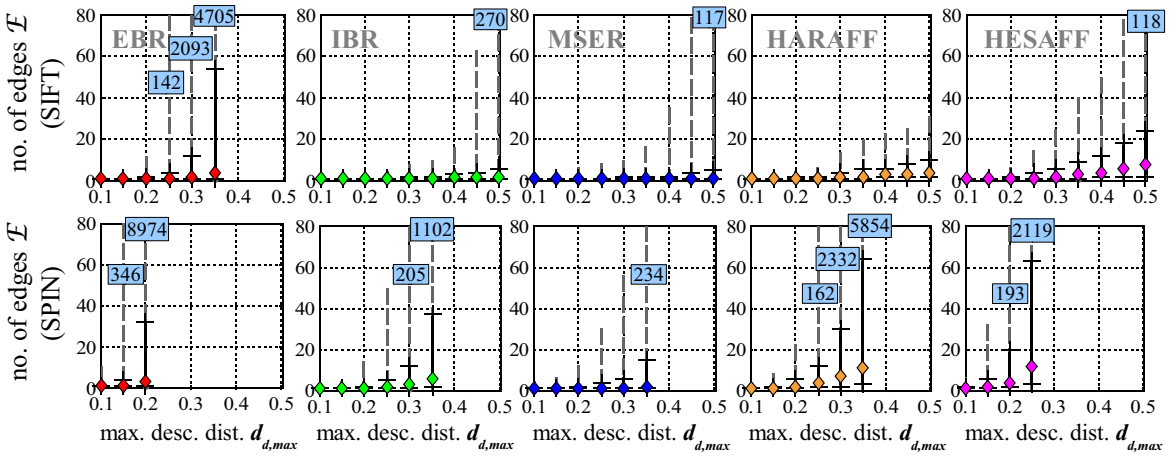
Percentage of Inliers In figure 3.14.1, the inlier percentage p_i is shown as a function of the maximally permissible descriptor threshold $d_{d,max}$ for *local tracking*. It can be seen, that for SIFT-based trajectories p_i stays well above 95 % over the entire range of investigated settings of $d_{d,max}$. While the MSER-detector maintains a relatively constant p_i, the EBR-detector exhibits the most significant decrease among all other methods, especially on the interval $0.4 \leq d_{d,max} \leq 0.5$. Between IBR, HARAFF and HESAFF however, differences are marginal. If SPIN is used as region descriptor instead, p_i rapidly degrades with increasing $d_{d,max}$ until a rate of $p_i = 85$ % at $d_{d,max} = 0.5$. For MSER-regions, p_i stays above 97 % for all settings of $d_{d,max}$. By considering the number of neighboring regions for this detector from 2.28 in the previous chapter, this behaviour finds an explanation: Among all detectors, MSER provides the smallest number of regions, which are most distinctive with regard to the mean shape overlap $\overline{d_o}$. Thus, the probability for outlier correspondences is accordingly low as well. Notably, the inlier percentage for HESAFF-regions with $d_{d,max} \geq 0.3$ is far better than for IBR- or HARAFF-regions, although the number of neighboring regions is higher (with $\overline{d_o}$ almost equivalent). For the EBR-detector, a setting of $d_{d,max} > 0.35$ lead to a strong increase in the number of potential correspondences within the gating region. The combinatorial optimization step from section 2.5.2 was thus too complex with regard to practical usability.

In figure 3.14.2, the inlier percentage p_i as a function of $d_{d,max}$ is shown for *KF-based tracking*. For the SIFT descriptor, the performance essentially compares to local tracking. Only for MSER-regions, p_i is slightly lower if Kalman-filtering is used. With regard to SPIN however, KF-based trajectories show a better performance, especially if HESAFF-regions are used. While for the first method p_i falls below 90 % for $d_{d,max} \geq 0.4$, KF-based trajectories stay above $p_i \geq 92$ % for all investigated settings of $d_{d,max}$.

In figure 3.14.3, the inlier percentage p_i as a function of $d_{d,max}$ is shown for *graph-based tracking*. It can be seen, that p_i stays well above 97 % for all five detectors with *both* SIFT- and SPIN-descriptors. Especially for the latter, the improvement of p_i with regard to the other tracking methods is significant. In practice, this means a significant



3.13.1: Number of graph nodes.



3.13.2: Number of graph edges.

Figure 3.13: The diagrams show the median number of nodes $\mathcal{N}$ (*top*) and edges $\mathcal{E}$ (*bottom*) within a track graph (diamond markers) for both SIFT and SPIN, the 25%- and 75%-percentiles (solid lines) and the 5%- and 95%-percentiles (dashed lines). The complexity of track-graph traversal is directly dependent on both $\mathcal{E}$ and $\mathcal{N}$ with complexity $O(\mathcal{E} + \mathcal{N} \log \mathcal{N})$. Additionally, table 3.1 shows the limits on the maximally permissible descriptor distance $d_{d,max}$ for each detector/descriptor combination.

reduction of the probability to encounter outliers (*i.e.* correspondences with $d_o > 0.5$) in the resulting trajectories. This is especially the case for the SPIN-descriptor, which performs almost identical to the SIFT-descriptor at the benefit of reduced computation time in both generation and distance computation.

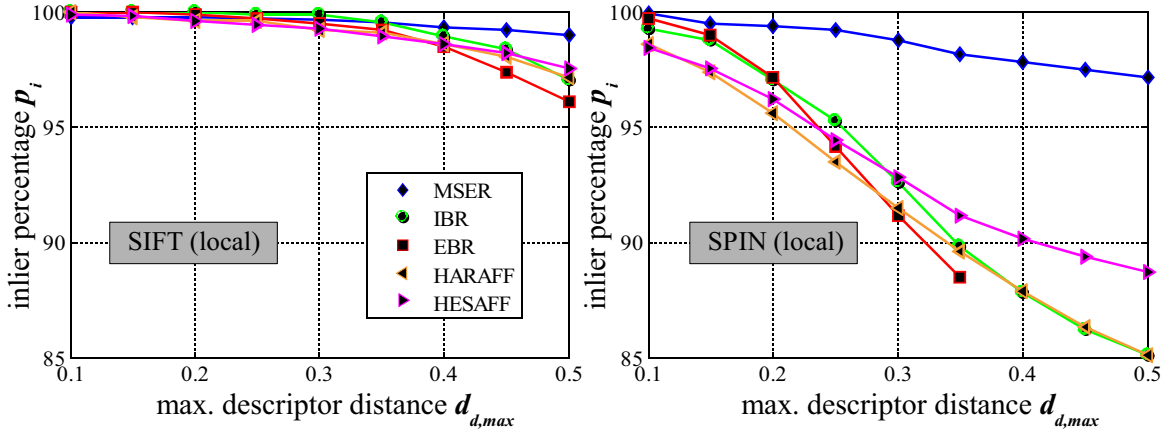
Number of Trajectories. In figure 3.15.1, the number of trajectories p_n is shown for *local tracking*. As before, all five detectors have been evaluated, in conjunction with either SIFT- or SPIN-descriptors. For both methods, the relative ordering of the detectors stays essentially the same, with HESAFF as the undoubtedly superior method. The second-highest number of trajectories is reached by EBR, albeit with significant distance to HESAFF. The MSER-detector shows the worst performance with only $\frac{1}{6}$ of the trajectories found by the best detector. Also, there is almost no increase in p_n to be observed over the entire range of $d_{d,max}$. For SPIN, p_n reaches a level of saturation for $d_{d,max} \approx 0.4$ while with SIFT, p_n still shows an increase, even at $d_{d,max} = 0.5$. With MSER, the number of trajectories saturates for $d_{d,max} \approx 0.15 - 0.20$ already. This might be owed to the low number and comparably low density of regions within the investigated image sequences, as shown in table 2.10 and figure 2.28.

Except for HARAFF, the relative ordering of the detectors is identical to figure 2.20 in the previous chapter, where the relative number of correspondences was evaluated against the maximally permissible region overlap $d_{o,max}$, although the relations between the detectors are of different magnitudes. Especially the relative difference between HESAFF and EBR is much more pronounced in figure 2.20.

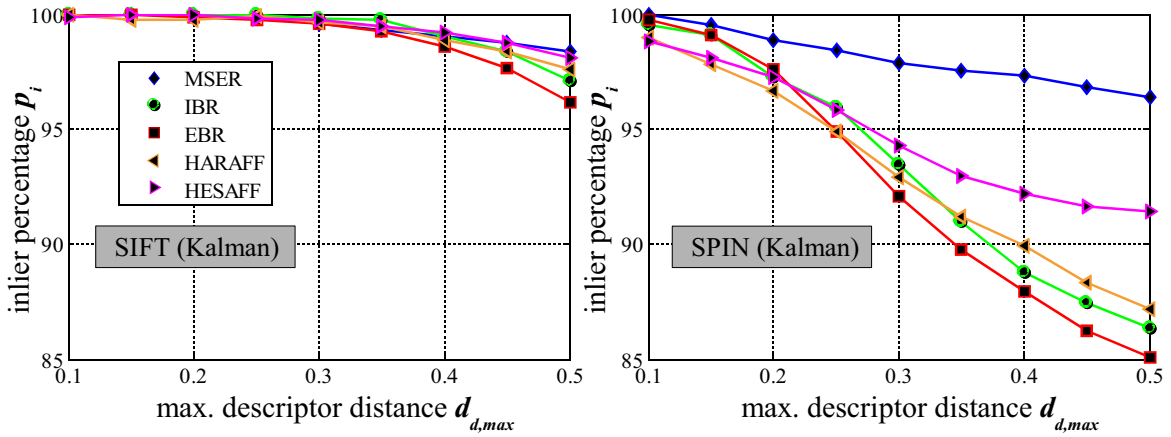
In figure 3.15.2, p_n as a function of $d_{d,max}$ is shown for *KF-based tracking*. For EBR and HESAFF, the number of trajectories is slightly higher than for local tracking while for MSER, IBR and HARAFF results are essentially identical. Also, the relation between detectors is largely identical to local tracking.

Finally, *graph-based tracking* has been analyzed with regard to p_n in figure 3.15.3. Notably, the number of trajectories is significantly lower than with both local or KF-based tracking. While the relative ordering of the detectors stays the same, p_n is lower by approximately 20 $\%$. For smaller settings of $d_{d,max}$ however, differences between the three methods are less pronounced (≈ 5 $\%$ with SIFT at $d_{d,max} = 0.2$).

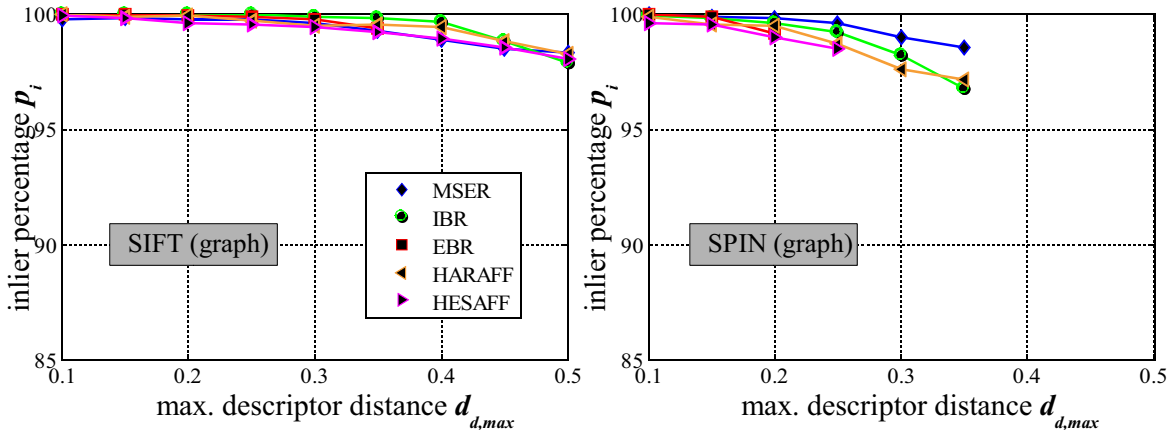
Distribution of Trajectory Length In figure 3.16.1, the dependency of trajectory lengths d_l on descriptor distance d_d is shown, again for both SIFT- (top) and SPIN-descriptors (bottom). The diagrams show the median p_l (diamond markers), the 25- and 75-percentiles (solid lines) and the 5- and 95-percentiles (dashed lines), estimated from all sequences. In all diagrams, a clear dependency of p_l on descriptor distance is observable. Generally, trajectories are longer if the descriptors of the associated features are similar to each other. Especially for MSER-regions, this relation is most obvious. With MSER, more than 50 $\%$ of all tracks contain at least 6 features for $d_d \leq 0.2$ (from 10



3.14.1: Multi-region tracking with constant gating region

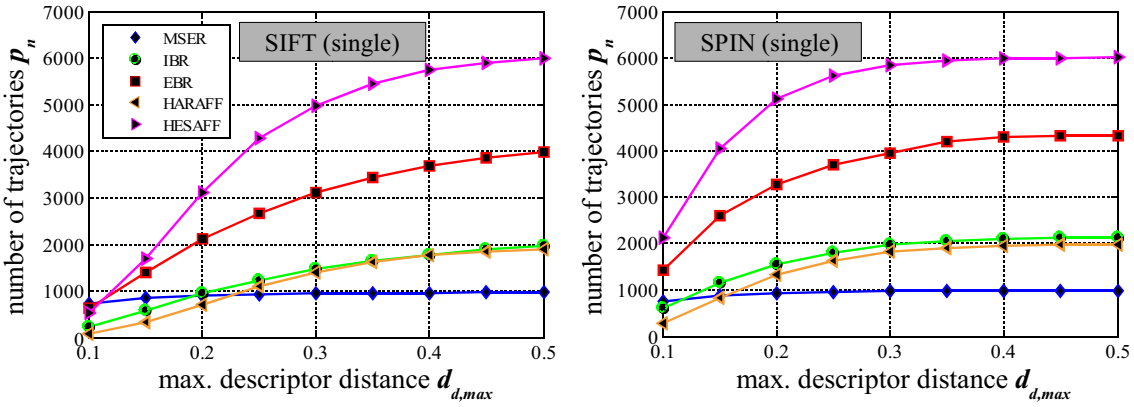


3.14.2: Multi-region tracking using Kalman-filtering

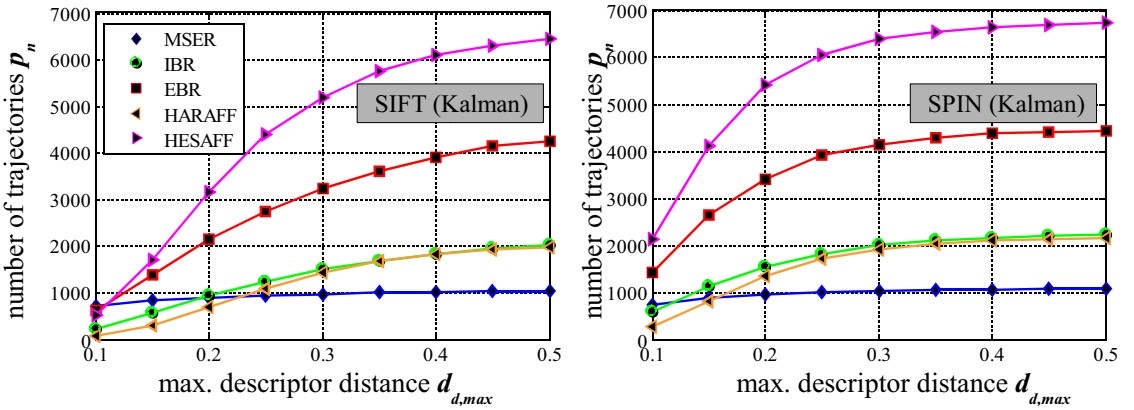


3.14.3: Graph-based multi-region tracking

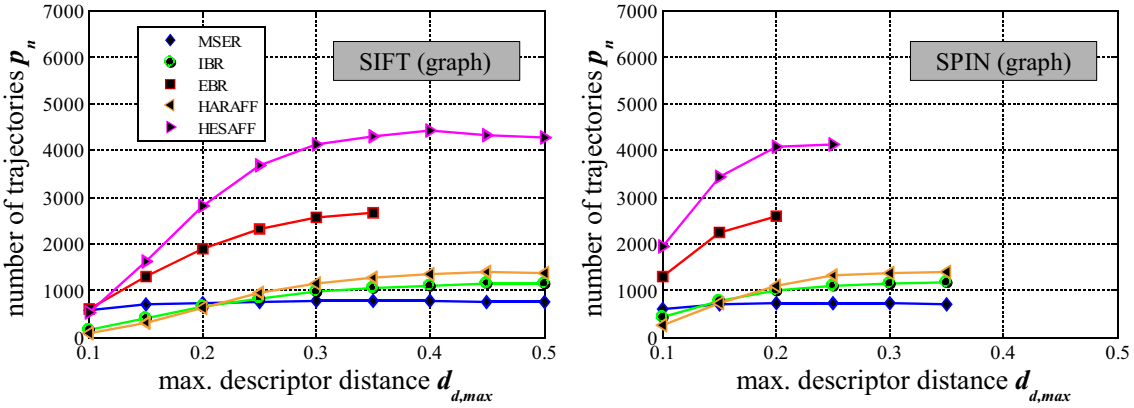
Figure 3.14: Inlier percentage p_i as a function of the maximally permissible descriptor threshold $d_{d,max}$ for the five affine-covariant region detectors.



3.15.1: Multi-region tracking with constant gating region



3.15.2: Multi-region tracking using Kalman-filtering



3.15.3: Graph-based multi-region tracking

Figure 3.15: Number of trajectories p_n as a function of the maximally permissible descriptor threshold $d_{d,max}$ for five affine-covariant region detectors. In order to avoid that trajectories containing outlier correspondences contribute positively to p_l, they have been split up at positions where $d_o > 0.5$.

frames of each sequence). Clearly, MSER-regions are superior to all other detectors with regard to p_l, especially for small descriptor distance. For the remaining detectors, the same dependency exists as well, although not as pronounced. The second-best detector is IBR, although differences toward the remaining three methods are comparably small. With SIFT, differences between EBR and HESAFF are marginal while with SPIN, the performance of HESAFF is slightly better.

Notably, differences between local tracking and KF-based tracking are virtually non-existent. For all detectors and for both SIFT and SPIN, the resulting distributions and dependencies are largely identical. Thus, the diagrams in figure 3.16.1 are used to represent the results for both methods.

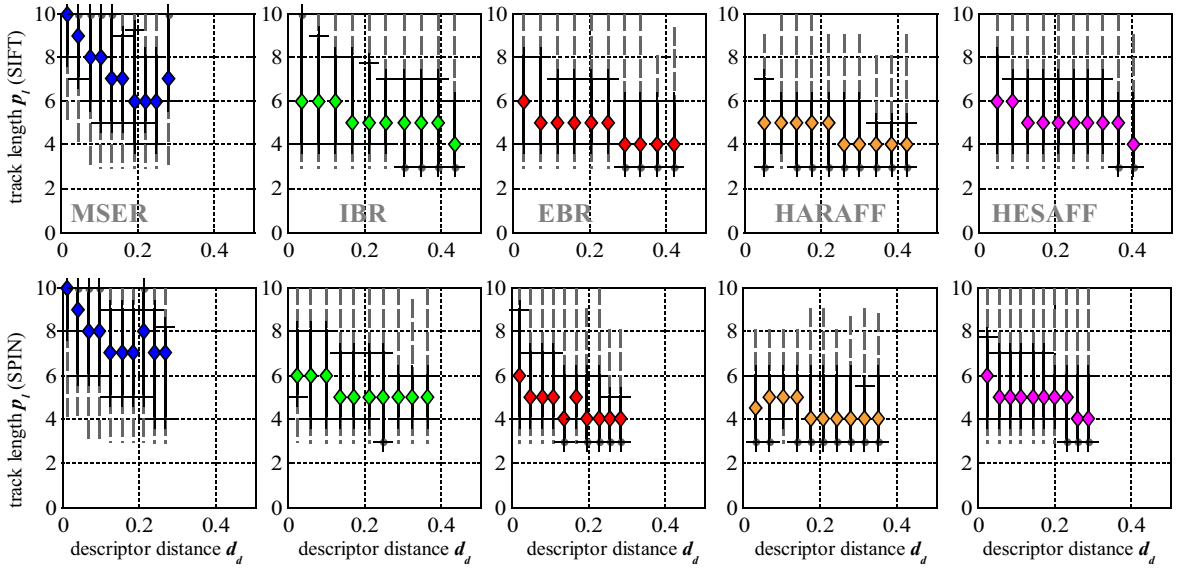
In figure 3.16.2, the results for *graph-based tracking* are shown. Compared to local and KF-based tracking, the decrease of p_l with increasing descriptor distance is much less obvious, although trajectory lengths are generally higher with every detector. Further, it can be said that HESAFF-based trajectories are superior to EBR for both SIFT and SPIN. As before, MSER again shows the best performance among all detectors, while there are almost no differences between EBR and IBR to be observed.

Position error In figure 3.17.1, the dependency of the position error d_l on descriptor distance d_d is shown for *local tracking* with either SIFT- (top) or SPIN-descriptors (bottom). The diagrams show the median error (diamond markers), the 25- and 75-percentiles (solid lines) and the 5- and 95-percentiles (dashed lines), estimated from all sequences. The relative ordering of the detectors corresponds to the right side of figure 2.36, where the distribution of d_l over the n-percentile has been evaluated. Repeatedly, MSER-regions show the smallest position error with the 75-percentile of d_l below $5\ pel$ over the entire investigated range of d_d. In all diagrams, an approximately linear dependency of d_l on descriptor distance exists. Except in the case of MSER, the position error is generally lower for SIFT-based trajectories than with SPIN.

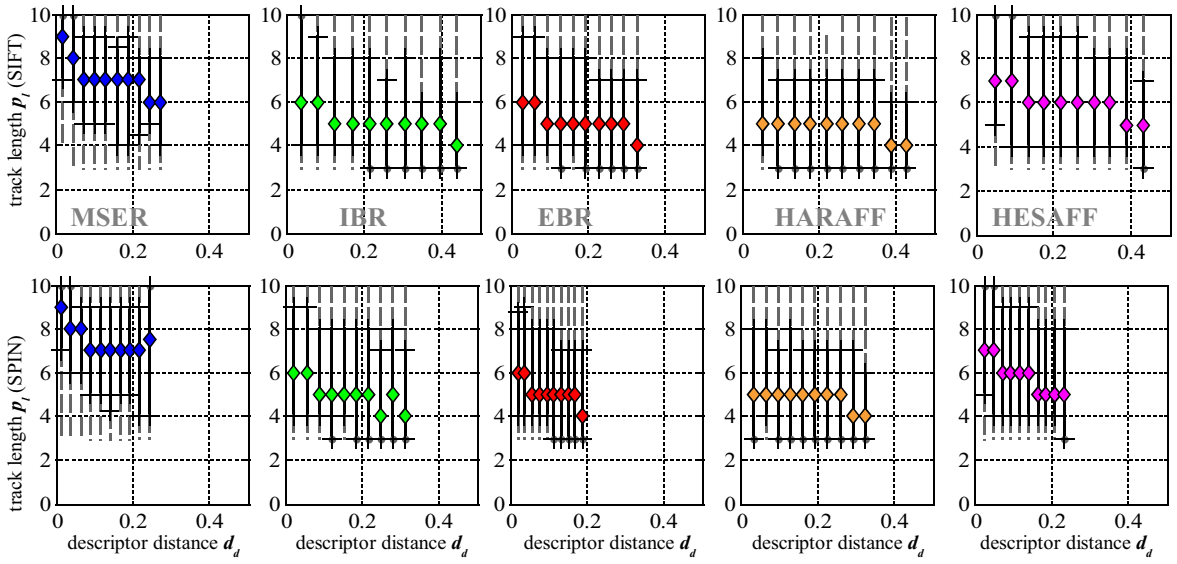
In figure 3.17.2, the results for *KF-based tracking* are shown. Compared to local tracking, the distribution of d_l is almost identical for all five detectors and both descriptors.

Lastly, figure 3.17.3 shows the results for *graph-based tracking*. While the relative ordering of the detectors essentially stays the same, the overall position error is significantly lower than with either local or KF-based tracking with regard to both the medians and the outer percentiles. For MSER-based trajectories however, the improvement is marginal.

Area overlap error Further, figure 3.18.1 shows the dependency of the area overlap error d_o on descriptor distance d_d for *local tracking*. As before, the results for both SIFT- (top) and SPIN-descriptors (bottom) are given. The diagrams show the median error (diamond markers), the 25- and 75-percentiles (solid lines) and the 5- and 95-

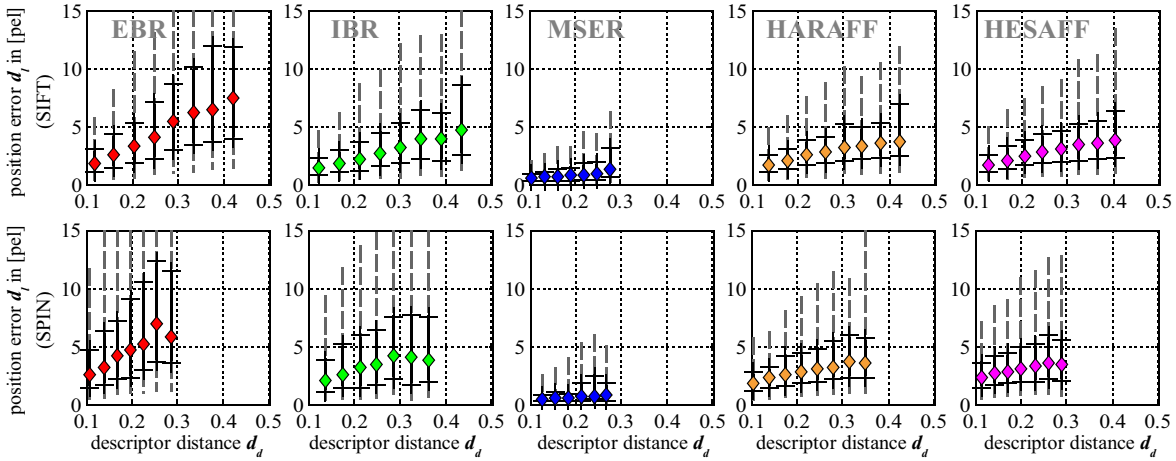


3.16.1: Multi-region tracking with constant gating radius. Notably, differences between local tracking and KF-based tracking are virtually non-existent. For all detectors and for both SIFT and SPIN, the resulting distributions and dependencies are more or less identical. Thus, the diagrams in this figure are used to represent the results of both methods.

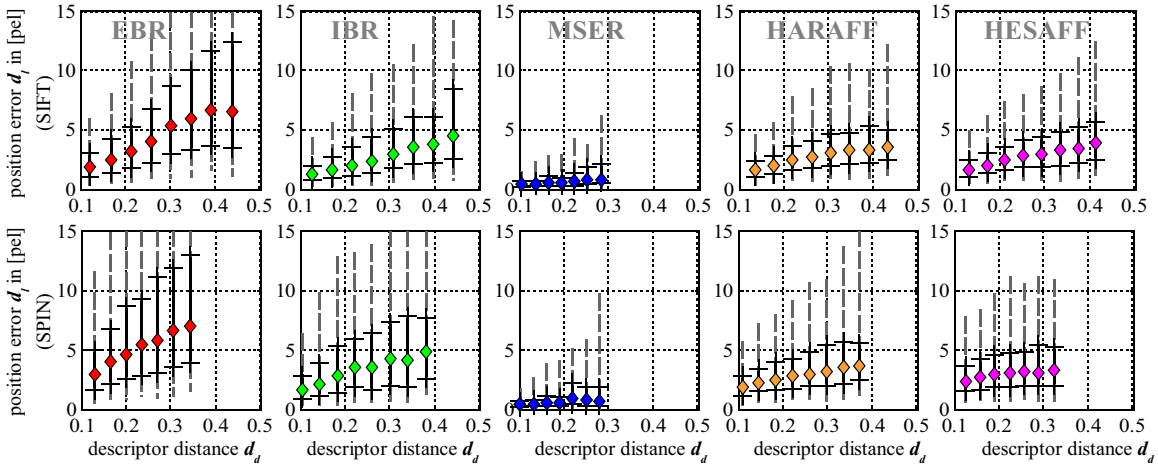


3.16.2: Graph-based multi-region tracking

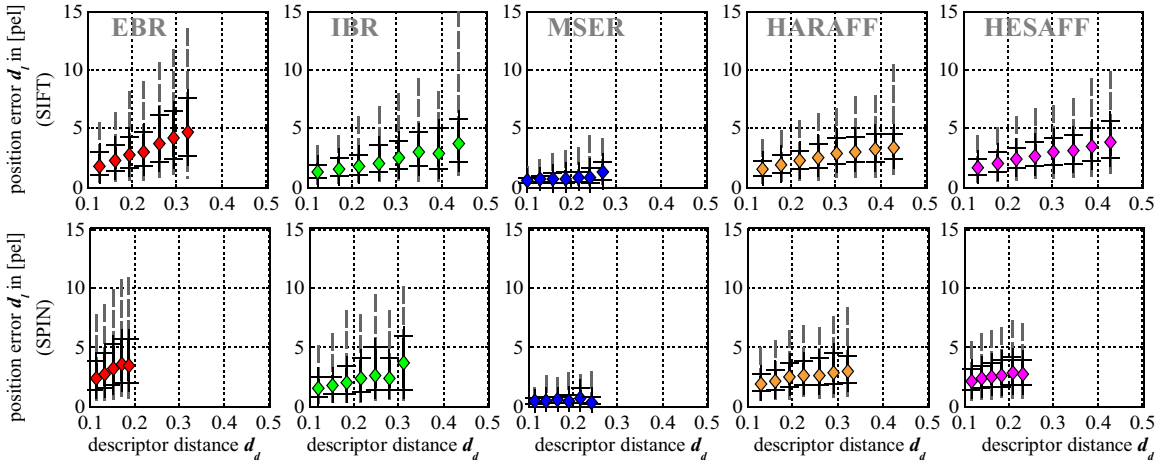
Figure 3.16: Trajectory length p_l as a function of descriptor distance d_d for the five affine-covariant region detectors is shown, based on either SIFT- (top) or SPIN-descriptors (bottom). The diagrams show the median p_l (diamond markers), the 25- and 75-percentiles (solid lines) and the 5- and 95-percentiles (dashed lines), estimated from all sequences. In order to avoid that trajectories containing outlier correspondences contribute positively to p_l, they have been split up at positions where $d_o > 0.5$. The minimally required length for the inclusion of a trajectory in the evaluation has been set to $p_l = 3$.



3.17.1: Multi-region tracking with constant gating region



3.17.2: Multi-region tracking using Kalman-filtering



3.17.3: Graph-based multi-region tracking

Figure 3.17: Position error d_l as a function of descriptor distance d_d for five affine-covariant region detectors is shown, based on either SIFT-descriptors (top) or SPIN (bottom).

percentiles (dashed lines), estimated from all sequences. As with d_l, there is a clear linear dependency of d_o on descriptor distance to be observed. The higher d_d, the larger are the corresponding overlap errors. In accordance with the distribution of d_o on the left side of figure 2.36 in the previous chapter, both HARAFF and HESAFF show the highest errors. Between EBR and IBR, there exist only few differences while MSER-regions show the smallest d_o. Notably, the overlap error is slightly smaller, if SPIN-descriptors are used during tracking.

Secondly, figure 3.18.2 shows the result for *KF-based tracking*. As with d_l, the results are comparable to local tracking although in this case, there are no significant differences between SIFT and SPIN to be observed.

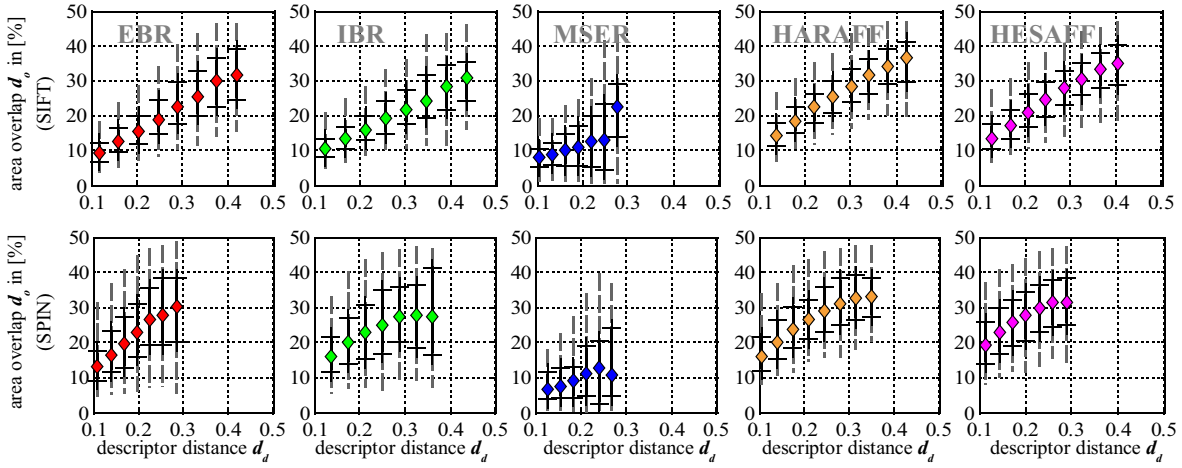
Finally, figure 3.18.3 shows the results for graph-based tracking. While MSER again shows the lowest error, the second-best performance is notably achieved by EBR-based trajectories. Also, contrary to d_l, the error is essentially the same as with both other tracking methods (except for IBR). Thus, graph-based tracking seems to improve the position accuracy alone, while the area overlap error stays approximately the same.

Robustness against perspective distortion In this section, the general robustness of descriptor-based region matching against perspective transformations of the object plane is evaluated with regard to the number of matches and the percentage of inliers. Contrary to the previous sections, the subsequent processing of the resulting correspondences with either one of the presented tracking methods is not considered here. Given the automated turn-table described in section 2.3, the rotation angle ϕ between two frames is increased by $5°$ with each step, starting from a frontoparallel view ($0° \leq \phi =\leq 30°$) of the investigated objects. For all rotation experiments, a maximally permissible descriptor distance of $d_{d,max} = 0.5$ has been selected. Figure 3.19 additionally shows the corresponding frames of the *carton*-sequence (bottom camera).

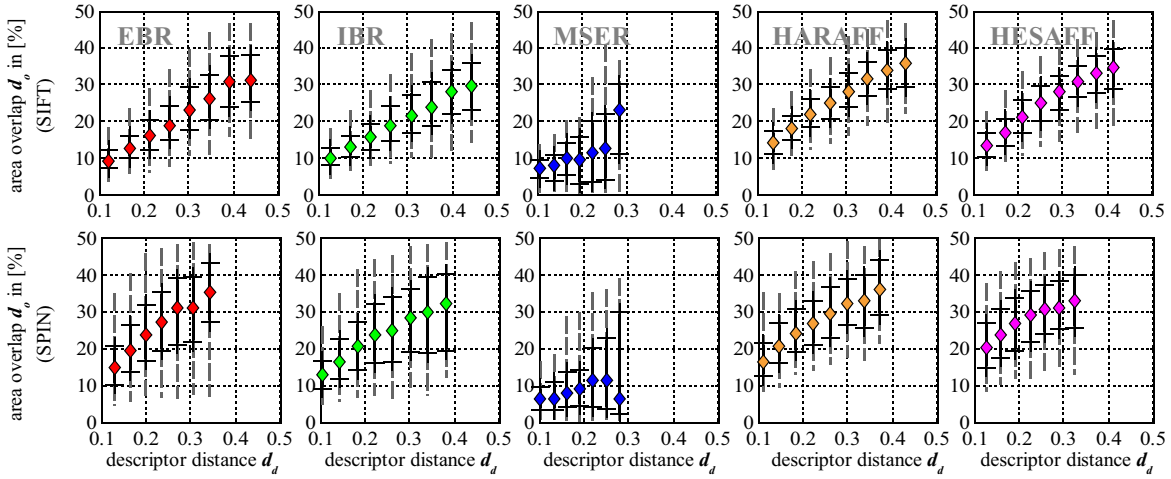
Figure 3.20.1 shows the percentage of inliers p_i (left) and the number of matches (right) as a function of the rotation angle for the SIFT-descriptor. Considering p_i at $\phi = 5°$, the results are approximately identical to figure 3.14.1, where the dependency of p_i on descriptor similarity d_d has been evaluated for local tracking. Until a rotation of $\phi = 20°$, the number of inliers for SIFT-based correspondences remains essentially constant, with only minor differences between the detectors. For a rotation above $\phi \geq 20°$, p_i starts to decreases linearly. In this regard, the MSER-detector performs worst with $p_i \approx 50$ % at $\phi = 30°$ while the IBR-detector exhibits the highest robustness against rotation with $p_i \approx 85$ % at $\phi = 30°$.

Considering the number of matches (right), the HESAFF-detector is clearly superior, followed by the EBR-detector in considerable distance. The relative ordering of the detectors and the ratios between them is largely consistent with figure 3.15, where the number of trajectories p_n is evaluated as a function of $d_{d,max}$. As with p_i, the slope decreases for $\phi \geq 20°$, although not with the same steepness.

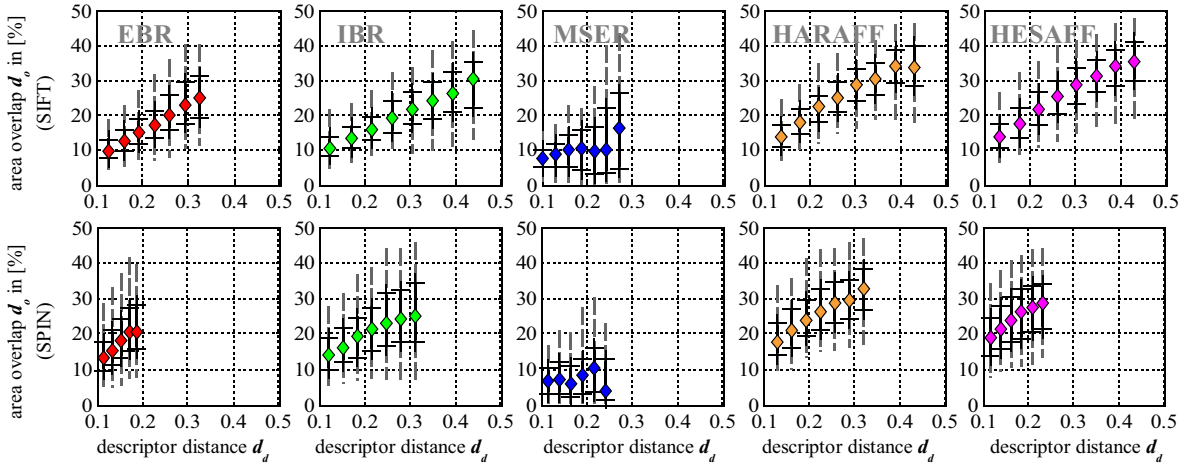
Figure 3.20.2 shows the corresponding results using SPIN. Except for MSER, p_i is



3.18.1: Multi-region tracking with constant gating region



3.18.2: Multi-region tracking using Kalman-filtering



3.18.3: Graph-based multi-region tracking

Figure 3.18: Area overlap error d_o as a function of descriptor distance d_d for five affine-covariant region detectors is shown, based on either SIFT- (top) or SPIN-descriptors (bottom).

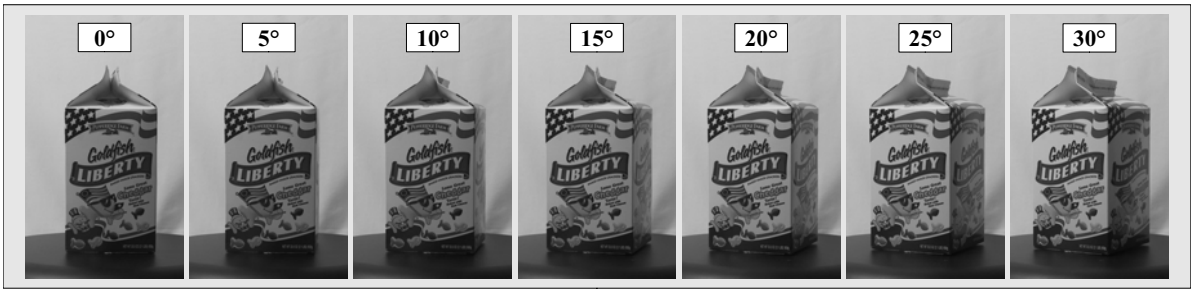
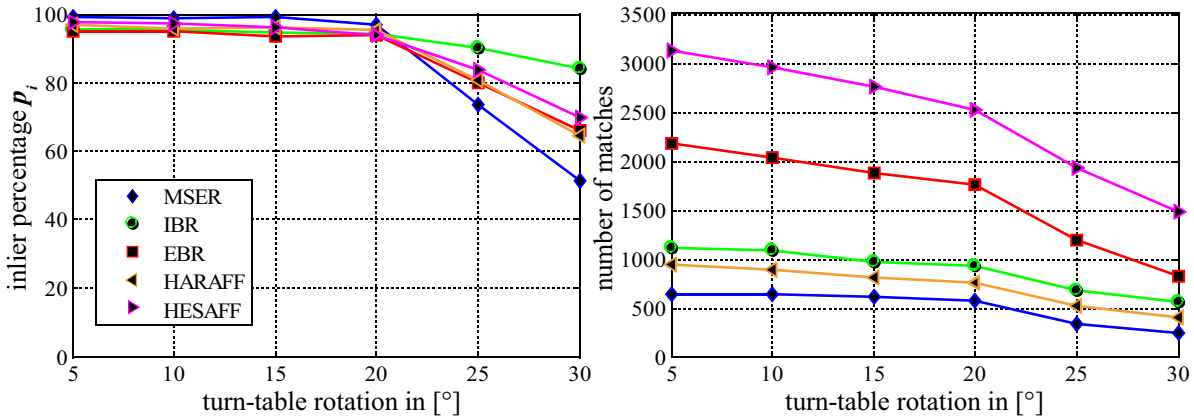
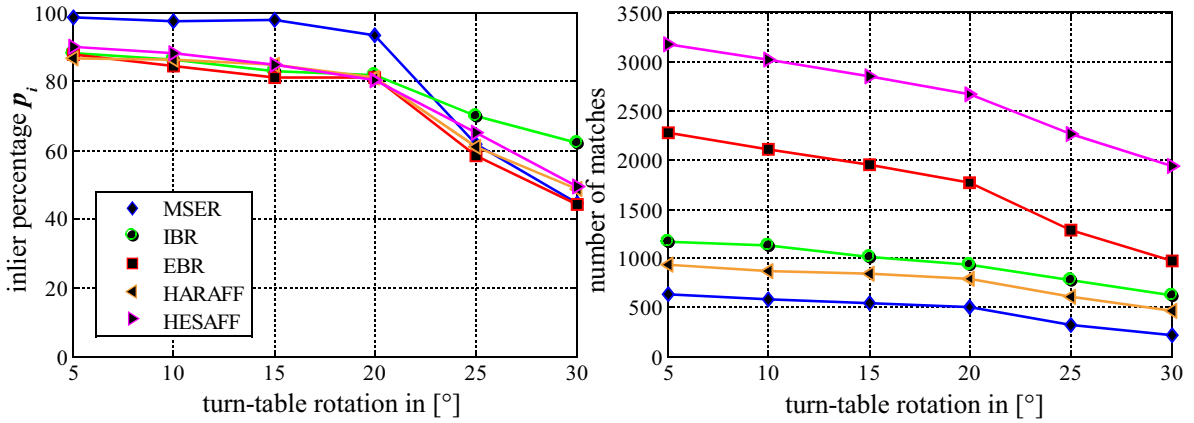


Figure 3.19: 7 frames of the *carton*-sequence (bottom camera) with a turn-table rotation of 5° between neighboring frames.

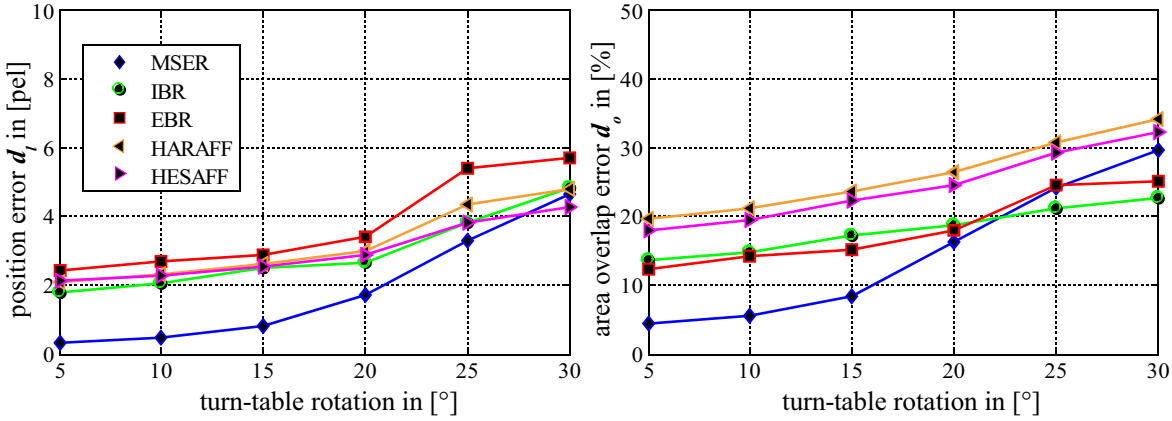


3.20.1: SIFT-based correspondences

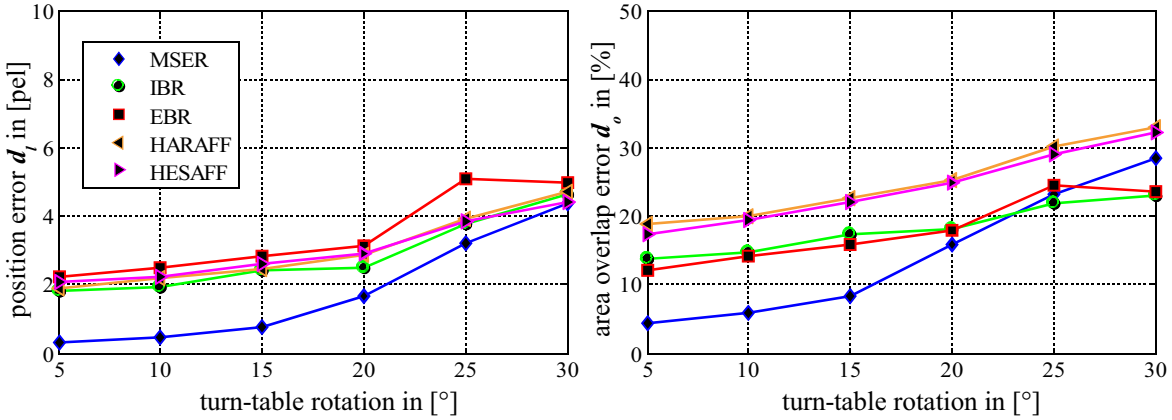


3.20.2: SPIN-based correspondences

Figure 3.20: The above diagrams show the percentage of inliers p_i (left) and the number of matches (right) as a function of the rotation angle ϕ for the five affine-covariant detectors.



3.21.1: SIFT-based correspondences



3.21.2: SPIN-based correspondences

Figure 3.21: The above diagrams show the median position error d_l (left) and the median area overlap error d_o (right) as a function of the rotation angle for the five affine-covariant detectors.

generally lower by $\approx 10\,\% - 20\,\%$. As before, there is a moderate decrease of p_i until a rotation angle of $\phi = 20°$. With further increasing ϕ, the steepness of the slope grows as well. With regard to the number of matches, HESAFF and EBR perform slightly better than with SIFT-based matching while for MSER, the number of matches is slightly lower with SPIN. Figure 3.21 shows the median position error d_l (left) and the median overlap error d_o (right) as a function of ϕ for both SIFT-descriptors and SPIN. With all diagrams, there is a steadily growing error with increasing ϕ to be observed. Between both descriptors, there exist no significant differences.

Concluding, affine-covariant regions provide a certain robustness against perspective transformations of the image plane. However, for a rotation above $\phi \geq 20°$, the matching performance and the localization accuracy rapidly decrease.

3.4.3 Kanade-Lucas-Tomasi Tracker: A Reference

A well-known and widely used feature tracking method for computer-vision applications is the *Kanade-Lucas-Tomasi*- or KLT-tracker as introduced in section 3.2.4. The basic principle of this method is the alignment of a template window within a reference frame to a target window in a second frame such that the similarity of the image signals within both is maximized, according to equation 3.22. The algorithm consists of two stages: In the first stage, a pure translational model with 2 degrees of freedom is used to find corresponding features within neighboring frames I_i and I_{i+1} under the assumption, that appearance changes are small. In the second stage, a similarity measure is computed between the first frame of feature appearance and the current frame under an affine model with 6 degrees of freedom. The underlying assumption is that the longer a feature is tracked, the more significant will be the appearance change toward the first frame. Here, a pure translational model would be an insufficient representation. The similarity coefficient resulting from the estimated affine transformation parameters is used as a quality measure: if it falls below a predefined threshold, the corresponding trajectory is terminated, otherwise tracking is continued to the next frame I_{i+1}.

A significant disadvantage of the KLT-tracker is its inability to select the feature scale automatically. Detection is performed using the second-order moment matrix as defined in equation 2.1, which involves the convolution of the image signal with a Gaussian filter kernel of predefined size and circular shape. The desired target feature size has to be provided manually by a human expert (*e.g.* according to the predominant feature size in the image sequence). Also, there are no region descriptors involved for the assignment of corresponding features. Thus, a one-to-one reproduction of the evaluation within the previous section is hardly possible. Alternatively, experiments have been performed for several settings of the convolution window size w_c. In the following, the performance measures introduced in section 3.4.1 are used to compare the KLT-tracker to the affine-covariant region trackers evaluated within the previous section.

Percentage of Inliers In figure 3.22, the percentage of inliers p_i and outliers p_o has been evaluated as a function of the convolution window size w_c (right). For small settings of w_c, the percentage of outliers within the feature trajectories attains as much as $p_o = 60\ \%$ (for $w_c = 10\ pel$). With increasing window size however, p_i rapidly increases, until a rate of almost $100\ \%$ for $w_c \geq 40\ pel$. Obviously, a large window size is a sound precondition for the stable estimation of the affine similarity model. The smaller w_c, the higher the probability for false decisions, which is especially evident in the case of multiple similar features within a close neighborhood of each other.

Compared to the results of the affine-covariant detectors in figure 3.14, the performance of the KLT-tracker is clearly superior with regard to p_i (given a sufficient window size), although most detectors achieve a performance close to $p_i = 100\ \%$ as well, given a sufficiently low setting of the maximally permissible descriptor distance $d_{d,max}$.

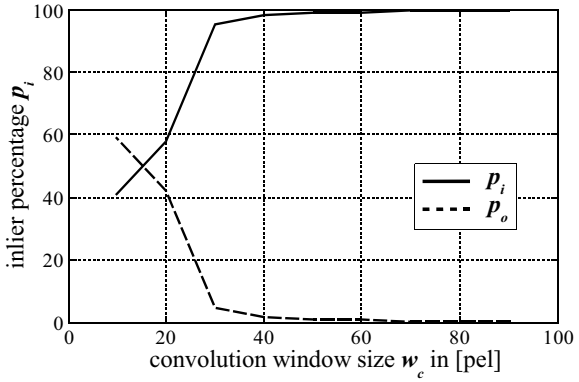


Figure 3.22: Percentage of inlier correspondences p_i with $d_o \geq 0.5$ as a function of the convolution window size w_c. Additionally, the percentage of outliers is given, which is defined as $p_o = 1 - p_i$.

Number and length of trajectories. In figure 3.23, the distribution of trajectory length p_l (left) and numbers p_n (right) are evaluated as a function of w_c. Compared to figure 3.16, p_l is significantly lower for the KLT-tracker than for affine-covariant regions for every investigated choice of w_c. For almost all settings of d_d (with both SIFT- and SPIN-descriptors) and each detector, p_l is superior for the latter. Only for the smallest investigated window size ($w_c = 10\ pel$), more than $25\ \%$ of all trajectories exhibit a length of $p_l \geq 5$. Notably, the trajectory lengths decrease with increasing window size. Like in the previous section, tracks have been split up at positions, where an element exceeded $d_o > 0.5$ in order to mitigate the influence of outliers on the evaluation results.

On the right side of figure 3.23, the number of trajectories p_n is additionally given, also as a function of w_c. Notably, p_n shows an increase until a maximum of approximately 6000 trajectories for a convolution window size of $w_c = 40\ pel$. With further increasing window size, p_n decreases and eventually falls below $p_n \leq 2000$. Compared to figure 3.15, the KLT-tracker can not compete with a HESAFF- or EBR-based tracker but comes at level or even exceeds HARAFF, IBR and MSER. For graph-based tracking as seen in figure 3.15.3, the difference between KLT- and region-based tracking is even smaller.

Position and area overlap error In figure 3.24, the dependency of the median position error d_l (left) and the median area overlap error d_o (right) have been evaluated as a function of w_c. As with p_l and p_n, there exists a dependency of both d_l and d_o on the convolution window size. For low settings of w_c, the position error is generally low as well. For larger settings of w_c however, d_l remains essentially constant with a median error of $d_{l,50} \approx 9\ pel$. In comparison to the results in figure 3.17, the KLT-tracker is thus clearly inferior to region-based tracking with regard to d_l. Considering d_o, one has to keep in mind that feature detection is based on circular convolution kernels.

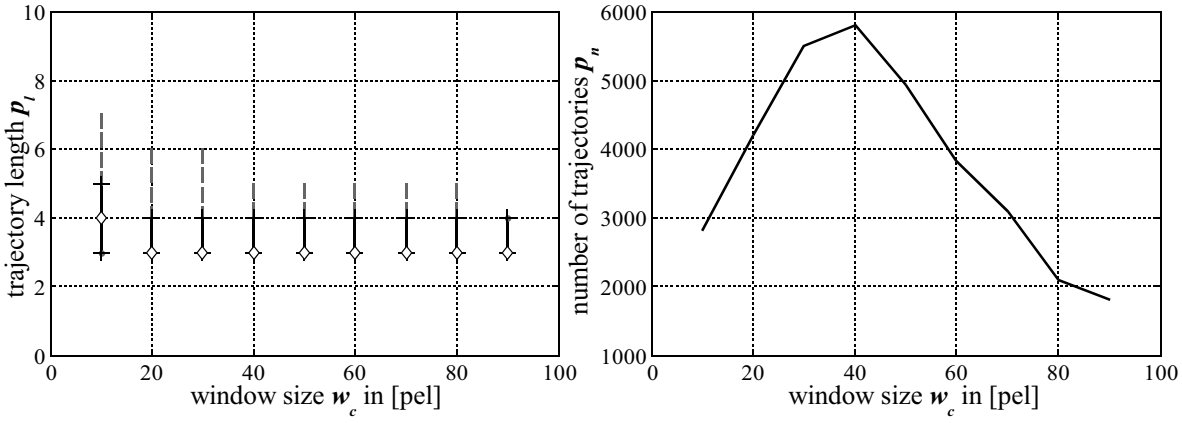


Figure 3.23: Trajectory length p_l as a function of the convolution window size w_c (left): The diagram shows the median length (diamond markers), the 25- and 75-percentiles (solid lines) and the 5- and 95-percentiles (dashed lines), estimated from all sequences. The right side shows the number of trajectories p_n as a function of w_c.

Thus, perspective distortions do not reflect in region shape, as with the affine-covariant detectors. Figure 3.24 (right) shows a notable decrease in d_o with increasing window size. For a setting of $w_c \geq 80\ pel$, the area overlap error stays below $d_o \leq 20\ \%$ for $95\ \%$ of all evaluated trajectories. Compared to the results in figure 3.18, the KLT-tracker is inferior for small window sizes, but draws level with most detectors for larger settings of w_c with respect to d_o. However, the processing time increases considerably for large settings of w_c, as more measurements have to be considererd during parameter estimation.

Robustness against perspective distortion In figure 3.25.1, the robustness of the KLT-tracker against increasing perspective distortion of the observed objects has been evaluated. Using the automated turn-table from the measurement setup, the rotation angle increment between two frames has been set to $5°$. The curves show both the number of tracking inliers p_i (left) and the number of matches (right) as a function of the overall turn-table rotation ϕ. All experiments have been performed with a fixed setting for the convolution window size at $w_c = 40\ pel$, which corresponds to the maximum number of trajectories in figure 3.23. As can be seen, p_i decreases from $\approx 100\ \%$ at $\phi = 5°$ to almost $0\ \%$ at $\phi = 30°$. The same experiment has also been conducted previously in section 3.4.2 using the affine-covariant detectors instead. There, the inlier percentage never fell below $p_i = 40\ \%$, even for a significant perspective distortion at $\phi = 30°$. For SIFT-based tracking, p_i even stayed above $60\ \%$ (except for MSER-regions).

Similarly, the number of matches (right) also decreases rapidly with increasing ϕ from

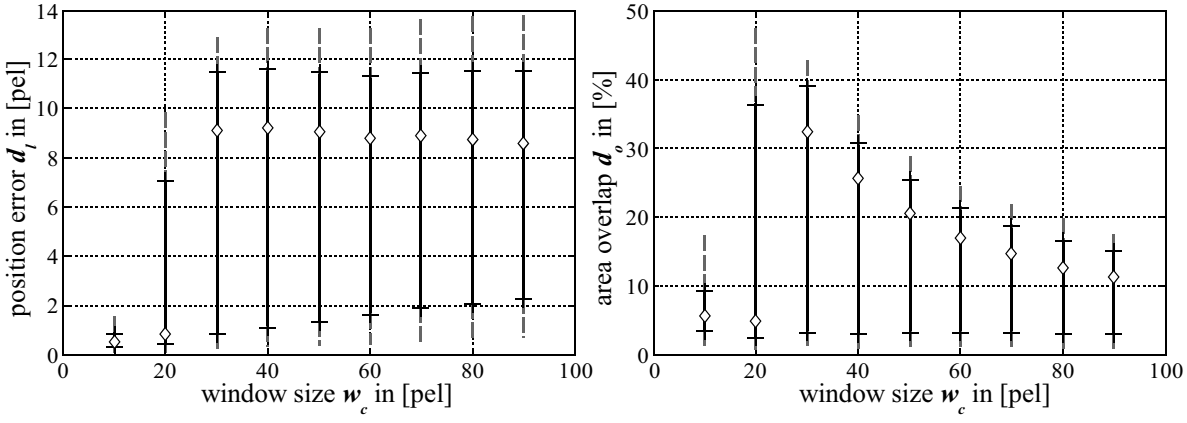


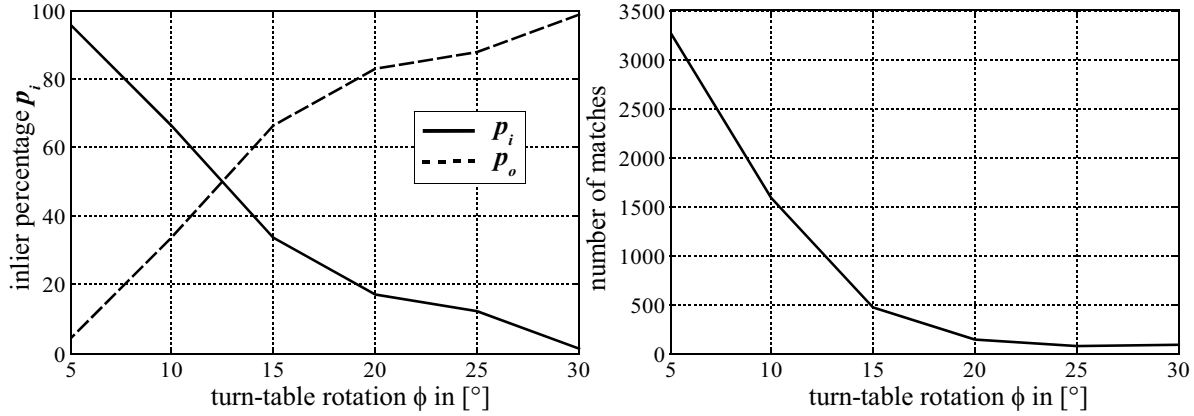
Figure 3.24: Position error d_l (left) and area overlap error d_o (right) as a function of the convolution window size w_c: The diagrams show the median errors (diamond markers), the 25- and 75-percentiles (solid lines) and the 5- and 95-percentiles (dashed lines), estimated from all sequences.

well over 3000 at $\phi = 5°$ to almost zero at $\phi = 30°$. A comparably steep decrease could not be found for either one of the detectors in figure 3.20. For a rotation angle of $\phi = 5°$ however, the KLT-tracker can easily compete with the best affine-covariant detector (HESAFF). At $\phi = 15°$ however, p_n is already lower for KLT than for any of the five detectors.

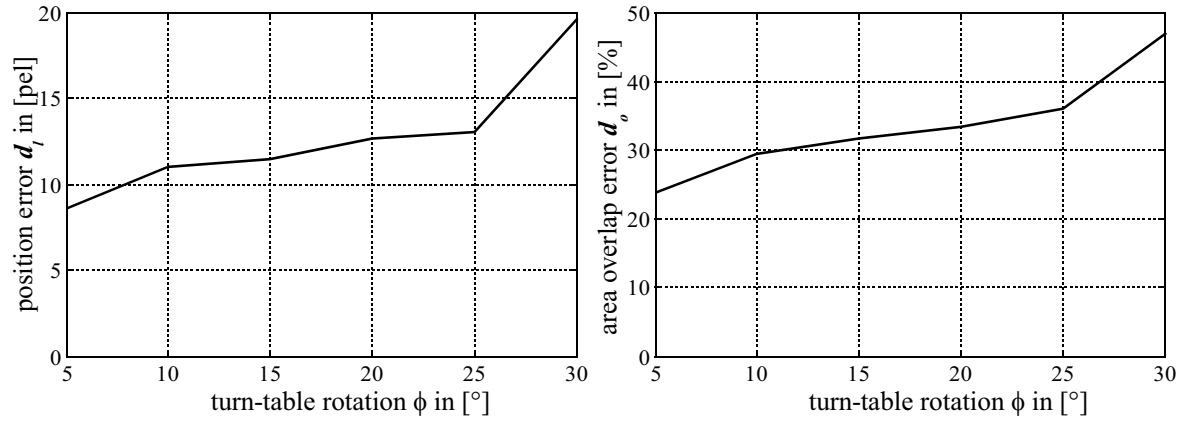
Finally, the dependency of the median position error d_l and the median overlap error d_o on turn-table rotation has been evaluated in figure 3.25.2. As ϕ increases, both measures also rise accordingly. Both steepness and curve progression are well comparable to figure 3.21, where the same behavior can be observed for all affine-covariant detectors.

Also, the KLT-tracker is less robust against increasing perspective distortions than the affine-covariant detectors from the previous section.

Concluding, the performance of the KLT-tracker strongly depends on the selected convolution window size w_c, which has to be predefined manually and does not change during the entire tracking procedure. Thus, the tracker is limited in its ability to react to significant changes of feature scales. With regard to the percentage of inliers p_i, KLT has shown clearly inferior for small settings of w_c. For larger settings, p_i increased to a rate of almost $100\ \%$, which is superior to affine-covariant tracking. Considering both length and number of the detected trajectories however, the KLT-tracker is clearly inferior. KLT could draw level for few combinations of detector types, descriptor distances and convolution window sizes, but was outperformed otherwise. The same was observed for both d_l and d_o. With increasing perspective distortion, the performance of the KLT-tracker rapidly decreased, with both p_i and p_n dropping to near zero for $\phi > 25°$.



3.25.1: Percentage of inliers p_i and number of matches.



3.25.2: Median position error d_l and overlap error d_o.

Figure 3.25: The above measures are evaluated as a function of the rotation angle for a fixed window size of $w_c = 40\ pel$.

3.5 Chapter Conclusion

One of the major contributions of this chapter is the comparison of three descriptor-based multi-region tracking algorithms, which operate on sets of affine-covariant regions with methods of increasing complexity. All three methods involve two major stages, which are region detection and correspondence assignment. While the detection of affine-covariant regions has been discussed extensively in chapter 2, this chapter was mainly concerned with the latter stage.

The first (and most simple) proposed tracking method searches for every region in the current frame an appropriate correspondence within a circular gating region of predefined size around the same position in the next frame. Potential ambiguities between regions are resolved for each frame pair using a global optimization technique (*local tracking*). The second tracking method employs a model-based state prediction of the presumed feature location within the next frame using a Kalman-filter in order to narrow the search area and thus reduce the number of correspondence candidates (*KF-based tracking*). As with local tracking, potential ambiguities between regions are resolved using the same global optimization technique. It has been shown, that both methods exhibit a relatively large number of tracking outliers (*i.e.* region correspondences with an area overlap error of $d_o \geq 0.5$).

To this end, a third (and novel) tracking method is proposed, which keeps all correspondence candidates (including ambiguities) until the end of an image sequence (or for a predefined number of successive frames) and integrates these into several weighted and directed graphs (*graph-based tracking*). Transitions from one region to the next (*i.e.* graph-edges) are weighed using both descriptor distance and a motion model. The latter is based on a well-known model from the literature and has been extended within this work to integrate additional region information supplied by the affine-covariant detectors. Thus, instead of prematurely selecting the locally most probable candidate (as with local and KF-based tracking), decisions on unique correspondences are postponed until sufficient global evidence has been gathered that allows for an improved extraction of trajectories by means of weighted graph-traversal.

The main purpose of the evaluation within this chapter has been three-fold. Firstly, the three above-mentioned tracking methods have been compared to each other with regard to a number of appropriate performance measures. Secondly, as each method has been implemented using the five affine-covariant region detectors from the previous chapter, the suitability of the latter for descriptor-based tracking has been additionally evaluated. Also, a detailed comparison to the well-known KLT-tracker has been presented. In the following, the major results are briefly summarized.

In order to evaluate the performance of the resulting trajectories, a number of adequate measures has been defined which includes the percentage of inliers p_i to the groundtruth homographies relating the object surfaces between adjacent frames (*i.e.* correspondences with an area overlap error of $d_o \leq 0.5$), the distribution of trajectory length p_l, the

number of resulting trajectories p_n and the localization accuracy in terms of position error d_l and area overlap error d_o. Also, robustness under varying perspective distortion has been evaluated by increasing the inter-frame angular increment ϕ of the automated turn-table.

For all three tracking methods, comparable results are achieved with regard to the inlier percentage p_i. If SIFT-descriptors are used for correspondence assignment, p_i is well above $95\ \%$ for all investigated settings of the maximally permissible descriptor distance $d_{d,max}$. The performance of graph-based tracking however is slightly better than of local or KF-based tracking, with $p_i \geq 98\ \%$ for all five detectors. If the less distinctive SPIN-descriptors are used for correspondence assignment instead of SIFT, p_i shows a steep decline with increasing $d_{d,max}$ (for local and KF-based tracking). For graph-based tracking however, the performance of SIFT-based and SPIN-based tracking is almost identical, with an inlier percentage above $p_i \geq 97\ \%$ for all settings of $d_{d,max}$. Among the five detectors, MSER shows the best performance with significant distance to the second-best candidate.

With respect to the number of trajectories p_n, the HESAFF-detector provides the best results with all three tracking methods, followed in considerable distance by EBR. Among all detectors, MSER performs worst with only $\approx 17\ \%$ of the number of trajectories attained with HESAFF-based tracking. Notably, differences between using the SIFT- or SPIN-descriptor with regard to p_n are very small. Whith KF-based tracking, p_n is slightly larger compared to local tracking, but only for HESAFF and EBR. For graph-based tracking, the number of trajectories is approximately $10\% - 20\%$ smaller than with the other two methods.

Considering the distribution of trajectory lengths p_l, MSER-based tracking provides the longest trajectories with all three methods, followed with distance by the IBR-detector. Between the remaining detectors, differences are only minor. If the descriptor distances between the elements of a trajectory decreases, an increase in p_l can generally be observed. This behavior was found for all three tracking methods. Notably, no significant differences could be observed between the results of local and KF-based tracking while for graph-based tracking, p_l was slightly larger. This observation is consistent with the smaller number of resulting trajectories provided by the graph-based method, assuming an identical number of detected regions in each frame.

Further, it has been shown that there exists a dependency between both position error and area overlap error and the distance of region descriptors among the elements of a trajectory. If the latter is large, the probability for a high error is significantly increased. This observation could be made for all three tracking methods. While differences are generally small between local and KF-based tracking, the graph-based method exhibits a significantly lower error for all five detectors and for both SIFT- and SPIN-descriptors. The relative ordering of the affine-covariant detectors (with all tracking methods) generally corresponds to the evaluation results presented in the previous chapter. While MSER showed the lowest errors, EBR performed worst closely followed by IBR.

Lastly, the robustness of the affine-covariant detectors against perspective distortions of the image content has been evaluated. It could be shown, that with increasing turn-table rotation ϕ, both the percentage of inliers as well as the number of matches decreased. Until $\phi \approx 20°$, the decline was still moderate for all detectors and both SIFT- and SPIN-descriptors. For $\phi > 20°$, the steepness of the decline increased significantly for both measures. Secondly, the position error d_l and the area overlap error d_o have been testes against increasing ϕ. For all five detectors, an increase in both error measures was found. As before, the MSER-detector performed best in this respect. Between SIFT- and SPIN-descriptors however, differences were generally small.

In order to assess the performance of affine-covariant region tracking with respect to the existing literature, a comparison to the well-known KLT-tracker was additionally presented. It could be shown, that the performance of the latter highly depends on the convolution window size w_c, which has to be preselected manually and does not change during the entire tracking procedure. Thus, the tracker is limited in its ability to react to significant changes of feature scales. With regard to the percentage of inliers p_i, KLT has shown inferior to affine-covariant tracking for small settings of w_c. For larger settings, p_i increased to a rate of almost $100\ \%$, which is superior to affine-covariant tracking but comes at the price of greatly increased computational complexity, as more measurements have to be considered during parameter estimation. With regard to both length and number of the detected trajectories however, the KLT-tracker has shown clearly inferior. For few combinations of detector types, descriptor distances and convolution window sizes, KLT could draw level but was outperformed otherwise. The same behaviour has been observed with regard to both d_l and d_o. With increasing turn-table rotation (and thus perspective distortion), the performance of the KLT-tracker rapidly decreased. For $\phi > 25°$, both p_i and p_n dropped to near zero. In this respect, the affine-covariant detectors in conjunction with the proposed tracking approaches have shown superior as well.

Concluding, the traditional KLT-tracker has shown inferior to all three methods based on affine-covariant regions, especially in the case of significant perspective distortions. Among the latter, the use of Kalman-filtering to predict the presumed location of regions in future frames improved the tracking performance slightly, especially with regard to the number of trajectories and to the percentage of inliers. The newly introduced graph-based method showed clearly superior with respect to most of the considered measures, including (and in particular) localization accuracy. Only the number of resulting trajectories was inferior to local and KF-based tracking. Thus, if one can afford the computational burden of graph-traversal and the delayed availability of tracking results, the graph-based method should be preferred.

The presented tracking concepts and the evaluation results serve as a prerequisite for the subsequent chapter 4, but may also be used in a self-contained way for the selection of an appropriate appropriate tracking method and a suitable affine-covariant detector for a given target application. The major aspects have also been published in [HJA08a].

4 Binocular Region Tracking and Spatial Reconstruction

4.1 Chapter Introduction

In the previous chapter, several monocular feature tracking techniques have been presented. Given a single camera, the two-dimensional motion of a set of affine-covariant regions has been observed by assigning inter-frame correspondences on the basis of region descriptors and by appropriately resolving ambiguities. Within this chapter, the second camera of the measurement setup is additionally used to extend the analysis of object motion to three dimensions. To this purpose, a novel technique is proposed which takes as input two sets of monocular feature trajectories (one from each camera of the stereo setup) and performs a correspondence analysis such that the most similar trajectories between both camera views are assigned to each other. The resulting matches are three-dimensional and thus represent the spatial motion of the associated objects (in conjunction with the camera calibration parameters).

In the literature, methods for three-dimensional scene reconstruction are often referred to as *stereo techniques*, *e.g.* in [FP02]. Generally, these involve three main steps. Firstly, a two-camera setup has to be calibrated with regard to both intrinsic and extrinsic parameters as described in chapter 2. Secondly, corresponding elements between both cameras have to be identified for each pair of images. To this respect, a classification of appropriate techniques into two broad categories is usually made in the literature: With *intensity-based* methods, correspondence search is performed based on intensity profiles of the two images, while with *feature-based* techniques, appropriate salient locations (features) are first extracted from the images and correspondence search is applied to these instead. A short overview of both intensity-based and feature-based techniques is presented within the background section of this chapter. Thirdly, each identified correspondence between both cameras is projected into the three-dimensional world coordinate system by using the calibration parameters from the first step. The result is a three-dimensional point cloud in world space, which represents the scene in front of the camera setup. However, information on object motion is not provided by such methods as correspondences between neighboring frame pairs do not exist. The method proposed in this section thus extends conventional stereo techniques by establishing temporal links between the reconstructed 3D-points. This process is referred to as binocular tracking.

Compared to monocular tracking, there exist two fundamental differences. Firstly, corresponding features are observed by different cameras. Depending on both mea-

surement setup and camera hardware, this might lead to synchronization errors, strong perspective distortions between the two viewpoints (foreshortening) and different imaging characteristics (such as brightness, contrast or sensor noise). Secondly, instead of searching for potential correspondences within a two-dimensional gating area, a one-dimensional search along the respective epipolar line is sufficient (given the camera calibration parameters). Depending on the number and density of identified features within a scene, this limits the number of matching candidates and thus the potential for ambiguities.

In this work, only feature-based stereo matching is considered. Intensity-based methods have already been subject to extensive evaluation, *e.g.* in [SS02]. Given the five affine-covariant detectors from the previous chapters, correspondence search between both cameras is performed using histogram-based region descriptors. As with monocular tracking, either the SIFT- or SPIN-method are used for this purpose. Potential ambiguities between both cameras are resolved using a combinatorial optimization approach, such that the final set of inter-camera region correspondences is unique.

However, as with monocular tracking, conventional optimization is based on a single frame pair only. In the case of high region density and low descriptor distinctiveness, the number of mismatches between similar regions is usually very high. Depending on both focal length and distance of the two cameras from each other, such mismatches may lead to severe errors in spatial depth of the reconstructed points. One of the major contributions of this chapter is thus the proposition of a novel stereo technique, which greatly reduces the number of such matching errors, especially with less distinctive region descriptors.

Basically, the proposed binocular tracking technique may be decomposed into three major stages: (1) monocular region detection and tracking, (2) trajectory correspondence assignment and (3) spatial reconstruction. While the first stage has been extensively discussed in the previous chapters already, the current chapter is concerned with the latter two. In order to provide a sound basis for resolving ambiguities between both cameras, the conventional feature-to-feature matching on the basis of a single frame pair is extended to trajectory-to-trajectory matching based on multiple frame pairs. The new method works as follows: Firstly, monocular tracking is performed separately in both cameras of the stereo setup according to the graph-based algorithm introduced in the previous chapter. Secondly, potentially corresponding features are searched between both cameras, based on descriptor similarity. Each match corresponds to a three-dimensional point in the world coordinate frame. Similar to graph-based monocular tracking, every reconstructed point is inserted as a new node into a graph structure. In the case of matching ambiguities, each one is treated as a single graph node. Based on the set of monocular feature trajectories from the first step, edges are introduced into the graph such that two nodes are connected, if either one of the associated affine-covariant regions belong to the same two-dimensional trajectory. Finally, unique spatial trajectories are extracted by means of weighted graph traversal using the same techniques as in the

previous chapter. The basic idea of matching corresponding trajectories instead of single features between two views of a scene has also been investigated in [NJ93] and [NJW92] in the context of particle tracking velocimetry.

In addition to an improved matching uniqueness, the proposed method also provides information on the spatial motion of the observed objects, as correspondences between temporally adjacent frames are supplied by monocular tracking in the first stage of the algorithm. It will be shown in an extensive evaluation, that the new method is superior compared to conventional feature-based stereo matching.

This chapter is organized as follows: Firstly, a brief overview of stereo techniques from the literature is presented in background section 4.2. Secondly, a traditional stereo technique based on feature-to-feature matching and the above-mentioned novel method based on matching trajectories are presented in section 4.3. Finally, an evaluation of the reconstruction results is given in section 4.4. A summary of the major results concludes this chapter in the last section 4.5.

4.2 Background: Overview of Stereo Matching Techniques

Correspondence and reconstruction The projection of rays of light onto the sensor of a digital camera produces an image of the world, that is inherently two-dimensional. In order to reconstruct information on the depth of an observed object, a second image from a different viewpoint is needed. Given two such images I_l and I_r, two problems have to be solved:

1. For a given point $\mathbf{x_1} = [x_1,\ y_1]^T$ in I_l, determine to which point $\mathbf{x_2}$ in I_r it corresponds to. The term *corresponds* means that both $\mathbf{x_1}$ and $\mathbf{x_2}$ are images of the same physical point $\mathbf{X} = [x,\ y,\ z]^T$ in world space. This is what is commonly known as *correspondence problem* in the literature [Fau93][HZ03].

2. Given two corresponding points $\mathbf{x_1}$ and $\mathbf{x_2}$, compute the 3D-coordinates of $\mathbf{X}$ relative to a global reference coordinate system. This is generally known as the *reconstruction problem*.

Disparity and depth The most simple configuration of the two cameras of a stereo system is the parallel case. In this configuration, the two image planes are horizontally displaced and coplanar in space. Also, both cameras have identical focal length. Given a scene point $\mathbf{X}$ and its two projections $\mathbf{x_1}$ and $\mathbf{x_2}$ as shown in figure 4.1 (left), the *disparity* d between both is defined as

$$d = x_2 - x_1\ . \tag{4.1}$$

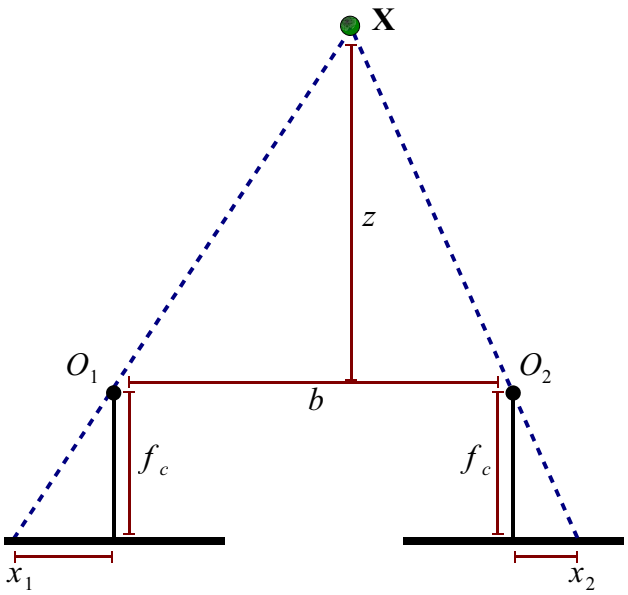


Figure 4.1: Stereo normal camera configuration. A spatial point $\mathbf{X} = [x,\ y,\ z]^T$ in the world coordinate system is projected into two parallel cameras with identical focal length f_c. The dashed lines represent rays of light from $\mathbf{X}$ through the camera centers O_1 and O_2 onto the image plane. There, they appear as the images of $\mathbf{X}$ at horizontal positions x_1 and x_2. The base width b denotes the absolute distance between the two camera centers.

As there exists no vertical parallax between the cameras, the relation $y_1 = y_2$ holds. The depth measure z of $\mathbf{X}$ is related to the disparity d as

$$z = \frac{f_c \cdot b}{d} \,, \tag{4.2}$$

where f_c is the focal length of both cameras and b represents the absolute distance distance between the camera centers O_1 and O_2. The latter is also referred to as the *base width* of the stereo setup. In parallel camera configurations (also called *stereo normal configuration*), correspondence search can be restricted to the horizontal (and thus one-dimensional) scan lines of the images. Thus, the search for potential correspondences is greatly simplified. In most applications however, cameras are not arranged in a true stereo normal configuration. If however both extrinsic and intrinsic calibration parameters are known, the taken images may be transformed into the latter. Given the calibration data from section 2.4, this has been performed for all evaluated images sequences. Extensive information on stereo normal configurations an on epipolar geometry can be found in [Fau93] or [HZ03].

Intensity-based stereo matching Approaches to the correspondence problem may be broadly classified into two categories, *i.e. intensity-based* and *feature-based*. Methods from the first category perform the matching procedure directly on the intensity profiles of the two images. With the second category however, features are first extracted from the images and the matching procedure is applied to the features instead. One of the earliest attempts to intensity-based matching was proposed by Barnard in [Bar87], where an energy function was minimized using simulated annealing as an optimization procedure. Robert [RDF92] later proposed the use of a multi-resolution scheme in conjunction with a smoothness constraint. In addition to a pure horizontal disparity, this approach also allowed for a vertical parallax. The major advantage of using intensity-based methods is a dense disparity map, which provides information on the depth of every pixel in the image. Unfortunately, convergence of said energy functions to the correct minimum is not always ensured, leads to spatial reconstruction errors. Also, not every location in an image provides sufficiently distinctive information to allow for a unique matching. An alternative approach in intensity-based stereo matching is the *window-based method*. The general idea is to match only locations in an image, that contain a sufficiently high variation of the intensity signal in both coordinate directions. One of the first approaches to detecting such salient locations is Moravec's interest operator [Mor79], which was widely used in a great variety of stereo matching systems. Once interesting points have been detected, a simple correlation scheme is applied in the matching process in order to assign regions that are sufficiently correlated.

One of the major problems of window-based stereo-matching is to find an adequate size of the correlation window. If the latter is chosen too small, intensity variations within will not be distinctive enough to allow for an unambiguous localization. In this

case, a high number of false matches results. If the window size is chosen too large instead, resolution and thus accuracy is lost, since neighboring regions with possibly different disparities (*e.g*. at object borders) will be considered jointly. Also, in the case of significant perspective distortions of the image content, a rectangular or circular window is an insufficient representation. This effect has already been discussed in chapter 2. There, figure 2.3 illustrates the problem. An extensive overview and comparison of intensity-based stereo matching techniques may be found in [SS02].

Feature-based stereo matching With feature-based approaches, each pair of images from the two cameras is first preprocessed by an appropriate detector, which extracts features that are stable under viewpoint changes. Instead of directly matching the intensity values as with the intensity-based method, a normalized multi-dimensional descriptor is computed for each feature. The matching process is then reduced to measuring the euclidean distance between the associated descriptors. Within this chapter, the five affine-covariant region detectors introduced in section 2.2.2 are used within the detection step. As in the previous chapter, either SIFT- or SPIN-descriptors are used for region description. In contrast to window-based matching, affine-covariant detectors are able to automatically adapt both size and shape of the support region to the underlying image content. Thus, the premature setting of a constant window size as with most intensity-based methods is avoided. Given their obvious advantages, only feature-based techniques are considered for evaluation within this work. An overview of feature-based stereo matching techniques may be found in [LT94].

4.3 Concepts for Feature-based Stereo Matching

4.3.1 Single-Frame Stereo Matching

In this section, a concept for the three-dimensional reconstruction of two-dimensional affine-covariant regions given two camera views is presented. Figure 4.2 illustrates the three basic steps of the algorithm, which are introduced in the following.

Data association In the first step, potential correspondences between the regions detected in both cameras have to be identified for every frame pair. Similar to the region tracking methods discussed in the previous chapter, descriptor distance between two regions is used as a similarity measure. Additionally, the perpendicular deviation from the epipolar line (*i.e*. the scan line) is used. The greater the vertical parallax, the higher will be the depth error of the reconstructed point. Given two potentially corresponding regions $r_{i,1}^m$ and $r_{i,2}^n$ within the frames $I_{i,1}$ and $I_{i,2}$ respectively (where the subscripts 1 and 2 refer to the first and second camera) , the epipolar distance d_e is defined as

$$d_e = |y_m - y_n| \,, \tag{4.3}$$

which is the absolute difference between the y-positions of the region centers within the images. As indicated in figure 4.2, the set of all correspondence candidates $\mathcal{L}$ may well contain ambiguous elements, *i.e.* the same region within the first camera is claimed by several regions within the second camera or vice-versa. In the data association step however, dependencies between competing regions are not yet resolved. This is done in the subsequent optimization step. In the end, each correspondence must be unique, as both associated regions are images of the same physical object.

Combinatorial optimization If for a given region from the first camera there exist several potential correspondences along the epipolar line with sufficiently similar descriptors, correspondence ambiguities arise. There may occur two cases: *Firstly*, a region in the first camera may find several potential correspondences in the second camera. In this case, the most similar candidate (with regard to d_d and d_e) is to be chosen while the remainder of dependent candidates is discarded. This approach is generally referred to as *nearest-neighbor assignment*, because the final region correspondences are nearest to each other in terms of their euclidean distance in descriptor space. *Secondly*, a region in the second camera may be additionally claimed by several regions from the first camera. In this case, the simple nearest-neighbor assignment is no longer guaranteed to provide the best possible results. Depending on the region processing order and on the dependencies between both cameras, it is well possible, that the local decision on the optimal candidate for a specific region might entail several sub-optimal assignments. Thus, it would be more favorable to find a global solution to the region-to-region assignment problem for each pair of frames, which considers all potential pairings at the same time. To this end, a combinatorial optimization approach is needed, which analyzes the interconnections between the regions of two frames and at the same time provides *the* subset of correspondences which globally minimizes the overall d_d and d_e. In the previous chapter, such a method has already been used in the context of monocular region tracking, based on the well-known *Hungarian algorithm*. Details on the latter may thus be found in section 3.3.1. The result of the combinatorial optimization procedure based on this method is a reduced set of unique correspondences $\mathcal{L}_0$ with minimal overall descriptor and epipolar distance.

Spatial reconstruction Given a unique set of corresponding regions $\mathcal{L}_o$ between both cameras, spatial reconstruction is performed on the basis of equation 4.2. As shown in the measurement setup from figure 2.10, the world coordinate center is defined as the origin of the bottom camera O_1. As can be seen in figure 4.2 (bottom-left), the result is a three-dimensional point cloud, which qualitatively reflects the rotation of a planar object on the automated turn-table. However, information on the motion of the observed object is not available with this approach, as correspondences only exist between both cameras and for a single time step. In order to reconstruct the spatial motion of a re-

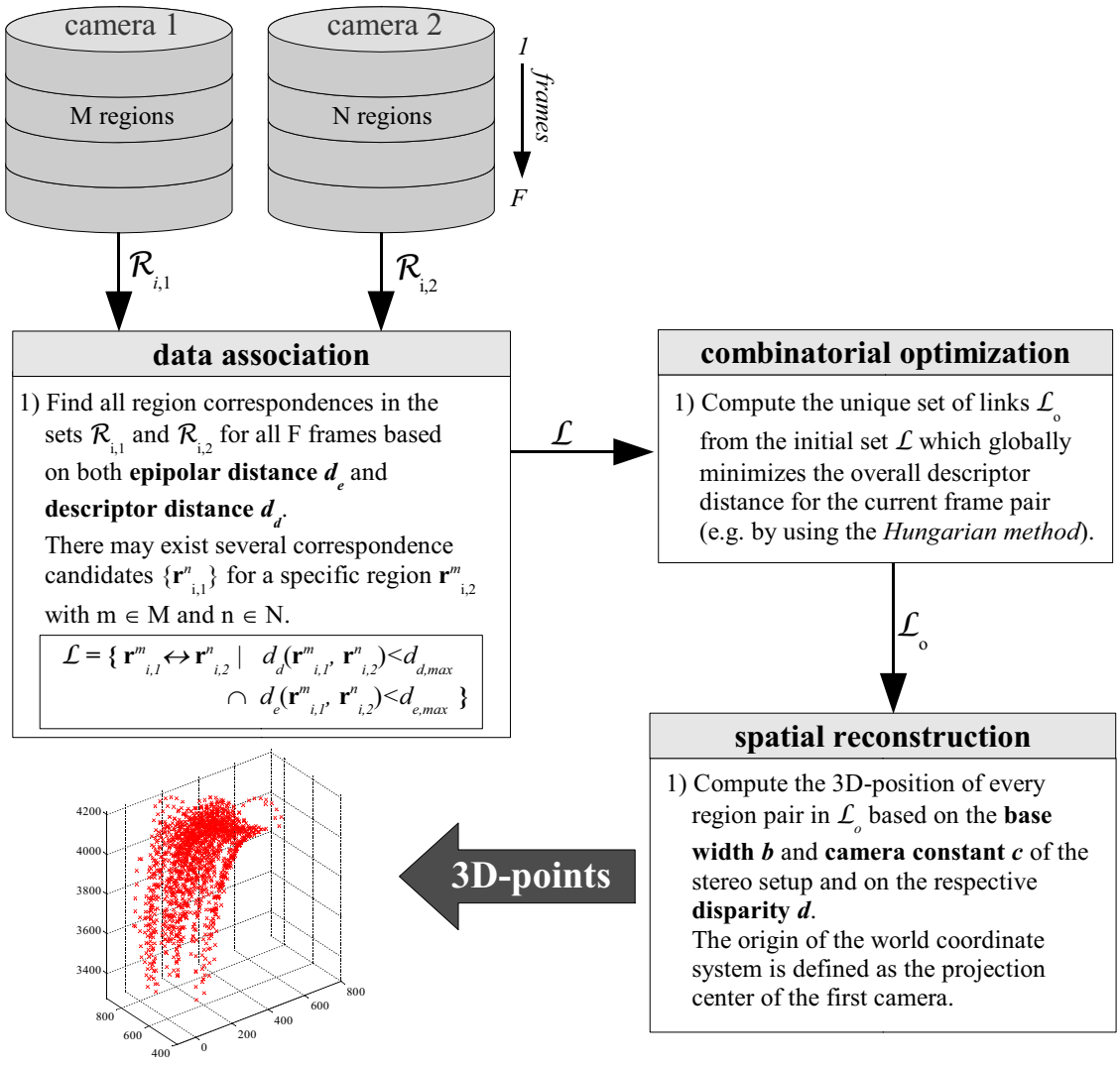


Figure 4.2: Algorithmic flow of descriptor-based region-to-region stereo matching: Firstly, potential correspondences for all regions from the first camera are searched along the respective epipolar lines (which coincide with the scan lines for stereo normal configurations), yielding a set of initial (and potentially ambiguous) correspondence candidates $\mathcal{L}$. Secondly, a subset $\mathcal{L}_0$ of these candidates is extracted such that only unique correspondences exist and the overall descriptor distance d_d and the epipolar distance d_e of $\mathcal{L}_0$ are minimal. These two goals are achieved by means of combinatorial optimization using the Hungarian method. Thirdly, spatial reconstruction of every correspondence is performed based on base width and focal length of the stereo setup.

constructed region-pair, additional information on the movement of both within their respective camera is needed.

4.3.2 Multi-Frame Graph-based Stereo Matching

Although the combinatorial optimization of ambiguous correspondences provides the best possible assignment between the regions of both cameras, only a single time-instant is considered with the previously discussed method. Hence, in the case of high region density and low descriptor distinctiveness, the number of mismatches between similar regions is usually very high. Depending on both focal length and the base width of the stereo setup, such mismatches may lead to severe errors in spatial depth of the reconstructed points. To this end, a novel stereo matching technique is proposed, which greatly reduces the number of such matching errors (especially with less distinctive region descriptors) and at the same time provides spatial trajectories instead of mere 3D-points. Figure 4.4 illustrates the basic steps of the algorithm, which are introduced in the following.

Graph-based monocular region tracking In the first step of the algorithm, monocular region tracking is performed in each camera separately, using the graph-based method from the previous chapter. The two resulting sets of unique two-dimensional trajectories $\mathcal{T}_1$ and $\mathcal{T}_2$ are then forwarded to the data association stage. Basically, every other tracking method might be used for this purpose (*e.g.* the KLT-tracker). However, as will be seen later, the graph-based method from section 3.3.3 is most appropriate, since many of its functions are also used for the proposed graph-based stereo matching technique and thus fit nicely into the overall algorithmic concept. Two sets of example trajectories for both the bottom- and top-camera are shown in figure 3.11 in the previous chapter.

Data association Given the two sets of trajectories T_1 and T_2 from the first and second camera, dependencies between both sets are identified in this step. Two trajectories are deemed dependent on each other, if they share at least one potential region pair. The latter are identified (as with the previous method) based on both descriptor distance d_d and on the deviation from the epipolar line d_e. The result of this step is a set of disjoint groups $\mathcal{D}$ of interdependent trajectories: Within each group in $\mathcal{D}$, all trajectories share at least one common region. Between two groups from $\mathcal{D}$, no such dependencies exist. This separation is necessary in order to reduce the complexity of combinatorial optimization, which is performed during graph construction.

Graph construction Given a set of interdependent trajectories in $\mathcal{D}$, the 3D-position of every region correspondence is computed using the base-width b and the respective

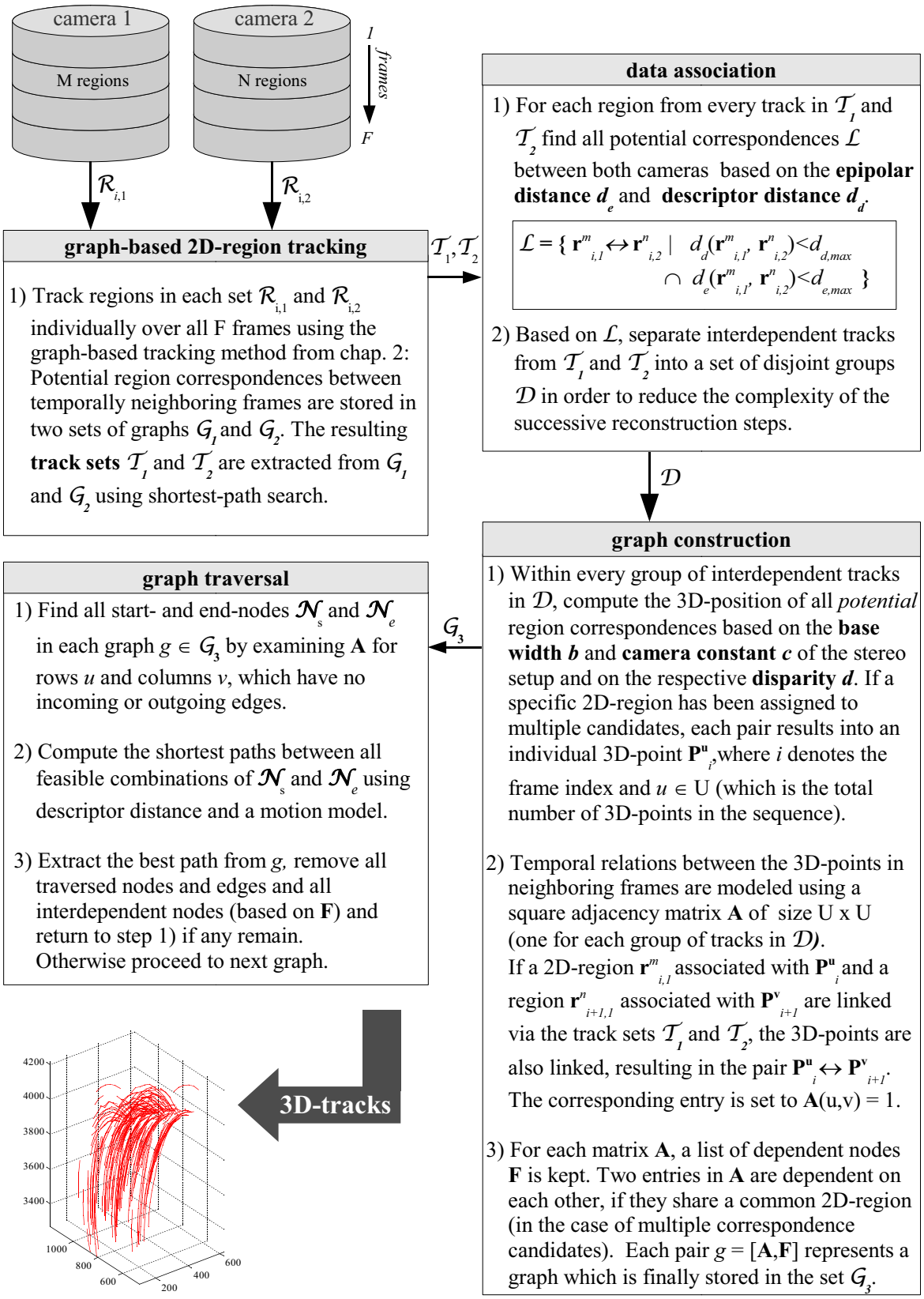


Figure 4.3: Algorithmic flow of graph-based stereo matching.

disparity d. As no uniqueness has been enforced yet, the set of correspondences may still contain ambiguities. Thus, if a specific region in the first camera has been assigned to multiple regions in the second camera, each pair results into an individual 3D-point $\mathbf{P}_i^u$, where i denotes the frame index and $u \in U$ (the total number of 3D-points in the sequence). Figure 4.4 illustrates the principle. There, a region $r_{i,1}^1$ in the left camera has two potential correspondences $r_{i,2}^1$ and $r_{i,2}^2$ in the right camera (red markers). From the two pairs $r_{i,1}^1 \leftrightarrow r_{i,2}^1$ and $r_{i,1}^1 \leftrightarrow r_{i,2}^2$, the 3D-points $\mathbf{P}_i^1$ and $\mathbf{P}_i^2$ are reconstructed. It is obvious, that a coexistence of both is not physically feasible. In order to enforce uniqueness among the reconstructed points later in the algorithm, a dependency list $\mathcal{F}$ is incrementally built. In conventional stereo reconstruction, region matching between the two cameras is performed anew with each new time step. The result after processing an entire sequence of frames is a temporally independent 3D-point cloud in world space. In the proposed approach, temporal matching is additionally performed, based on the available monocular region trajectories from the previous step. As illustrated in figure 4.4, $\mathbf{P}_i^1$ is connected to $\mathbf{P}_{i+1}^3$ based on a two-dimensional intra-camera correspondence from region tracking (gray line). In order to assign two nodes, it is sufficient that the corresponding regions in a single camera are connected to each other. Thus, if a trajectory is incomplete or split in one camera, there may still exist a spatial connection, if the associated regions in the second camera have been successfully tracked. It will be shown later, that the resulting spatial trajectories are superior to the monocular trajectories with regard to their length.

Formally, if a region $r_{i,1}^m$ associated with $\mathbf{P}_i^u$ and a region $r_{i+1,1}^n$ associated with $\mathbf{P}_{i+1}^v$ are linked two-dimensionally via the track sets $\mathcal{T}_1$ and $\mathcal{T}_2$, the resulting 3D-points are also linked to each other. Such temporal relations between 3D-points in neighboring frames are modeled using a square adjacency matrix A of size $U \times U$. For each correspondence $\mathbf{P}_i^u \leftrightarrow \mathbf{P}_{i+1}^v$, the respective entry is set to $A(u, v) = 1$. This procedure corresponds in principle to the graph-based tracking approach from the previous section. In analogy to the latter, all spatial points $\mathbf{P}$ are considered as graph nodes, while the tracking-based dependencies among them are introduced as graph edges. For each group of trajectories in $\mathcal{D}$, a matrix A is constructed along with a list of interdependent nodes $\mathcal{F}$. Finally, each pair $g = [A, \mathcal{F}]$ is stored in a list $\mathcal{G}_3$, which contains the same number of elements as in $\mathcal{D}$.

Graph traversal In this step, uniqueness is enforced within each element of $\mathcal{G}_3$ by removing a subset of ambiguous graph nodes by means of shortest-path search. To this purpose, all start nodes $\mathcal{N}_s$ and end nodes $\mathcal{N}_e$ are detected for each $g \in \mathcal{G}_3$ by examining the adjacency matrix A for rows u and columns v, which have no incoming or outgoing edges, respectively. Then a shortest-path search between all feasible combinations of elements from $\mathcal{N}_s$ and $\mathcal{N}_e$ is performed using Dijkstra's algorithm as described in section 3.3.3. A combination is considered feasible, if the time index is smaller for the start node than for the end node. By enforcing this constraint, the number of potential pairings

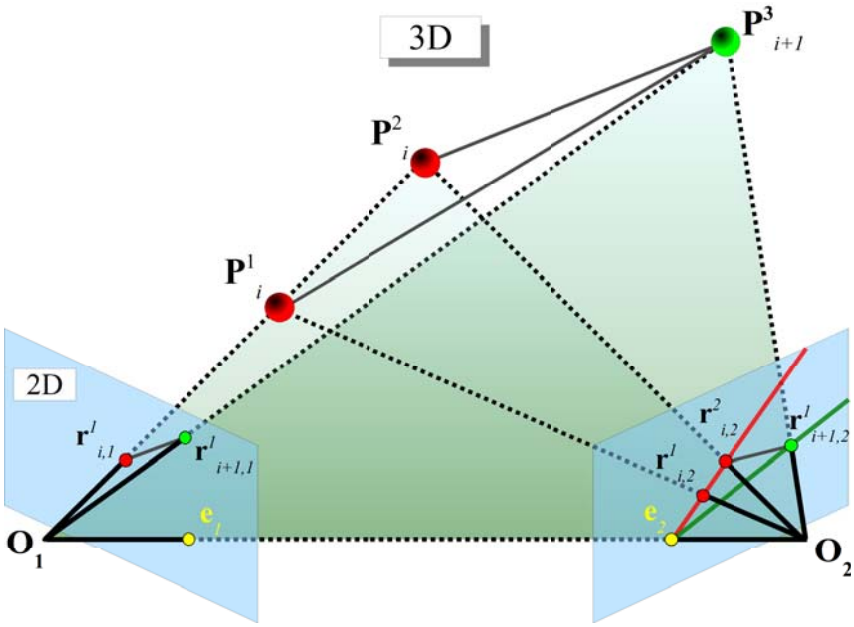


Figure 4.4: Stereo graph construction: For a given region $r^1_{i,1}$ in the left camera, two potential correspondences $r^1_{i,2}$ and $r^2_{i,2}$ exist along the epipolar line within the right camera (red line). For both pairs, spatial points $\mathbf{P}^1_i$ and $\mathbf{P}^2_i$ are constructed. As a result from monocular region tracking within each camera, a temporal correspondence $r^1_{i+1,1}$ (green) has been assigned to $r^1_{i,1}$ (gray line). Thus, the spatial points $\mathbf{P}^1_i$ and $\mathbf{P}^3_{i+1}$ are also connected.

and thus computational complexity is reduced, especially with longer sequences. Edge transitions between two nodes are weighted using the descriptor distance between the associated regions (*i.e.* $d_d(r_{i,1}^m, r_{i,2}^n)$) and the motion model from equation 3.42. Among all traversals through A, the best path is extracted and kept as the first resulting spatial trajectory. Based on A and $\mathcal{F}$, all traversed and dependent nodes are then removed from the graph. On the reduced set of nodes, a second shortest-path search is performed. This process is repeated, until there are no nodes or edges (and thus trajectories) left in A. This procedure is performed on all elements in $\mathcal{G}_3$. After the last iteration, a set of unambiguous 3D-trajectories exists, which represents the spatial motion of the observed objects. In the next section, a thorough evaluation and comparison of both presented methods is performed.

4.4 Evaluation

4.4.1 Matching Performance Measures

In order to assess the quality of the reconstructed 3D-trajectories, a set of adequate measures has to be defined. In analogy to section 3.4.1 from the previous chapter, the resulting 3D-trajectories should ideally

1. be equivalent or superior to the set of 2D-trajectories from intra-camera monocular tracking with regard to their length.

2. reflect the 3D-surface of the observed objects (*i.e.* a plane).

3. reflect the measurement setup (*i.e.* a rigid object positioned on a turn-table with a fixed angular increment)

From the above list, a number of performance measures is derived, which are introduced in the following.

Trajectory lengths As described in section 4.3.2, the proposed algorithm for the spatial reconstruction of object motion finds temporal correspondences by combining monocular region trajectories from both cameras. In order to assign two spatial points, it is sufficient that the corresponding two-dimensional regions in a single camera are connected to each other. Thus, if a trajectory is incomplete or split in the first camera, there may still exist a spatial connection, if the associated regions in the second camera have been successfully tracked. Therefore, the distribution of spatial trajectory lengths p_l is compared against the results from monocular feature tracking. Ideally, the latter are inferior for both the first and the second camera. Additional information on the proposed measure may be found in section 3.4.1 of the previous chapter, where an evaluation of different concepts for monocular region tracking has been performed.

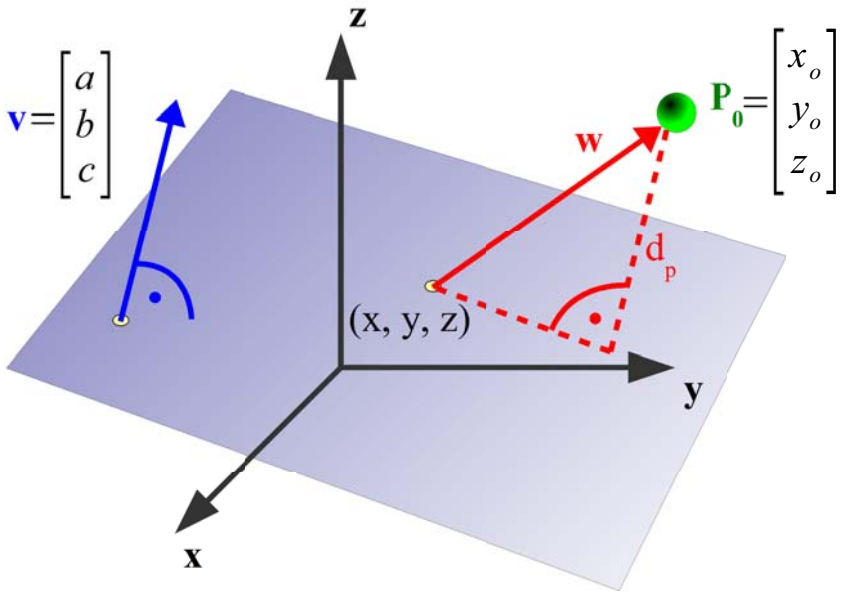


Figure 4.5: Distance $\mathbf{w}$ of a point $\mathbf{P}_0$ from a regression plane. In order to obtain the orthogonal distance d_p, vector $\mathbf{w}$ has to be projected onto the plane normal vector $\mathbf{v}$. The scalar entity d_p is used as an error measure for the assessment of spatial point accuracy.

Orthogonal distance from regression plane A commonality of all observed objects in figure 2.11 is the (approximate) planarity of their surfaces. As can be seen, the area for region detection has been limited by appropriate bounding polygonals (dashed white lines). Thus, all reconstructed 3D-points (at the same time instant) should theoretically be coplanar. In the following, a measure is derived based on a statistical model fit, which allows for the assessment of 3D-point accuracy in terms of their orthogonal distance from a regression plane. As mentioned before, the results of this evaluation should not be considered absolute, due to deviations of the object surfaces from perfect planarity. Instead, all curves should be seen relative to each other (*e.g.* the relative performance of five affine-covariant detectors for a given method, or of two methods for a given detector).

Given a plane in three dimensions of the form

$$ax + by + cz + d = 0 \tag{4.4}$$

and a point $\mathbf{P}_0 = [x_o, y_o, z_o]^T$ according to figure 4.5, the normal of the plane is defined as

$$\mathbf{v} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{4.5}$$

and a vector $\mathbf{w}$ from the plane to the point is given by

$$\mathbf{w} = - \begin{bmatrix} x - x_o \\ y - y_0 \\ z - z_0 \end{bmatrix} . \tag{4.6}$$

In order to compute the orthogonal distance of $\mathbf{P}_0$ to the plane, the vector $\mathbf{w}$ has to be projected onto the normal vector $\mathbf{v}$. The resulting absolute distance d_p is given as

$$d_p = \frac{|\mathbf{v} \cdot \mathbf{w}|}{|\mathbf{v}|} \tag{4.7}$$

$$= \frac{|ax_0 + by_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}} . \tag{4.8}$$

This can be expressed in a more convenient way using the plane representation in Hessian normal form as

$$d_p = \hat{\mathbf{n}} \cdot \mathbf{P}_0 + p , \tag{4.9}$$

where $\hat{\mathbf{n}} = \frac{\mathbf{v}}{|\mathbf{v}|}$ is the unit normal vector of the plane and $p = \frac{d}{\sqrt{a^2+b^2+c^2}}$.

The expression from equation 4.9 represents the error measure used for the evaluation of all 3D-points $\mathbf{P}$. In order to achieve a stable estimate of the plane parameters in equation 4.4, the influence of outliers has to be mitigated using a robust estimation approach. To this purpose, the RANSAC-method from section 2.2.4 is used in a different configuration. The latter estimates an initial model (*i.e.* the parameter set) from 3 randomly selected points and evaluates the consent of the entire dataset to this model. If a sufficiently high number of inliers exist, the resulting parameters are used as starting values for a refined least-squares estimate of the plane.

Least-squares estimation of a rotating rigid-body model The regression plane estimation from the previous chapter enables the direct assessment of a set of spatial points under the assumption, that the observed object surfaces are planar. For the evaluation of the region-to-region matching method from section 4.3.1, this is largely sufficient. For the novel trajectory-to-trajectory method from section 4.3.2 however, the accuracy of the resulting spatial trajectories is only insufficiently captured. Thus, an appropriate method is presented in the following.

To this purpose, the measurement setup described in section 2.10 has been modeled mathematically. A sketch of the model is shown in figure 4.6. It is assumed that the bottom-camera represents the origin of the world coordinate system. Further, the turntable (represented by a gray disc) is assigned a local coordinate system of its own, whose $\mathbf{y}'$-axis is congruent with the rotation axis $\mathbf{r}$. The table is rotated by a fixed angle of $\phi = 5°$ between two frames. However, the $\mathbf{x}'$- and $\mathbf{z}'$-axes remain fixed.

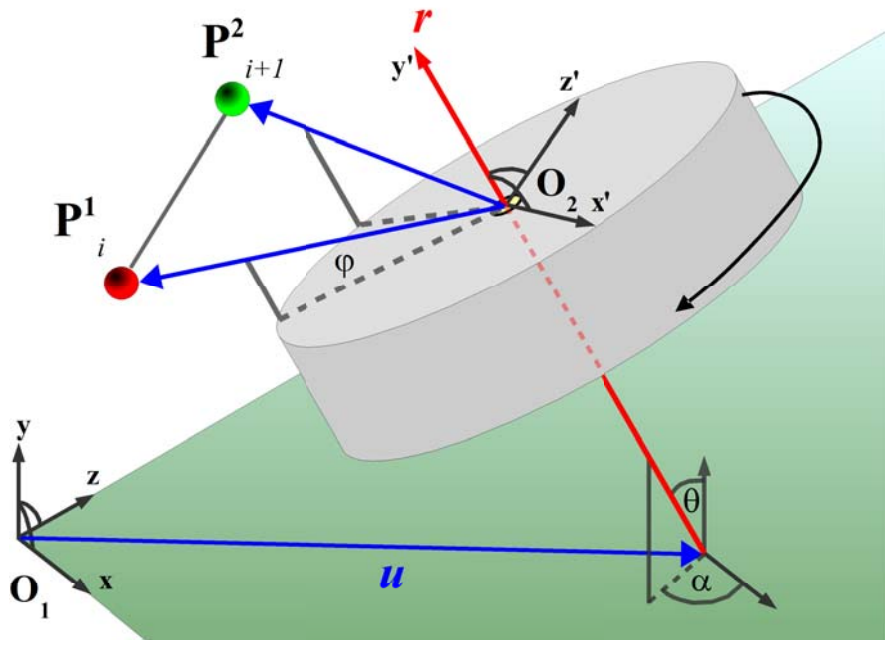


Figure 4.6: Schematic model of the measurement setup shown in section 2.3. The turntable (gray disc) rotates around an axis d (red) with a fixed angle of $\phi = 5°$ between two frames. A spatial point $\mathbf{P}_i^1$ (red ball) is thus rotated onto a new position $\mathbf{P}_{i+1}^2$ (green ball). The origin of the bottom camera $\mathbf{O}_1$ is assumed to be the center of the world coordinate system.

Given ϕ, a 3D-point $\mathbf{P}^1_{i,T}$ in turn-table coordinates is rotated from time index i into $i+1$ according to

$$\mathbf{P}^1_{i+1,T}{}' = \mathbf{R}_\phi \mathbf{P}^1_{i,T} \ , \tag{4.10}$$

where the rotation matrix $\mathbf{R}_\phi$ is defined as

$$\mathbf{R}_\phi = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} . \tag{4.11}$$

As can be seen from the matrix structure, the rotation is performed around the $\mathbf{y}'$-axis of the local turn-table coordinate system. For a point correspondence of the form $\mathbf{P}^2_{i+1} \leftrightarrow \mathbf{P}^1_i$ as supplied by the track matching algorithm from section 4.3.2, the relation $\mathbf{P}^1_{i+1,T}{}' = \mathbf{P}^2_{i+1,T}$ must hold in the ideal case. In order to verify this relation, a transformation of both points into the turn-table coordinate system has to be performed. The necessary equations are derived in the following.

As shown in figure 4.6, a point $\mathbf{P}^1_i$ given in world coordinates can be expressed in turn-table coordinates by following a vector $\mathbf{u}$ from the origin of the world coordinate system to the intersection point of the turn-table rotation axis $\mathbf{r}$ and the $\mathbf{xz}$-plane. From there, the origin of the turn-table coordinate system can be reached by following $\mathbf{r}$ for a scalar distance λ. In order to facilitate the estimation process, λ is set to zero. Note that for clarity of demonstration, this setting has not been adapted in figure 4.6. Once the turn-table origin is reached, both coordinate systems have to be aligned to each other with regard to the direction of the three axes. This is achieved by means of a rotation matrix $\mathbf{R}_{WT}$. Assuming that $\mathbf{x} - \mathbf{x}'$ and $\mathbf{z} - \mathbf{z}'$ are aligned already, the transformation of the $\mathbf{y}$-axis onto the $\mathbf{y}'$-axis is given by the matrix

$$\mathbf{R}_{WT} = \mathbf{I} + 2\mathbf{y}'\mathbf{y} - \frac{1}{\mathbf{y}\cdot\mathbf{y}'}(\mathbf{y}+\mathbf{y}')(\mathbf{y}+\mathbf{y}')^T \ , \tag{4.12}$$

where $\mathbf{I}$ is a 3×3 unity matrix [Luh00]. The assumption $\mathbf{y} = [0,1,0]^T$ further facilitates equation 4.12 to

$$\mathbf{R}_{WT} = \begin{bmatrix} 1 - \frac{1}{1+y_{y'}} x^2_{y'} & x_{y'} & -\frac{1}{1+y_{y'}} x_{y'} y_{y'} \\ -x_{y'} & y_{y'} & -z_{y'} \\ -\frac{1}{1+y_{y'}} x_{y'} z_{y'} & z_{y'} & 1 - \frac{1}{1+y_{y'}} z^2_{y'} \end{bmatrix} , \tag{4.13}$$

where $[x_{y'}\ y_{y'}\ z_{y'}]^T$ are the coordinate components of the $\mathbf{y}'$-axis. Given the aligned coordinate axes, the point $\mathbf{P}^1_i$ can be reached by simply following $\mathbf{P}^1_{i+1,T}{}'$ as illustrated in figure 4.6.

The model parameters to be estimated are the x- and z-component of the intersection between rotation axis $\mathbf{r}$ and the ground plane $[u_x\ 0\ u_z]^T$, and the components of $\mathbf{r}$,

which are expressed as

$$\mathbf{r} = \begin{bmatrix} \cos(\theta)\sin(\alpha) \\ \cos(\alpha) \\ \sin(\theta)\sin(\alpha) \end{bmatrix} . \tag{4.14}$$

The rotation axis has been expressed in spherical coordinates, so that the remaining parameters to be estimated are α and θ.

For a pair of points $\mathbf{P}_{i+1}^2 \leftrightarrow \mathbf{P}_i^1$, the euclidean distance d_r between both is expressed as:

$$d_r = \left\| \mathbf{u} + \mathbf{R}_{WT}\mathbf{R}_\phi\mathbf{R}_{WT}^{-1}\left[\mathbf{P}_i^1 - \mathbf{u}\right] - \mathbf{P}_{i+1}^2 \right\| . \tag{4.15}$$

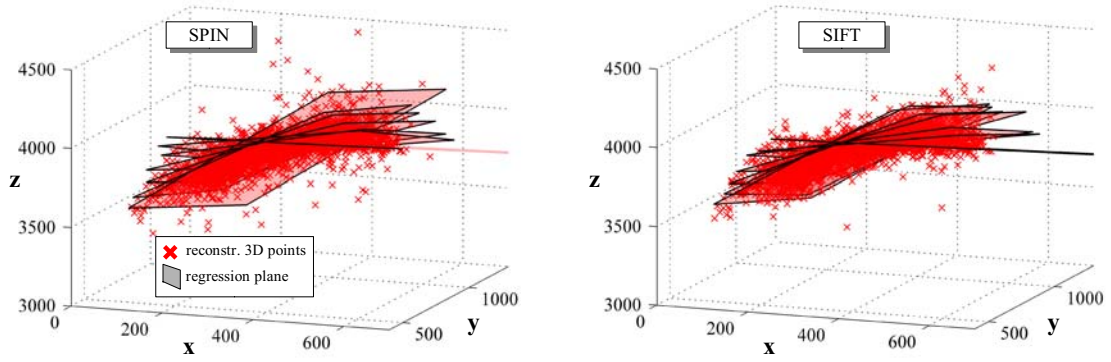
Finally, given a set of spatial trajectories, the model parameters $[u_x,\ u_y,\ \alpha,\ \theta]$ are found from all point correspondences using a least-squares estimation with equation 4.15 as the model function. For an ideal point cloud free of errors, d_r should be zero with every pair of 3D-points.

4.4.2 Single-Frame Stereo Matching

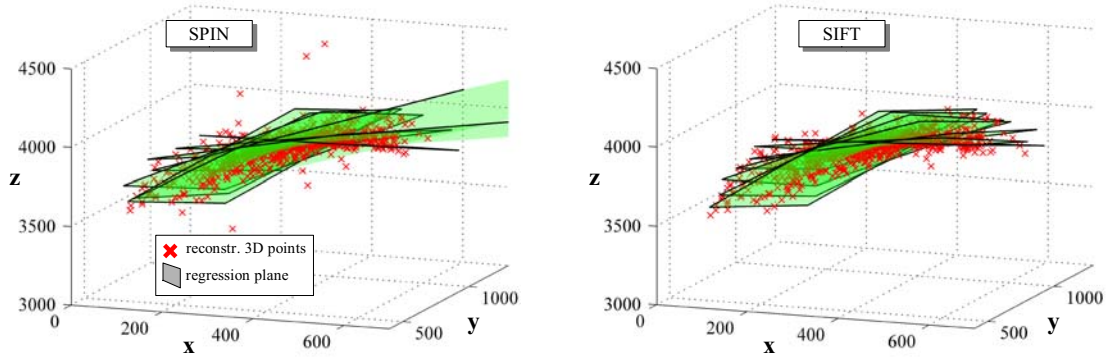
Regression plane estimation. In the following, the single-frame stereo matching algorithm from section 4.3.1 is evaluated using regression plane estimation. Figures 4.7 and 4.8 show the results for the five affine-covariant detectors. At each time step, regions from the bottom and top camera of the measurement setup have been matched based on the distance of the associated descriptors. The left columns show the results for SPIN-based matching while in the right columns SIFT-descriptors have been used. As described in section 4.3.1, the Hungarian method has been used for resolving assignment ambiguities such that the resulting correspondences are unique. A regression plane has been estimated from the 3D-points using a robust RANSAC-based estimation, followed by a least-squares refinement. For all detectors except MSER, the SPIN-based reconstruction contains a significantly higher number of outliers than with SIFT. This coincides with the previous chapters, where SPIN-based trajectories were also inferior with regard to the percentage of outlier correspondences and localization accuracy. With SIFT-based matching, the reconstructed points lie much closer to the estimated regression planes. However, the figures only permit a purely qualitative assessment.

With regard to the number of reconstructed points, the HESAFF-detector is clearly superior, followed with distance by the EBR-detector. Notably, the remaining detectors show similar numbers of reconstructed regions. According to the results from the previous chapter, a threshold of $d_{d,max} = 0.3$ on the maximally permissible descriptor distance has been chosen for SIFT-based reconstruction. For SPIN-descriptors, the setting $d_{d,max} = 0.2$ was used instead. Both settings lead to a comparable number of correspondences, as can be seen in table 4.1. There, the number of reconstructed 3D-points is shown for all detectors and both SIFT and SPIN.

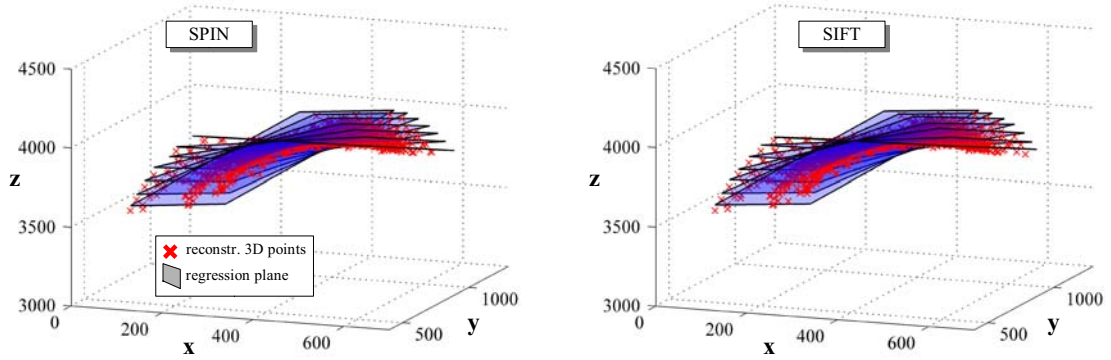
In figure 4.9, the orthogonal distance of all points from the respective regression plane d_p is plotted against the n-percentile. The diagram may be interpreted as follows: For



4.7.1: EBR

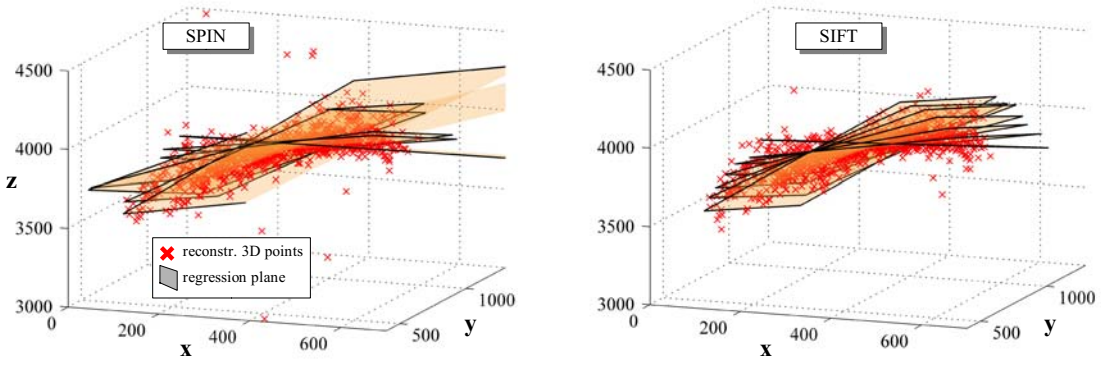


4.7.2: IBR

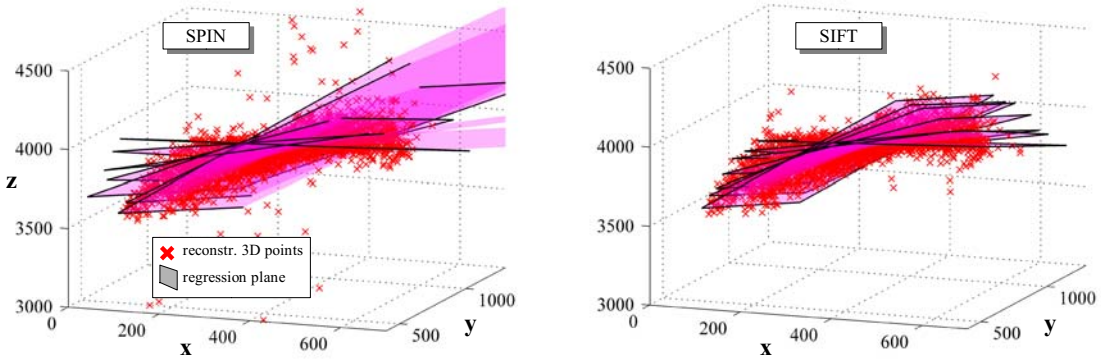


4.7.3: MSER

Figure 4.7: Single-frame stereo matching: Reconstructed 3D-points, regression planes for each frame and estimated rotation axis. For results of HARAFF and HESAFF see figure 4.8.



4.8.1: HARAFF



4.8.2: HESAFF

Figure 4.8: Single-frame stereo matching: Reconstructed 3D-points, regression planes for each frame and estimated rotation axis. For results of EBR, IBR and MSER see figure 4.7.

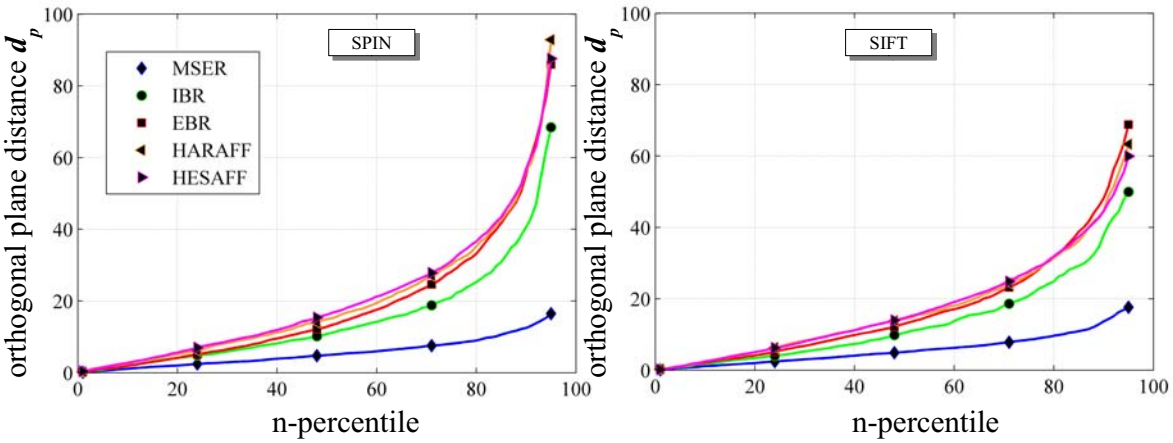


Figure 4.9: Single-frame stereo matching: Distribution of the regression plane residual error d_p over the n-percentile.

Table 4.1: Number of reconstructed 3D-points for all five affine-covariant detectors using single-frame matching with SIFT (top row) and SPIN (bottom row) descriptors.

	MSER	IBR	EBR	HARAFF	HESAFF
SIFT: $\sum$ 3D-points	2333	2633	5669	2397	8833
SPIN: $\sum$ 3D-points	2044	2570	5802	2185	9261

a given error d_p, the percentage of points with a lower error may be found by intersecting the respective detector curve. With IBR and SPIN-based matching for example, approximately $70\ \%$ of all points have an error of $d_p \leq 20\ pel$. For SPIN-based points, the regression error is significantly higher than for SIFT-based points. This result coincides with the evaluation of monocular trajectories from the previous chapter, where SPIN-based trajectories were generally inferior with regard to accuracy. Thus, although the computational complexity of SPIN is clearly smaller than for SIFT-descriptors, the use of the former for stereo reconstruction leads to a significantly higher error rate and should be used only in applications, where processing power is limited and accuracy is of secondary importance.

Notably, there exist only minor differences between HARAFF-, HESAFF- and EBR-based points. For all three methods, the amount of points with an error of $d_p \leq 20\ pel$ is at $60\ \%$. The performance of IBR is slightly better for both SIFT- and SPIN-descriptors. Among all five methods, MSER achieves the best performance, with $80\ \%$ of all points below $d_p = 10\ pel$. Notably, there is no significant difference between both descriptors to be seen. This observation again coincides with the evaluation results from the previous chapters, where a more detailled discussion on this topic may be found.

4.4.3 Multi-Frame Graph-based Stereo Matching

Regression plane estimation. In the following, the multi-frame stereo matching algorithm from section 4.3.2 is evaluated using regression plane estimation. Figures 4.10 and 4.11 show the results for the five affine-covariant detectors. As in the previous section, regions from the bottom and top camera of the measurement setup have been matched based on the distance of the associated descriptors. Again, the left columns show the results for SPIN-based matching while in the right columns SIFT-descriptors have been used. There are two major differences to single-frame matching to be noted. Firstly, the number of reconstructed points is significantly lower for all five detectors. This observation can be explained by the smaller number of two-dimensional regions which are available for matching, as shown in figure 3.15 in section 3.4.2. Table 4.2 additionally shows the number of successfully reconstructed regions for all five detectors, which allows for an additional quantitative comparison. Secondly, the amount of

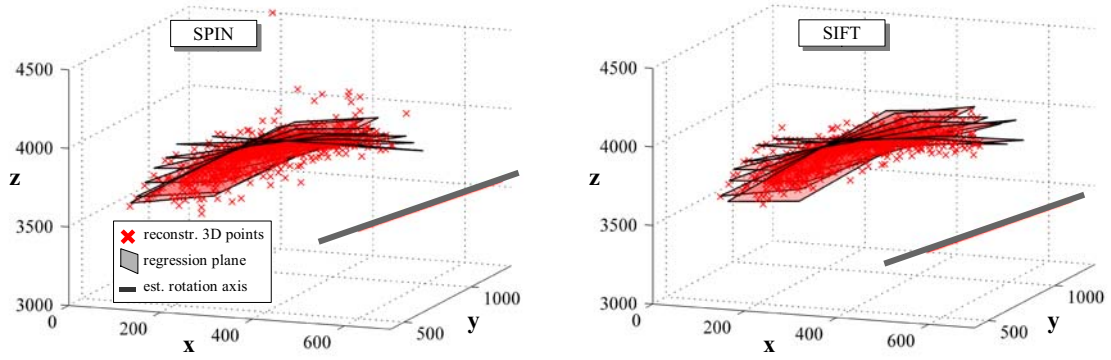
reconstruction outliers is much smaller than with single-frame reconstruction. This observation is especially evident for the HESAFF-detector, where the spread around the regression plane is significantly lower. Also, there exist almost no differences between SIFT- and SPIN-based matching. Thus, contrary to conventional single-frame reconstruction, the new method permits the use of the less-distinctive but also less expensive SPIN-descriptor. This proves beneficial in applications, where the available hardware is limited.

Table 4.2: Number of reconstructed 3D-points for all five affine-covariant detectors using multi-frame matching with SIFT (top row) and SPIN (bottom row) descriptors.

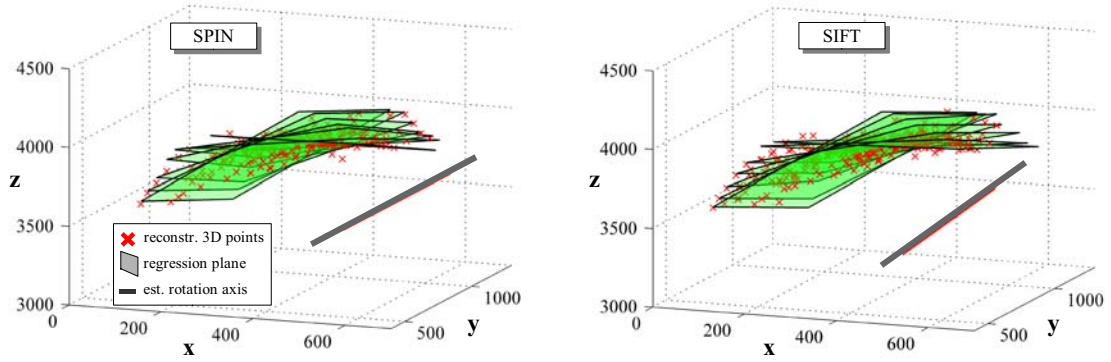
	MSER	IBR	EBR	HARAFF	HESAFF
SIFT: $\sum$ 3D-points	1830	778	1506	242	2238
SPIN: $\sum$ 3D-points	1482	640	1422	156	2290

Additionally, figure 4.12 shows the distribution of residuals for all five detectors and both descriptors over the n-percentile. As indicated by the qualitative distribution of reconstructed points in figures 4.10 and 4.11, the residual errors for multi-frame matching are significantly lower than with single-frame matching. Although SIFT-based points still perform better, the distance to SPIN is clearly smaller. Compared to figure 4.9, the relative ordering of the detectors is still identical. While MSER is again superior with regard to reconstruction accuracy, the HESAFF-detector shows the worst performance among all five methods.

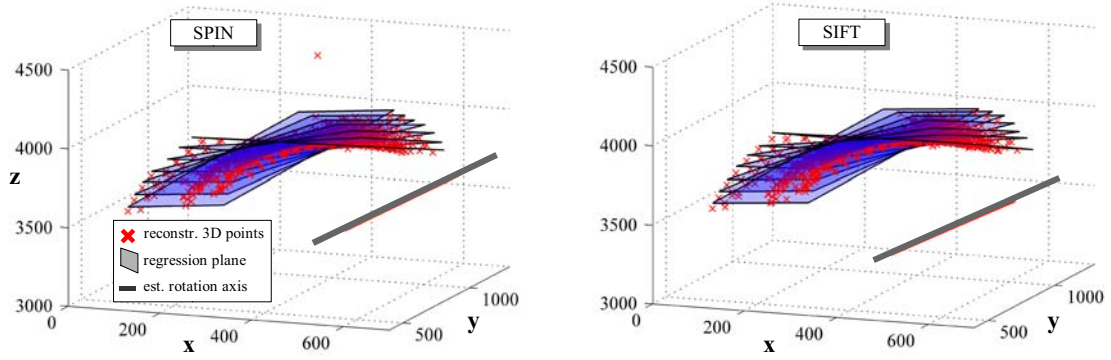
Rigid-body model estimation. In addition to single-frame matching, the proposed multi-frame method also captures temporal relations between the reconstructed points in adjacent frames. Thus, not only a spatial point cloud but a set of trajectories results. This additional information has not yet been considered in the previous evaluations. To this purpose, the previously described turn-table rotation model is used in a least-squares estimation. However, due to the nature of the model, results can only be obtained for multi-frame and not for single-frame matching. The estimated parameters represent the turn-table rotation axis $\mathbf{r}$, which has also been visualized in figures 4.10 and 4.11. In figure 4.13, the distribution of the error residuals d_r according to equation 4.15 over the n-percentiles can be seen. Most notably, there exist almost no differences between SIFT- and SPIN-based reconstruction. Except for the HESAFF-detector, the residual errors using SPIN-descriptors are higher. The relative ordering of the five detectors is widely identical to the regression plane estimation in figure 4.12: While MSER shows the lowest error among all detectors with $60\ \%$ of all points below $d_r \leq 5\ pel$, HESAFF exhibits the highest errors with $40\ \%$ of all points above $d_r \leq 8\ pel$ for both SIFT- and SPIN-based points.



4.10.1: EBR

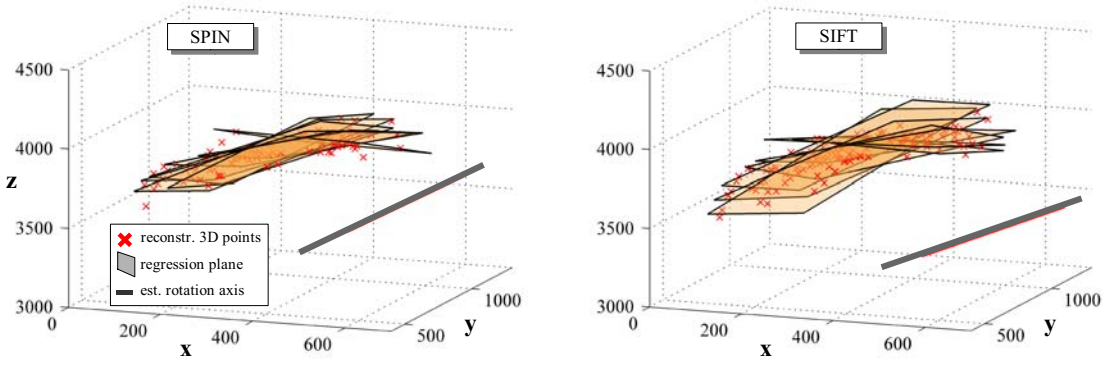


4.10.2: IBR

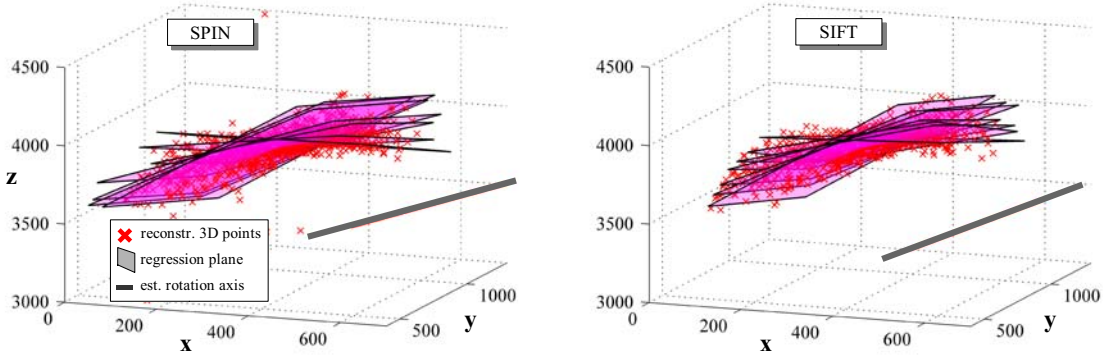


4.10.3: MSER

Figure 4.10: Multi-frame stereo matching: Reconstructed 3D-points, regression planes for each frame and estimated rotation axis. For results of HARAFF and HESAFF see figure 4.11.



4.11.1: HARAFF



4.11.2: HESAFF

Figure 4.11: Multi-frame stereo matching: Reconstructed 3D-points, regression planes for each frame and estimated rotation axis. For results of EBR, IBR and MSER see figure 4.10.

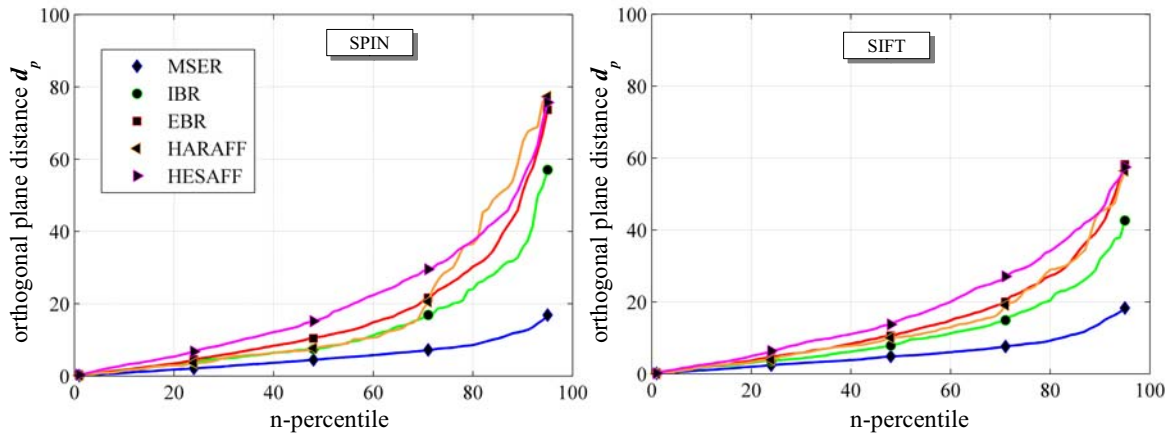


Figure 4.12: Multi-frame stereo matching: Distribution of the regression plane residual error d_p over the n-percentile.

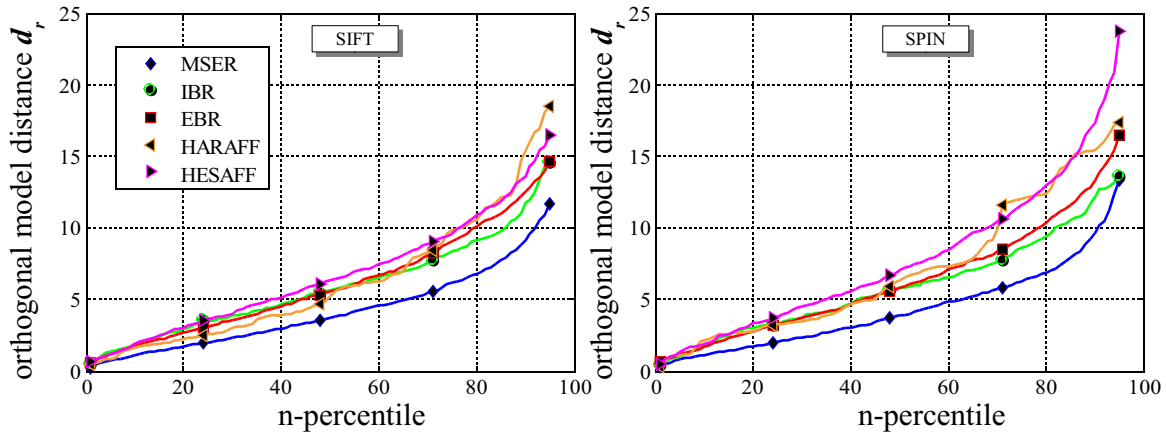
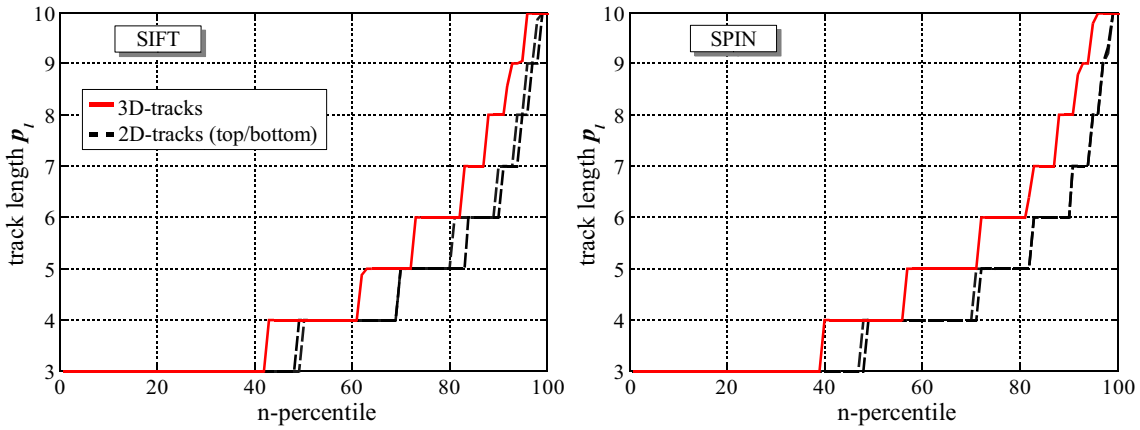
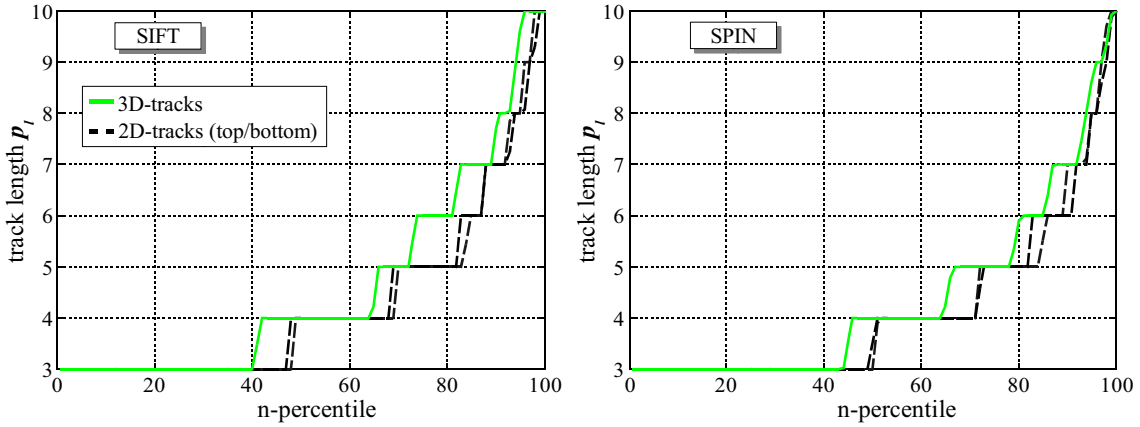


Figure 4.13: Multi-frame stereo matching: Distribution of the turn-table model residual error d_r over the n-percentile.

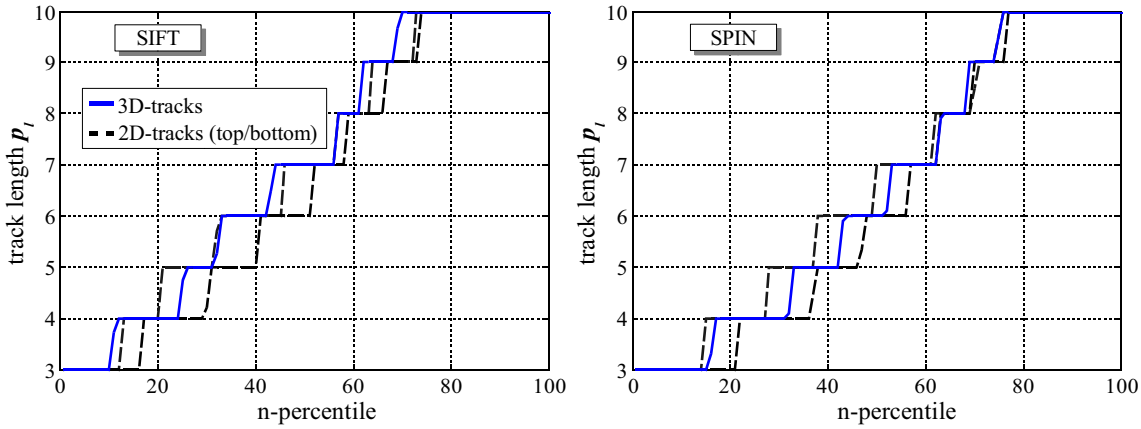
Trajectory lengths. One of the major advantages of the proposed multi-frame tracking method is its ability to link two spatial points on the basis of only a single monocular trajectory. As elaborated in section 4.3.2, a broken link in the first camera may easily be compensated for by an existing link in the second camera. In this section, the distribution of trajectory length has been evaluated, for both sets of monocular trajectories and for the reconstructed spatial trajectories. Figures 4.14 and 4.15 show the distribution of trajectory lengths p_l over the n-percentile. The left columns show the SIFT-based results while the right columns contain the SPIN-based distributions. With almost every detector-descriptor combination, the spatial trajectories exhibit a greater length than both monocular sets. Especially with EBR-regions, this advantage is most evident. While for monocular tracking 70 % of all trajectories contain less than 5 elements, the amount of spatial trajectories of the same length is at only 60 % for SIFT and at 55 % for SPIN. Only for HARAFF-based tracking, the length of spatial trajectories is inferior to monocular tracking. As already indicated by the low number of reconstructed features from tables 4.1 and 4.2, this result confirms that the use of this particular detector for spatial tracking can not be recommended.



4.14.1: EBR

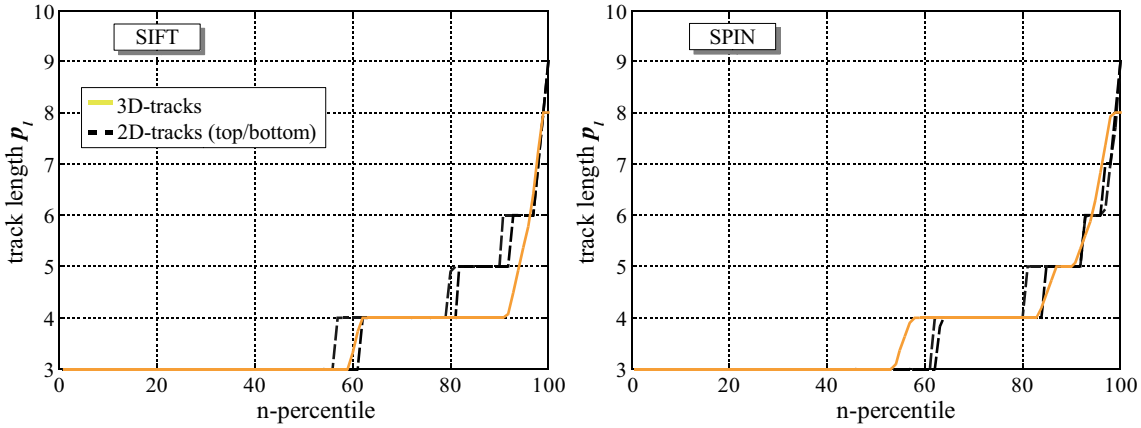


4.14.2: IBR

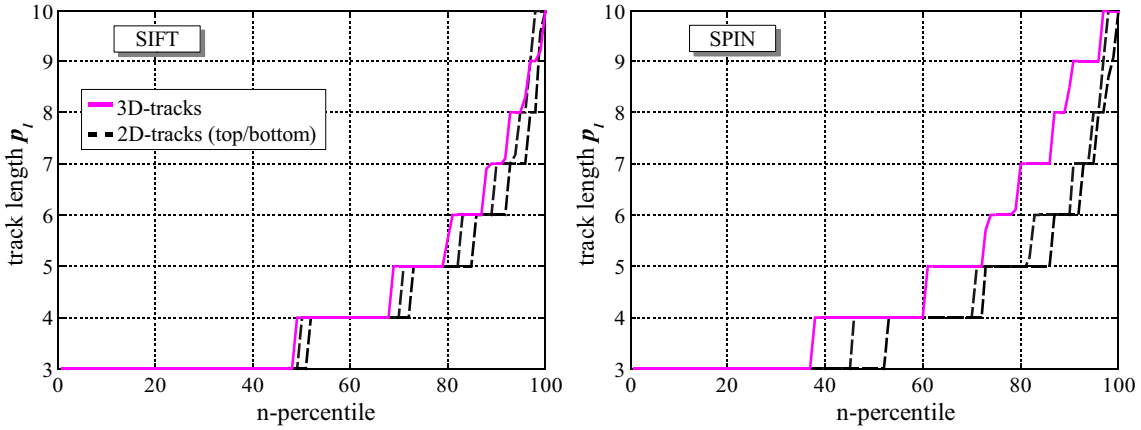


4.14.3: MSER

Figure 4.14: Distribution of 3D-track lengths (solid lines) and 2D-track lengths (dashed lines) for both SIFT and SPIN. For results of HARAFF and HESAFF see figure 4.15.



4.15.1: HARAFF



4.15.2: HESAFF

Figure 4.15: Distribution of 3D-track lengths (solid lines) and 2D-track lengths (dashed lines) for both SIFT and SPIN. For results of EBR, IBR and MSER see figure 4.14.

4.5 Chapter Conclusion

Within this chapter, region tracking has been extended from monocular to binocular image sequences. One of the major contributions is a novel method for the reconstruction of spatial object motion, which takes as input two sets of two-dimensional trajectories from both the first and second camera of the measurement setup. Based on the latter, a correspondence analysis is performed such that the most similar trajectories between both camera views are assigned to each other. The resulting matches are three-dimensional and thus represent the spatial motion of the associated objects. Based on the concept of graph-based tracking from chapter 3, the new method performs a search for region correspondences between both cameras based on the epipolar geometry and considers each potential match within a weighted and directed graph.

Depending on the density of regions and on the distinctiveness of the descriptors, one region in the first camera might have several matches in the second camera. Instead of considering only a single frame pair for reconstruction (as with conventional stereo matching), the new method incorporates information from all frames of the sequence into the matching procedure. Ambiguous correspondences are reliably resolved by means of shortest-path search using Dijkstra's algorithm. Additionally, the proposed technique provides spatial trajectories instead of mere points and thus captures the motion of the observed objects. To the best knowledge of the author, a similar method has not yet been proposed in the literature.

In this chapter, the new graph-based method has been compared to a conventional technique from the literature. The latter performs the search for corresponding regions between both cameras for each time step and resolves potential ambiguities by means of combinatorial optimization. In order to compare the performance of both methods, a concept based on the estimation of a regression plane from the set of reconstructed points has been used. It has been shown, that the new technique is superior with regard to the accuracy of reconstruction and to the occurrence of model outliers. Although the number of points has shown lower for graph-based tracking, the gain in accuracy toward conventional single-frame matching is significant for all five affine-covariant detectors. Among those, MSER-based points achieved the highest accuracy. With regard to the sheer number of points, HESAFF was clearly superior.

With single-frame matching, the use of SPIN-descriptors for region matching led to a high number of outliers and a poor overall accuracy of the reconstructed 3D-points. The use of SIFT-descriptors showed an improved accuracy, albeit at the expense of an increased computational complexity. With graph-based multi-frame matching however, the use of SPIN-descriptors showed similar results as with SIFT. Although the performance of the latter was still slightly superior, the reduced computational effort of using the SPIN-method instead proves an attractive alternative for applications, where hardware restrictions prohibit the use of complex and expensive algorithms.

Also, it could be shown that the resulting spatial trajectories exhibit a greater length

than the monocular region trajectories from both cameras. This benefit arises from the ability of the graph-based method to establish links between two spatial features on the basis of one camera alone. A broken link in the first camera may easily be compensated for by an existing link in the second camera. In application scenarios where occlusions occur frequently, the proposed method is still able to provide a set of stable spatial trajectories. This ability has proved especially beneficial for the EBR-detector.

Considering the five affine-covariant detectors, MSER provides the longest trajectories with the highest accuracy. This coincides with the evaluation results from the previous chapters, where MSER also showed superior to all other methods with regard to region accuracy. In applications, where a high number of trajectories is required, the HESAFF-detector should be used. Although with an inferior accuracy, it provides almost $25\ \%$ more trajectories than the MSER-detector. Alhough the HARAFF-detector provided satisfactory results with monocular region tracking, both the number of reconstructed spatial points and the length of the resulting trajectories proved clearly inferior to all other detectors. Concluding, the latter should not be used for the purpose of spatial motion analysis.

5 Conclusions

Major achievements at a glance. In this work, the task of spatial motion reconstruction from image sequences has been considered using a stereoscopic camera setup.

With regard to the first step in the algorithmic processing chain - feature detection - a selection of five well-known affine-covariant region detectors from the literature has been compared to each other with regard to a number of performance measures. The given results extend the available literature on the topic and facilitate the well-founded selection of an appropriate detector according to the requirements of specific target applications (with regard to the required localization accuracy, the number of regions or the robustness under perspective distortions).

Considering the second step - monocular region tracking - a novel graph-based tracking concept has been proposed and compared against three standard approaches from the literature (local-, Bayesian- and KLT-tracking). It has been shown, that the new method is clearly superior with regard to most of the investigated performance measures, especially tracking accuracy. Also, the above-mentioned region detectors have been additionally compared with regard to their suitability for monocular tracking.

In the third step, a novel concept for the spatial reconstruction of object motion - binocular tracking - has been proposed. Based on sets of monocular trajectories from each camera of the stereo setup, the new method uses a graph-based approach to find correspondences not between single features (as with the standard methods from the literature) but between entire trajectories instead. Thereby, the influence of locally ambiguous correspondences is mitigated significantly. The resulting spatial trajectories contain both the three-dimensional structure *and* the motion of the observed objects at the same time. It has been shown, that the reconstruction errors and the number of mismatches (and thus outliers) are significantly lower than with conventional stereo reconstruction.

Detailed content summary. In chapter 2, a set of five affine-covariant region detectors has been compared with regard to a number of performance measures. In the currently available literature, a thorough assessment of detector performance - especially region accuracy - does not exist. Therefore, the selected detectors have been closely scrutinized in this regard using a set of freely-available image sequences. Among the investigated performance measures rank the detector localization accuracy in terms of position and shape of the surrounding support area, the number and percentage of successfully matched regions and the dependency of region accuracy on intrinsic properties such as scale, shape and the region density within a local neighborhood. Based on these

results, potentially error-prone regions may be identified and removed from further processing on the basis of these properties alone. Thus, the overall accuracy and also the complexity of a subsequent matching step can be significantly improved. The results from chapter 2 extend the available literature on the topic and facilitate the selection of appropriate detectors for specific target applications (*e.g.* monocular feature tracking). Also, the proposed generic evaluation methodology allows for a seamless inclusion of future detectors and descriptors.

In chapter 3, the same selection of affine-covariant detectors has been compared with regard to monocular region tracking. To this purpose, three algorithms of increasing complexity have been discussed. While the first (and most simple) method searches for region-to-region correspondences within a circular gating area of constant size and decides on the pair with highest descriptor similarity (local tracking), the second method performs a model-based state prediction of the presumed feature location in the next frame (Bayesian-filtering approach) to narrow the search area and thus reduce the number of correspondence candidates. Both methods employ a combinatorial optimization scheme for the resolution of ambiguities among the candidates of the two most recent frames. A significant drawback of both methods is the resulting high number of tracking outliers (*i.e.* region correspondences with an overlap of $\leq 50\ \%$), especially in the case of repetitive image structures.

In order to compensate for this drawback, a novel method for an improved treatment of correspondence ambiguities has been proposed: Instead of deciding on unique correspondences for every frame pair (as with both the local approach and Bayesian filtering), the proposed algorithm keeps all candidate correspondences until the end of the sequence (or for a predefined number of frames) in so-called *track graphs*, which model the relations between regions based on descriptor distance and an extended motion smoothness model. Instead of prematurely selecting the locally most probable candidate, decisions on specific correspondences are postponed until sufficient evidence has been gathered that allows for an improved extraction of trajectories by means of weighted graph-traversal. It has been shown, that the new graph-based method is superior with regard to most of the investigated performance measures. Especially in cases where the available hardware is limited, the proposed method allows for the use of less distinctive descriptors at the price of a moderate performance decrease. With conventional methods from the literature, the latter was much more pronounced in most cases. Although the results from chapter 3 serve mainly as a prerequisite for three-dimensional reconstruction in chapter 4, they may well be used in a self-contained way for purely two-dimensional scenarios.

In chapter 4, motion analysis is extended from the monocular single-camera case to binocular tracking with two cameras. To this purpose, a novel technique has been proposed which takes as input two sets of monocular feature trajectories (one from each camera of the measurement setup) and performs a correspondence analysis such that the most similar trajectories between both camera views are assigned to each other. The

resulting three-dimensional matches and the temporal relations between them thus represent the spatial motion of the observed objects. Compared to conventional stereo reconstruction techniques from the literature, the new method is less susceptible to mismatches between the two camera views, especially with less distinctive region descriptors such as spin-images. Also, the overall accuracy of the reconstructed 3D-points has been significantly improved.

Within all major chapters, the entire selection of the five affine-covariant detectors has been compared against each other, mostly in conjunction with both histogram-based region descriptors. A close scrutiny has been performed with regard to properties such as localization accuracy, the number of detected regions or the lengths of the resulting trajectories. The detailed evaluation results allow for a well-founded selection of an appropriate combination of region detectors and descriptors for a given target application and thus represent a sensible and useful extension to the existing literature on the topic.

Outlook on future work. In future work, the *salient region detector* proposed in [KZB04] will be included into the evaluation. The method performs detection on the basis of the entropy within the support area around a salient location and has shown promising potential in a number of publications.

Within all experiments performed in this thesis, the author default implementations and parameters of the respective detectors and descriptors have been used in order to allow for a direct comparison to existing work on the topic. In future research, different sets of parameters and the resulting behavior in the context of the evaluation framework should be tried.

With regard to the proposed graph-based method for monocular region tracking, the full number of 10 frames per sequence has always been used for graph construction within this thesis. Especially in time-critical tracking scenarios, a smaller number of accumulated frames is often mandatory. Thus, it should be investigated in how far the performance of graph-based tracking is affected by this parameter.

Also, the now separate steps of monocular tracking and binocular reconstruction could be merged into a single unified algorithm, which constructs a spatial track-graph by means of simultaneous matching between temporally adjacent images within the same camera *and* between spatially displaced images from both cameras of the stereo setup.

Bibliography

[AB86] H. Asada and M. Brady. The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):2–14, 1986.

[AJN05] A. Agarwal, C.V. Jawahar, and P.J. Narayanan. A survey of planar homography estimation techniques. Technical report, IIT-Hyderabad, 2005.

[AMG+02] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S.A. Adelaide. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[ATRB95] A.P. Ashbrook, N.A. Thacker, P.I. Rockett, and C.I. Brown. Robust recognition of scaled shapes using pairwise geometric histograms. *British Conference on Machine Vision*, 2:503–512, 1995.

[Bal87] A.V. Balakrishnan. Kalman filtering theory. *Optimization Software, Inc. Communication And Control Systems*, 1987.

[Bar87] S.T. Barnard. Stereo matching by hierarchical, microcanonical annealing. *International Joint Conference on Artificial Intelligence*, pages 832–835, 1987.

[Bau00] A. Baumberg. Reliable feature matching across widely separated views. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:774–781, 2000.

[Bea78] P.R. Beaudet. Rotationally invariant image operators. *International Conference on Pattern Recognition*, pages 579–583, 1978.

[BFY98] M.J. Black, D.J. Fleet, and Y. Yacoob. A framework for modeling appearance change in image sequences. *International Conference on Computer Vision*, pages 660–667, 1998.

[BH92] R.G. Brown and P.Y.C. Hwang. Introduction to random signals and applied Kalman filtering. 1992.

[BL98] L. Bretzner and T. Lindeberg. Feature tracking with automatic selection of spatial scales. *Computer Vision and Image Understanding*, 71(3):385–392, 1998.

[BL03] M. Brown and D.G. Lowe. Recognising panoramas. *International Conference on Computer Vision*, 1(2):3, 2003.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[BNM98] S. Baker, S.K. Nayar, and H. Murase. Parametric feature detection. *International Journal of Computer Vision*, 27(1):27–50, 1998.

[Bou98] S. Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. *International Conference on Computer Vision*, pages 790–796, 1998.

[Bou99] J.Y. Bouguet. *Visual methods for three-dimensional modeling (PhD thesis).* PhD thesis, California Institute of Technology, 1999.

[Bou00] J.Y. Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm. *Intel Corporation, Microprocessor Research Labs*, 2000.

[Bro71] D.C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[BSFC90] Y. Bar-Shalom, T.E. Fortmann, and P.G. Cable. Tracking and Data Association. *The Journal of the Acoustical Society of America*, 87:918, 1990.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[Cox93] I.J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[CP84] J.L. Crowley and A.C. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2):156–170, 1984.

[CV99] D.P. Chetverikov and J.P. Verestóy. Feature point tracking for incomplete trajectories. *International Conference on Computer Vision*, 62(4):321–338, 1999.

[DB06] M. Donoser and H. Bischof. Efficient Maximally Stable Extremal Region (MSER) Tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:553–560, 2006.

[Dij59] E.W. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

[FA91] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[Fau93] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

[FB81] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FP02] D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[Fra05] A. Frank. On Kuhns Hungarian Method - A tribute from Hungary. *Naval Research Logistics*, 52(1):2–5, 2005.

[FTVG04] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. *European Conference on Computer Vision*, 1:40–54, 2004.

[G+46] D. Gabor et al. Theory of communication. 1946.

[Har94] R. Hartley. An algorithm for self calibration from several views. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 908–912, 1994.

[HB98] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[HJA08a] A. Haja, B. Jähne, and S. Abraham. A Comparison of Region Detectors for Tracking. *DAGM-Symposium*, 1:112–121, 2008.

[HJA08b] A. Haja, B. Jähne, and S. Abraham. Localization Accuracy of Region Detectors. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, 15:50, 1988.

[HS97] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, 1997.

[HVS90] R. Horaud, F. Veillon, and T. Skordas. Finding geometric and relational structures in an image. *European Conference on Computer Vision*, pages 374–384, 1990.

[Hwa89] V.S.S. Hwang. Tracking feature points in time-varying images using an opportunistic selection approach. *Pattern Recognition*, 22(3):247–256, 1989.

[HZ03] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[IB96] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. *European Conference on Computer Vision*, 1(2), 1996.

[IB98] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. *European Conference on Computer Vision*, 1:893–908, 1998.

[Jac96] O.L.R. Jacobs. *Introduction to Control Theory*. Oxford U. Press, 1996.

[Jäh05] B. Jähne. *Digital Image Processing*. Springer, 2005.

[JH99] A.E. Johnson and M. Hebert. Using spin-images for efficient object recognition in cluttered 3dscenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[Kuh55] H.W. Kuhn. The Hungarian Method for the Assignment Algorithm. *Naval Research Logistics Quarterly*, 1(1/2):83–97, 1955.

[KvD87] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375, 1987.

[KVD99] J.J. Koenderink and A.J. Van Doorn. The structure of locally orderless images. *International Journal of Computer Vision*, 31(2):159–168, 1999.

[KZB04] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. *European Conference on Computer Vision*, 500:228–241, 2004.

[LF97] Q.T. Luong and O.D. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *International Journal of Computer Vision*, 22(3):261–289, 1997.

[LG97] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-d shape cues from affine deformations of local 2-d brightness structure. *Image and Vision Computing*, 15(6):415–434, 1997.

[Lin98] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[LK81] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *DARPA Image Understanding Workshop*, pages 121–130, 1981.

[Low92] D.G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992.

[Low99] D.G. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision*, 2:1150–1157, 1999.

[Low04] D.G. Lowe. Distinctive image features from scale-invariant key-points. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LSP03] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[LT94] R.A. Lane and N.A. Thacker. Tutorial: Overview of Stereo Matching Research. *Tina Memo*, 1:1–10, 1994.

[Luh00] T. Luhmann. *Nahbereichsphotogrammetrie-Grundlagen, Methoden und Anwendungen*. Herbert Wichmann Verlag, Heidelberg, 2000.

[MA07] M. Mühlich and T. Aach. High accuracy feature detection for camera calibration: A multi-steerable approach. *Lecture Notes in Computer Science*, 4713:284, 2007.

[May82] P.S. Maybeck. *Stochastic models, estimation and control. Vol. 1*. Academic Press London, 1982.

[MCUP04] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[MF92] S.J. Maybank and O.D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[Mor77a] J.J. Mor. The levenberg-marquardt algorithm: Implementation and theory. *Lecture Notes in Mathematics*, 630:105–116, 1977.

[Mor77b] H.P. Moravec. Toward automatic visual obstacle avoidance. *International Joint Conference on Artificial Intelligence*, 584, 1977.

[Mor79] H.P. Moravec. Visual mapping by a robot rover. *International Joint Conference on Artificial Intelligence*, pages 598–600, 1979.

[MP07] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, 73(3):263–284, 2007.

[MS98] F. Mokhtarian and R. Suomela. Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1376–1381, 1998.

[MS01] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. *International Conference on Computer Vision*, 1:525–531, 2001.

[MS04] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[MTS+05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1):43–72, 2005.

[Neg98] S. Negahdaripour. Revised definition of optical flow: integration of radiometric and geometric cues for dynamic scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):961–979, 1998.

[Nei67] U. Neisser. *Cognitive psychology*. Appleton-Century-Crofts New York, 1967.

[NJ93] T. Netzsch and B. Jähne. Ein schnelles Verfahren zur Lösung des Stereokorrespondenzproblems bei der 3D-Particle Tracking Velocimetry. pages 43–50, 1993.

[NJW92] T. Netzsch, B. Jähne, and D. Wierzimok. Dreidimensionale Messung turbulenter Strömung mit Bildverarbeitung. pages 150–157, 1992.

[Pul05] G.W. Pulford. Taxonomy of multiple target tracking methods. *Radar, Sonar and Navigation*, 152(5):291–304, 2005.

[RDF92] L. Robert, R. Deriche, and O.D. Faugeras. Dense Depth Map Reconstruction Using Multiscale Regularization. *International Conference on Image Processing*, pages 123–127, 1992.

[Rei79] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

[Roh92] K. Rohr. Recognizing corners by fitting parametric models. *International Journal of Computer Vision*, 9(3):213–230, 1992.

[RTH96] C. Rasmussen, K. Toyama, and G.D. Hager. Tracking objects by color alone. *Workshop on Applications of Computer Vision*, 1996.

[Sed83] R. Sedgewick. Algorithms. 1983.

[SJ87] I.K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):56–73, 1987.

[SM97] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.

[SMB00] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[SP72] E.A. Smith and D.R. Phillips. Automated cloud tracking using precisely aligned digital ATS pictures. *IEEE Transactions on Computers*, 21:715–729, 1972.

[SS02] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision*, 47(1):7–42, 2002.

[ST94] J. Shi and C. Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[STVL04] S. Sarkka, T. Tamminen, A. Vehtari, and J. Lampinen. Probabilistic Methods in Multiple Target Tracking (technical report). Technical report, Laboratory of Computational Engineering, Helsinki University of Technology, 2004.

[SZ02] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or 'how do i organize my holiday snaps?'. *European Conference on Computer Vision*, 1:414–431, 2002.

[TH96] K. Toyama and G. Hager. Incremental focus of attention for robust visual tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 189–195, 1996.

[tHRFSV93] B.M. ter Haar Romeny, L. Florack, A.H. Salden, and M.A. Viergever. *Higher Order Differential Structure of Images*. Springer-Verlag London, UK, 1993.

[TK91] C. Tomasi and T. Kanade. Detection and tracking of point features (technical report). *School of Computer Science, Tech. Rep. CMU-CS-91–132*, 1991.

[TVG04] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[VGP05] J. Vermaak, S.J. Godsill, and P. Perez. Monte Carlo filtering for multitarget tracking and data association. *IEEE Transactions on Aerospace and Electronic Systems*, 41(1):309–332, 2005.

[Vin96] M. Vincze. Optimal Window Size for Visual Tracking for Uniform CCDs. *International Conference on Pattern Recognition*, 96:786, 1996.

[VK95] M. Vetterli and J. Kovačević. *Wavelets and sub-band coding*. Prentice Hall PTR Englewood Cliffs, NJ, 1995.

[VRB01] C.J. Veenman, M.J.T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54–72, 2001.

[WB01] G. Welch and G. Bishop. An Introduction to the Kalman Filter. *ACM SIGGRAPH 2001 Course Notes*, 2001.

[Wit08] D. Withopf. *Reliable Real-Time Vehicle Vehicle Detection and Tracking (PhD thesis)*. PhD thesis, Interdisciplinary Center for Scientific Computing, University of Heidelberg, 2008.

[Wol94] J.M. Wolfe. Guided search 2.0. A revised model of visual search. *Psychonomic bulletin & review*, 1(2):202–238, 1994.

[Wue04] H. Wuest. Dynamic tracking and data association in image sequences (diploma thesis). Master's thesis, Department of Mathematics and Computer Science,University of Mannheim, 2004.

[Zha97] Z. Zhang. Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.

[Zha99] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *International Conference on Computer Vision*, 1:666–673, 1999.

[ZW94] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. *European Conference on Computer Vision*, 1994.