Occlusion Handling in Scene Reconstruction from Video

Von der Fakultät für Elektrotechnik und Informatik

der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des akademischen Grades

Doktor-Ingenieur

(abgekürzt: Dr.-Ing.)

genehmigte Dissertation

von

Dipl.-Math. Kai Cordes

geb. am 17. Dezember 1973 in Nienburg

2013

Referent:	Prof. Dr.-Ing. Jörn Ostermann
Korreferent:	Prof. Dr.-Ing. Thorsten Thormählen
Prüfungsvorsitz:	Prof. Dr.-Ing. Markus Fiedler
Tag der Promotion:	09.08.2013

Vorwort

Die vorliegende Dissertation entstand während meiner wissenschaftlichen Tätigkeit am Institut für Informationsverarbeitung (TNT) der Leibniz Universität Hannover.

Mein besonderer Dank gilt Herrn Professor Dr.-Ing. Jörn Ostermann für die Betreuung der Arbeit und die Übernahme des Hauptreferats.

Herrn Professor Dr.-Ing. Thorsten Thormählen danke ich für die Übernahme des Koreferat der Arbeit.

Herrn Professor Dr.-Ing. Bodo Rosenhahn möchte ich für die anregenden Diskussionen danken, aus denen oft gute Ideen entstanden sind.

Herr Professor Dr.-Ing. Markus Fiedler sprang kurzfristig für die Übernahme des Prüfungsvorsitzes ein. Dafür herzlichen Dank.

Mein Dank gilt ebenfalls meinen Kollegen und einigen Studenten am Institut für Informationsverarbeitung (TNT). Für das Gelingen der Arbeit trugen insbesondere Herr Dipl. Math. Oliver Müller durch seine Mitarbeit an der Lokalisationsgenauigkeit des SIFT - Detektors sowie Herr Dipl. Math. Björn Scheuermann durch unsere Zusammenarbeit an der Bild- und Videosegmentierung bei.
Auch viele weitere hier nicht namentlich erwähnte Kollegen trugen ihren Teil zu dieser Arbeit und zur kreativen Athmosphäre am Arbeitsplatz bei. Das war eine wirklich schöne Zeit am TNT!

Nicht zuletzt möchte ich mich herzlich bei meinen Lieben Maike, Emma und Justus und bei meinen Eltern bedanken, ohne die das alles nicht möglich gewesen wäre.

Contents

Nomenclature

Symbol

a	scalar
$\mathbf{a}$	vector
$\underline{\mathbf{a}} = (a_x, a_y)^\top$	Euclidean vector im two-dimensional space
$\mathbf{a} = (a_x, a_y, 1)^\top$	homogeneous vector im two-dimensional projective space
$\underline{\mathbf{A}} = (A_x, A_y, A_z)^\top$	euclidean vector im three-dimensional space
$\mathbf{A} = (A_x, A_y, A_z, 1)^\top$	homogeneous vector in three-dimensional space
$\underset{\sim}{\mathbf{a}} = (a_x, a_y, a_i)^\top$	vector in scale space
$\mathtt{A}$	matrix
$\hat{a}$	estimated value
$\tilde{a}$	observation value
$\delta(\underset{\sim}{\mathbf{n}}) = (\delta(n_x), \delta(n_y), \delta(n_i))^\top$	fullpixel and fullscale coordinates of a feature $\underset{\sim}{\mathbf{n}}$
$\delta(\mathbf{n}) = (\delta(n_x), \delta(n_y))^\top$	fullpixel coordinates of a feature $\mathbf{n}$
$\varepsilon(\underset{\sim}{\mathbf{n}}) = (\varepsilon(n_x), \varepsilon(n_y), \varepsilon(n_i))^\top$	subpixel and subscale coordinates of a feature $\underset{\sim}{\mathbf{n}}$
$\varepsilon(\mathbf{n}) = (\varepsilon(n_x), \varepsilon(n_y))^\top$	subpixel coordinates of a feature $\mathbf{n}$
$(\delta_x, \delta_y) := (\delta(n_x), \delta(n_y))$	abbreviation for fullpixel coordinates
$(\varepsilon_x, \varepsilon_y) := (\varepsilon(n_x), \varepsilon(n_y))$	abbreviation for subpixel coordinates

Notation

$\mathbf{0}$	null vector
$\mathtt{A}$	camera matrix
a	element of the camera matrix
$\mathbf{a}$	vector consisting of the 12 elements of the camera matrix
$\mathbf{a}_1^\top, \mathbf{a}_2^\top, \mathbf{a}_3^\top$	row vectors of the camera matrix
$\alpha_{\mathtt{H}}$	larger eigenvalue of $\mathtt{H}$
$\beta_{\mathtt{H}}$	smaller eigenvalue of $\mathtt{H}$
B_{CRF}	maximal number of feature points per image
$\mathtt{B}$	intermediate value for F-matrix estimation
$\underline{\mathbf{C}} = (C_x, C_y, C_z)^\top$	center of projection
$\underline{\mathbf{c}} = (c_x, c_y)^\top$	principal point
$d(\cdot)$	Euclidean distance
$d(\cdot)_\Lambda$	Mahalanobis distance
$\det(\cdot)$	determinant of a matrix
d_{GUIDED}	distance between reprojected 3D object point and detected

	feature point for FTR
$d^{\max}_{\text{GUIDED}}$	maximal distance between reprojected 3D object point and detected feature point for FTR
$d_{closest}$	nearest neighbor
$d_{2closest}$	second nearest neighbor
D_{KLT}	threshold for the displacement distance between two iterations of the KLT tracker
$\lvert D(\underset{\sim}{\mathbf{n}})\rvert$	contrast of a SIFT feature $\underset{\sim}{\mathbf{n}}$
$D(\cdot)$	value in DoG pyramid
D_r	robust cost function for ICP calculation
κ	weighting parameter for scale computation in D_r
s	scale parameter in D_r
$\mathtt{E}$	unit matrix
$E[\cdot]$	expectation value
E_{KLT}	threshold for the intensity difference between two iterations of the KLT tracker
φ_i	regional costs in energy function $E(f)$
$\varphi_{i,j}$	boundary costs in energy function $E(f)$
γ	coefficient in Energy function $E(f)$ for weighting of regional and boundary costs
β	weighting coefficient in boundary cost calculation $\varphi_{i,j}$
$E(f)$	energy function consisting of regional and boundary costs
ε	distance in the camera image used in RANSAC
$\varepsilon_{\max}$	maximum distance in the camera image used in RANSAC
ε_{OBJ}	normalized distance between CAD object and object point
$\varepsilon_{\text{OBJ}}[mm]$	normalized distance between scene object and object point in [mm]
ε_{BA}	error induced by a 3D-2D correspondence in the bundle adjustment
$\varepsilon_{\text{RMSE}}$	reprojection error of a 3D-2D correspondence
i^{E}	layer localization error
σ^{E}	scale localization error
ξ^{E}	spatial localization error
$\eta_{\mathtt{A},\min}$	minimum number of correspondences between image k and $k-l$ for outlier elimination in FTR using the camera matrix
$\eta_{\mathtt{F},\min}$	minimum number of correspondences between image k and $k-l$ for outlier elimination in FTR using the F-matrix
$\mathtt{F}$	fundamental matrix
$\mathbf{f}$	vector consisting of the F-matrix elements
f	focal length
L_{MAX}	maximum number of past images considered for FTR

L_{MIN}	minimal length of feature trajectory for being considered for FTR
$\mathtt{G}$	intermediate value used for the feature selection
γ	intermediate value for F-matrix estimation
g_x, g_y	image signal gradients in x- and y-direction
H_a	image signal transfer function
ω_ρ	radial displacement of a lens
H_{ag}	Gaussian approximation of H_a
$\mathtt{H}$	hessian matrix with x and y components
$\underset{\sim}{\mathtt{H}}$	hessian matrix with x, y, and scale components
$\mathtt{I}$	intensity of the luminance signal
J	number of 3D object points
$\mathbf{P}$	3D object point
j	index of a 3D object point $\mathbf{P}_j$
J_{OBJ}	number of 3D object points located nearby the CAD model
$\mathscr{J}$	set of trajectories with a reconstructed object point $\mathbf{P}_j$
$\mathtt{K}$	calibration matrix
K	index of the current camera image
k	factor between scales of the DoG pyramid
κ_3	parameter of inverse radial distortion
K_{CRF}	constant for the calculation of the CRF
$\mathbf{l}$	epipolar line
$\Lambda_{\mathbf{p}}$	error covariance matrix of the feature points
$\underline{\mathbf{m}} = (x_m, y_m)$	image point in the coordinate system of a window
M_{CRF}	minimal distance between two feature points
$\underline{\mathbf{n}} = (n_x, n_y)^\top$	image point in the picture coordinate system
N	number of scale in each octave in SIFT
N_{RSC}	maximum number of iterations used in RANSAC
N_{GMM}	number of mixture components used for the GMM
N_{hist}	neighborhood used for the occlusion property verification
N_{occl}	diameter of an occlusion information disc
N_{SIFT}	size of a SIFT feature neighborhood used for descriptor calculation
N_x, N_y	number of elements in the picture memory
O_{KLT}	maximum number of iterations used by the KLT tracker
$mag(x,y)$	magnitude sample in the neighborhood of a SIFT feature
r_{Ori}	radius defining the neighborhood of a SIFT feature for orientation calculation
Θ	orientation of a SIFT feature
$\theta(x,y)$	orientation sample in the neighborhood of a SIFT feature
ϕ_x, ϕ_y, ϕ_z	rotation angles

$\mathbf{p}_{j,k}$	feature point, projection of the object point $\mathbf{P}_j$ into a camera image at time k
$\mathbf{p}_{j,k}^{\text{invisible}}$	image point resulting from projection of the in camera k not visible object point $\mathbf{P}_j$ into camera k
$\mathbf{p}_{j,k}^{\text{visible}}$	feature point assigned to object point $\mathbf{P}_j$ which is visible in camera k
r_{SIFT}	principal curvature of a feature point
$\mathtt{R}$	rotation matrix
ε_{DoG}	residuum of a DoG SIFT feature
$\varepsilon_{\text{SIFT}}$	residuum of a SIFT features
R_{RSC}	threshold for the relation of supported correspondences and number of correspondences used in RANSAC
L	scale space
S_{CRF}	threshold value for the calculation of the CRF
σ	standard deviation of Gaussian filter
σ_0	σ of first scale of each octave used in SIFT
Σ	covariance matrix of a Gaussian feature blob
σ_f	standard deviation of Gaussian feature blob
σ_{init}	initially assumed σ of input image
$\sigma_{\max}$	maximum standard deviation of the feature point location error
s_x, s_y	scale factors of the picture elements in x- and y-direction
$\underline{\mathbf{T}}$	intermediate value for the estimation of initial camera parameters
τ_{SIFT}	threshold parameter for SIFT second nearest neighbor matching
$\text{trace}(\cdot)$	trace of a matrix
$\mathbf{t}_j$	trajectory of object point $\mathbf{P}_j$
$\lvert\bar{\mathbf{t}}\rvert$	mean trajectory length
T	number of trajectories T with a non-consecutive correspondence
$\lvert\mathbf{t}_j\rvert$	length of a trajectory $\mathbf{t}_j$
U	index of camera image, in which the feature point is selected
Υ_{KLT}	number of resolution scales used by the KLT tracker
$\mathbf{p}$	vector of parameters used for optimization
X_c, Y_c, Z_c	axes of the camera coordinate system
X, Y, Z	axes of the world coordinate system
$(\cdot)^+$	Moore-Penrose pseudoinverse

Abbreviations

2D	two-dimensional

3D	three-dimensional
ALO	appearance learning from occlusions
CAD	computer aided design
CCD	charge-coupled device
CRF	cornerness response function
DoG	Difference of Gaussians
DoG SIFT	SIFT feature localization using the the image signal model
FTR	feature trajectory retrieval
GMM	Gaussian mixture model
GRIC	geometric robust information criterion
ICP	iterative closest point
KLT tracker	Kanade Lucas Tomasi tracker
MTF	modulation transfer function
RANSAC	random sample consensus
SAM	structure and motion
SIFT	scale invariant feature transform
SSD	sum of squared distances
SVD	singular value decomposition

Abstract

In this work, a scene reconstruction algorithm is extended with the handling of foreground occlusions. Scene reconstruction approaches are based on trajectories of image features in consecutive images of the input sequence. Foreground objects cause discontinued trajectories due to occlusion. If the occluded scene content reappears, redundant and therefore erroneous 3D object points are generated. By assigning features of reappearing scene content to the correct already reconstructed object points, these errors are avoided. Additionally, the knowledge of the occlusion of the background with foreground scene content is extracted. This knowledge enables the automatic segmentation of the video. The video segmentation eases the application of integrating virtual objects into the video sequence significantly. In particular, the automatic occlusion of integrated virtual objects with foreground scene content is presented. For this application, the accurate estimation of the camera parameters as well as the video segmentation is required.

The work is divided into the following parts. First, a reference method for *structure and motion* (SAM) estimation is introduced. This method is based on feature detection, correspondence analysis, outlier elimination and incremental bundle adjustment. Then, an accuracy analysis of the SIFT (*scale invariant feature transform*) key point localization is done. For the feature trajectory retrieval, the SIFT feature detection and correspondences analysis is required. The SAM reference method is extended with the following contributions.

- the improvement of the SIFT key point localization
- the extension of the structure and motion recovery algorithm with feature correspondences in non-consecutive frames
- the development of a new approach for the automatic foreground segmentation of the video

Finally, experiments show the increased accuracy of the reconstructed scene regarding two error measures: (1) the reprojection error, which is commonly used for accuracy evaluation and (2) the distance of the reconstructed point cloud to the known model of the observed scene.

The main application is the integration and automatic occlusion of virtual objects. This application is demonstrated with several natural image sequences. The integrated objects are occluded with foreground scene content by using the developed automatic video segmentation.

Keywords: scene reconstruction, feature detection, feature localization, SIFT, video segmentation, estimation of camera parameters, camera calibration, computer vision, bundle adjustment, integration of virtual object into natural image sequences

Kurzfassung

In dieser Arbeit wird ein Verfahren zur Szenenrekonstruktion um die Handhabung von Verdeckungen durch Vordergrund erweitert. Algorithmen zur Szenenrekonstruktion basieren auf Trajektorien von Bildmerkmalen in aufeinander folgenden Bildern der Eingangssequenz. Vordergrundobjekte sind Ursache dafür, dass Trajektorien unterbrochen werden. Wird der durch die Vordergrundobjekte zeitweise verdeckte Szeneninhalt wieder sichtbar, so werden möglicherweise neue, redundante und fehlerhafte 3D Objektpunkte generiert. Werden die Merkmale des wieder sichtbaren Szeneninhaltes zu den richtigen, bereits bestehenden Objektpunkten zugeordnet, so können diese Fehler vermieden werden. Zusätzlich kann Wissen über die Verdeckung des Hintergrundes durch Vordergrund gewonnen werden. Dieses Wissen kann für die automatische Segmentierung der Videosequenz genutzt werden. Durch die Segmentierung wird die Integration von virtuellen Objekten in die Bildsequenz erheblich vereinfacht, insbesondere dann, wenn die integrierten Objekte durch Vordergrund der natürlichen Szene verdeckt werden sollen. Für diese Anwendung ist sowohl die genaue Bestimmung der Kameraparameter als auch die Segmentierung des Vordergrundes erforderlich.

Zunächst wird ein Referenzverfahren zur Szenenrekonstruktion basierend auf Merkmalsdetektion, Korrespondenzanalyse, Ausreißerelimination und inkrementellem Bündelausgleich vorgestellt. Dann wird eine Genauigkeitsanalyse der Merkmalslokalisation des SIFT (*scale invariant feature transform*) Detektors durchgeführt. Für das Wiederfinden von Merkmalstrajektorien wird der SIFT Detektor und dessen Korrespondenzanalyse benötigt. Das Verfahren zur Szenenrekonstruktion wird wie folgt erweitert:

- Verbesserung der SIFT Merkmalslokalisation
- Erweiterung der Szenenrekonstruktion durch Merkmalskorrespondenzen in nicht aufeinander folgenden Bildern
- Entwicklung eines neuen Verfahrens zur automatischen Vordergrundsegmentierung einer Videosequenz

Schließlich wird die Verbesserung der Rekonstruktionsgenauigkeit mittels zweier Fehlermaße belegt: (1) anhand des Rückprojektionsfehlers, der üblicherweise zur Genauigkeitsevaluation verwendet wird und (2) anhand des Abstandes der rekonstruierten Punktwolke zu einem bekannten Modell der beobachteten Szene.

Die Hauptanwendung ist die Integration sowie die automatische Verdeckung von virtuellen Objekten. Diese Anwendung wird anhand von natürlichen Videosequenzen demonstriert. Die integrierten Objekte werden mit Hilfe der vorgestellten Videosegmentierung durch den Vordergrund der natürlichen Szene verdeckt.

Schlagwörter: Szenenrekonstruktion, Merkmalsdetektion, Merkmalslokalisation, SIFT, Videosegmentierung, Schätzung von Kameraparametern, Kamerakalibrierung, Rechnersehen, Bündelausgleich, Integration virtueller Objekte in natürliche Bildsequenzen

1 Introduction

The technique of *structure and motion* (SAM) recovery, also called *structure-from-motion*, estimates the rigid scene geometry from an image sequence. It is a well-established technique in computer vision with numerous fully automatic algorithms which have been developed over the last decades. In most approaches, the processing pipeline consists of feature detection, correspondence analysis, and scene reconstruction including outlier elimination and bundle adjustment.

In media production, various applications show the usability of these techniques for post production as well as for the live broadcast. In the former setup, movie editors integrate virtual objects into the image sequence which is captured by a video camera. In the latter setup, information content like customer specific commercials or scene explanations in sports events are integrated in the transmission with minimal delay. The same techniques are used in robot and car navigation for localization and mapping approaches, in which a subject uses a camera to localize itself in an unknown, to be discovered environment.

For each of these applications, the accuracy and the reliability of the scene reconstruction is of key interest. Thus, long and reliable feature trajectories consisting of automatically extracted feature points are desired. Their localization accuracy determines the accuracy of the scene reconstruction. Furthermore, it is crucial to provide a solution for situations in which scene content is temporarily not visible, e.g. when objects with a larger distance from the camera are occluded by foreground objects. If this situation is not covered, the reliability of the reconstruction suffers as too many feature tracks are lost and the scene is temporarily not represented by well distributed features. Even if the reconstruction is still reliable the estimation accuracy decreases due to a well known, undesired phenomenon in SAM recovery called drift. Drift results from small errors which accumulate with an increasing number of input images to a noticeable reconstruction error. A point which reappears after being invisible induces a redundant 3D object point. Both object points approximate one real 3D scene point. It is very likely that these points adopt different positions, which leads to small reconstruction errors.

After being occluded, scene content may be observed by the moving camera from a very different viewpoint. It follows that the surrounding texture information has changed significantly due to perspective distortion and lighting changes. Thus, the desired method for the correspondence analysis between non-consecutive frames has to be robust to these entities.

State of the Art

Feature Detection and Localization The usage of image features can be characterized by the reduction of the amount of data being processed for certain image measurements while preserving the same information content [16]. If the mapping from one image to a second is known, the feature localization accuracy can be derived from the distance between a mapped feature from the first image and the corresponding feature in the second image [70]. Inversely, as camera motion estimation algorithms are based on detected features in the images, the localization accuracy and the correspondence analysis are of key interest.

The Canny detector [16] uses an edge operator based on a gradient filter which uses the first derivative of a Gaussian function. Although the results are theoretically very good [50], the Canny detector looses precision because its localization is done using full pixel coordinates. Harris and Stephens [42] present a combined corner and edge detector resulting in consistent feature points as shown in [82, 83]. The Harris detector [42] results in full pixel coordinates, too. In following approaches, the localization accuracy is improved using coordinates that are determined with subpixel accuracy. Therefore, the image gradient signal surrounding a feature point is approximated using a function which should adopt its extremum at the feature position.

The interpolation of the image gradients with a parabolic function is examined by Rockett [77] for the Canny detector. Rockett shows that using the parabolic interpolation still leads to a systematic error and proposes a Gaussian approximation of the image signal for improvement. But, Rockett neglects to estimate the Gaussian parameters and uses a correcting look-up table instead. Mikulastik et al. derive the systematic error in the feature localization analytically and verify it experimentally [72, 73]. The magnitude of the systematic error depends on the subpixel position and is up to 0.025 pixel for a one dimensional feature. The systematic error is eliminated using a Gaussian approximation of the image signal surrounding a Canny edge [71]. The impact of this improvement on a SAM recovery approach and its reprojection error has not been examined so far. Hillman et al. demonstrate [47] that the accuracy of such algorithms in industrial movie production is very important. It is shown that reprojection errors of 1/4 pixel in high-resolution images are still visible and may disturb the observer. An increase in processing time for the computation is of subordinate importance as movie production companies have massive computation resources available [47].

Recent feature detection approaches use the scale space to detect interest points invariant to scale changes [6, 7, 58, 61, 66]. The scale space is defined by Lindeberg [55, 56, 57, 58]. It is built by cascading Gaussian filters of differing standard deviation σ, which is proven to provide the optimal filter kernel for this purpose [57]. The best known feature detector which uses the scale space for the detection is the SIFT (*scale invariant feature transform*) detector [61]. Numerous approaches are published to extend the scale invariance to affine invariance. The mapping between corresponding regions of two images can be modeled by an affine transformation [70], if perspective effects are

small on these local regions. The affine extension to SIFT uses several pyramids for modeling the affine scale space [101]. Based on affine normalization, the Harris-Affine [67] and Hessian-Affine [68] detectors determine the elliptical shape with the second moment matrix of the intensity gradients. In [64], maximally stable extremal regions (MSER) are constructed using a segmentation process. However, these approaches are not truly affine invariant since they do not select features in an affine scale space. In practice, the SIFT detector is the mostly used detection method for scale invariant features and provides enough stable features for moderate affine distortion of up to 40° [36].

In [55], a model for the features detected by a scale invariant feature detector, called feature blob, is developed. Its shape is a circular Gaussian function, which is used for studying the behavior of blob detectors. Brown and Lowe [13, 61] use a 3D quadratic function for the subpixel and subscale localization of SIFT features. Although its localization accuracy has not been analyzed so far, Brown and Lowe [13] explicitly stress the necessity of the subpixel estimation of SIFT keypoints, because these features are also detected in pyramid levels with low resolution. In these pyramid levels the subpixel localization is crucial to ensure sufficient accuracy for the features in the pyramid base level.

Since the signal gradients surrounding a feature do not have the shape of a 3D quadratic, the localization used in [13, 61] is suboptimal. A feature localization technique using a Gaussian approximation of the image signal for a scale invariant feature detector has not been developed so far.

Correspondence Analysis The correspondence analysis associates a feature of one image with the corresponding feature of a second image which shows the same scene. Corresponding feature points are projections of one 3D point of the observed scene into different camera images. For small baselines between the cameras, feature tracking methods like KLT [63] are appropriate to obtain stable and accurate correspondences. For larger viewpoint changes, feature matching methods have shown impressive performance in determining stable correspondences. The SIFT method presented by Lowe [61] establishes correspondences using distinctive characteristics of the feature neighborhood. These characteristics are usually assembled to a vector, called *descriptor*, which is used for the comparison. The SIFT descriptor is designed to be invariant to changes in brightness, rotation, and scale.

The enormous impact of the SIFT method induced many researchers to improve the descriptor construction in detail [14, 51, 69]. Nevertheless, using the SIFT descriptor is still the most widely used correspondence analysis method for wide baselines [38, 39, 60, 88, 102]. Its results are sufficient for the scene reconstruction scenario. Hence, it is used in this work.

Scene Reconstruction The scene reconstruction from an image sequence is a key technique in many computer vision applications [39, 44, 76, 88, 93, 100]. It consists of camera motion estimation and simultaneous reconstruction of the rigid scene geometry.

For the parametrization, a pinhole camera model with intrinsic and extrinsic camera parameters is used [32, 43]. The intrinsic parameters determine the camera mapping while the extrinsic parameters describe the localization and orientation of the camera in the world coordinate system. The basis for the estimation are corresponding features which arise from one 3D point being mapped to different camera image planes. By using a statistical error model which describes the errors in the position of the detected feature points, a maximum likelihood estimator can be formulated which simultaneously estimates the camera parameters and the 3D positions of feature points. This joint optimization is called bundle adjustment [96].

Most sequential approaches for structure and motion recovery determine corresponding features in consecutive frames. Thormählen [90] and Pollefeys et al. [76] determine newly appearing feature points using the Harris detector and correspondences between consecutive frames with the KLT Tracker [63]. For unordered images using a picture database such as Flicker, Snavely et al. [87, 88] and Frahm et al. [38, 39] use the SIFT descriptor to establish correspondences. This is essential for their application because the baseline between the cameras of two images is large in most cases. Outliers are usually removed from the scene estimation using the *random sample consensus* (RANSAC) [34] approach.

For the scene reconstruction from video sequences, it is crucial to provide a solution for situations in which scene content is temporarily invisible. Otherwise, noticeable drift occurs resulting from accumulated estimation errors. In recent literature, a few publications address this problem for different application scenarios. In case of a closed sequence, the drift can be reduced by enforcing the constraint between the cameras observing the same scene content (i.e. first and last camera view after a complete circuit) [31, 35, 60]. In [28], the drift is reduced by estimating the transformation between reconstructed 3D point clouds using RANSAC. In [93], broken trajectories caused by occlusion are merged using a combination of localization and similarity constraints of the reprojected object points in the images. These approaches are only applicable in a post processing step, i.e. after the last frame of the image sequence is processed. Furthermore, the estimation has to be accurate before applying the merging. Otherwise it would be impossible to match the cameras, point clouds, or the reprojections of object points. A recent approach [102] uses the SIFT descriptor for incorporating correspondences in non-consecutive frames into sequential structure and motion recovery. An additional homography constraint for planar features is included to stabilize the feature tracking using a two-pass matching.

The usage of an appropriate combination of feature matching and feature tracking for sequential scene estimation has not been developed so far.

Video Segmentation An important visual effect (VFX) used in movie production is the integration of virtual objects into the sequence. For the perspectively correct integration of the objects, the accurate computation of the camera parameters for each view is crucial. If the integrated object(s) should be occluded by the foreground of the nat-

ural scene, video segmentation, often called *matte creation* [47] is required. In movie production industry, the matte creation is still done manually [47]. To help the editor, semi-automatic algorithms are developed [8, 10, 78]. The user guides the algorithm by marking the foreground object(s) and the background with strokes [10, 80] or by a bounding box [78]. For subpixel accuracy at object boundaries, alpha mattes are incorporated [86]. The computation of alpha mattes is based on an initial segmentation with fullpixel accuracy.

Challenges

In standard structure and motion recovery approaches, a feature without a correspondence in the previous frame is regarded as a newly appearing object point. If an image feature has been temporarily occluded, the new object point and the object point that has been generated before the occlusion adopt different 3D positions. As a consequence, errors accumulate and drift occurs. This problem arises from foreground occlusion, moving objects, repeated texture, image noise, motion blur, or because tracked points temporarily leave the camera's field of view.

For the scene reconstruction, the length of the feature trajectories is of key interest. Long feature tracks are desired for the estimation. However, feature points of long tracks tend to loose accuracy which leads to a discontinuation. A second cause for discontinued feature tracks is the occlusion with foreground objects. If the features reappear, the construction of a new 3D object point should be avoided. The reappearing feature should be assigned to the previously discontinued trajectory and its 3D object point. As the point of view may have changed significantly after the discontinuation, a wide baseline correspondence analysis method as provided by SIFT is required. But, scale invariant feature detectors suffer from limited localization accuracy.

Successful scene reconstruction results are sensitive to the reconstruction accuracy and therefore sensitive to the localization accuracy of the feature points. A reprojection error of 1/4 pixel may already be disturbing to an observer [47]. Thus, the evaluation and improvement of the localization accuracy of the SIFT detector is needed.

The accuracy evaluation of scene reconstruction results with merged scene content is challenging. The commonly used reprojection error may not be an appropriate measure for the comparison of reconstructions with a differently constrained bundle adjustment. A more accurate solution with more constraints may have a higher reprojection error [28, 53], because it is more likely to find a reconstruction solution for a less constrained system of equations. While the accuracy of the reconstruction increases by enforcing more correct scene relations in the bundle adjustment, the reprojection error may increase because of the additional constraints.

Feature correspondences in non-consecutive frames provide information about the observed scene. If the non-consecutive correspondence is induced by foreground occlusion, cues for the position of foreground in the images can be extracted. These cues can be used for the automatic foreground segmentation of the video.

Solutions

The goal of this work is to solve the problem of temporarily not visible scene content. Hence, feature correspondences for non-consecutive images are incorporated into the sequential scene reconstruction approach. Due to a possibly significant change in illumination and perspective between the camera views on the disappearing and reappearing feature, a wide baseline correspondence analysis technique is used. The additional correspondences increase the reliability and accuracy of the reconstruction. The drawback of a wide baseline feature matching technique, such as SIFT, is the limited localization accuracy of the detected points. This is compensated by using a signal adapted localization technique which leads theoretically and experimentally to an increased localization accuracy of the features. Correspondences are established using feature tracking for consecutive and wide baseline feature matching for non-consecutive frames. The combined method increases the accuracy and reliability of the scene reconstruction.

The trajectories which contain a non-consecutive correspondence provide cues for the extraction of foreground regions in the images. These regions provide a reasonable initialization of the video segmentation algorithm. The segmentation is used together with the reconstructed scene for the application of integrating virtual objects into the video. Due to the highly accurate estimation of the camera parameters, the objects are integrated perspectively correct in each camera view. By combining the augmented sequence with the resulting video segmentation, the virtual objects are occluded automatically with the foreground objects of the real scene. Demonstrations show the usability of the presented approaches.

Overview

The fundamental models for scenes and the basics for scale invariant feature detection are explained in Chapter 2. A reference SAM recovery algorithm is introduced in Chapter 3. The localization accuracy of SIFT is analyzed in Section 4.1. The method for the increase of the localization accuracy is presented in Section 4.2 and in Section 4.3. The extension of the SAM recovery which incorporates non-consecutive correspondences is presented in Chapter 5. The extension includes the new approach of automatic video segmentation. Experimental results are presented in Chapter 6. In Chapter 7, the thesis is concluded.

2 Fundamentals

In this chapter, the necessary fundamental principles of the mathematics and state of the art approaches are introduced. The Sections 2.1, 2.2.1, 2.2.2, 2.2.4, and 2.3 are based on the work of Thormählen [90]. In the Sections 2.1- 2.3, the models for the mapping of a scene point to an image point is explained. In Section 2.4, the modeling of the scale invariant representation of an image which is based on the work of Lindeberg [57] is briefly reviewed.

2.1 Scene Model

The scene model describes the natural scene with a parametric model. This model determines positions and orientations of lighting, objects, and cameras in a fixed (X,Y,Z)-world coordinate system as shown in Figure 2.1. The scene model used in this work employs a camera model, an object model, and a simple lighting model. The lighting model

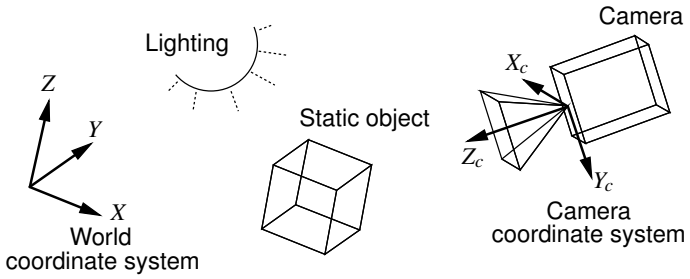


Figure 2.1: Scene model (from [90]).

assumes diffuse background lighting. Thus, every object surface in the scene receives the same level of illumination. This simple model does not consider the more complex scene properties, such as shadows and reflections.

The camera model describes the mapping of 3D objects in the world coordinate system into the camera plane. The position and orientation of the camera relative to the world coordinate system is determined by the (X_c,Y_c,Z_c)- camera coordinate system.

The object model assumes static object geometry. Additionally, static objects do not move relative to the world coordinate system.

2.2 Camera Model

To represent a real camera, a parametric camera model is used. It consists of a perspective mapping model, a lens model, an image signal model, and a sensor model as shown in Figure 2.2. While the parameters of the perspective mapping model and the image signal model are unknown, the parameters of the lens model and the sensor model are usually a priori known.

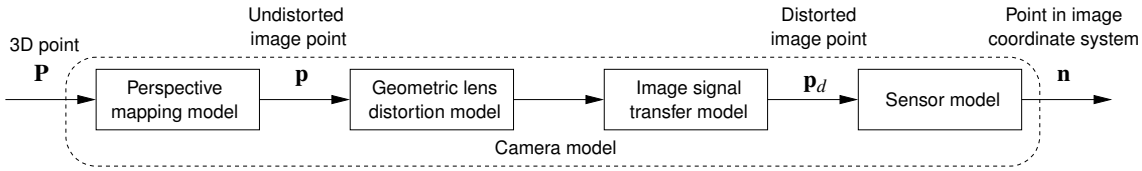


Figure 2.2: Camera model (extended from [90])

2.2.1 Perspective Mapping Model

The perspective mapping model describes the mapping of a 3D object point **P** to a 2D point **p** in the camera coordinate system. The focal length f is the distance between the

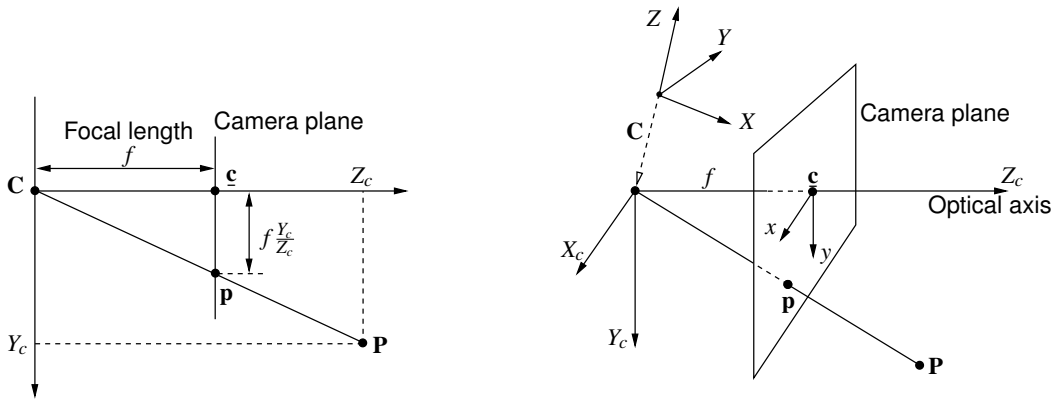


Figure 2.3: Perspective mapping of a 3D point **P** to a 2D point **p** onto the camera plane (from [90]).

center of projection $\underline{\mathbf{C}} = (C_x, C_y, C_z)^\top$ and the camera plane. As shown in Figure 2.3, the mapping of **P** to **p** can be determined using the intercept theorems:

$$x = f\frac{X_c}{Z_c} \qquad \text{and} \qquad y = f\frac{Y_c}{Z_c} \quad . \tag{2.1}$$

The principal point $\underline{\mathbf{c}} = (c_x, c_y)^\top$ is the intersection point of the optical axis and the camera plane. In equation (2.1), the center of projection and the principal point are identical. In real cameras, they are likely to have a differing position. Therefore, equation (2.1) is generalized to:

$$x = f\frac{X}{Z} + c_x \qquad \text{and} \qquad y = f\frac{Y}{Z} + c_y \quad . \tag{2.2}$$

Using projective geometry, the mapping equation (2.2) can be expressed as a linear system of equations. The 3D object point $\mathbf{P}$ and the 2D point $\mathbf{p}$ are represented by homogeneous vectors:

$$\mathbf{P}_c = (X_c, Y_c, Z_c, 1)^\top \quad \text{and} \qquad \mathbf{p} = (\check{x}, \check{y}, \check{z})^\top \quad . \tag{2.3}$$

With $x = \frac{\check{x}}{\check{z}}$ and $y = \frac{\check{y}}{\check{z}}$, from equation (2.2) follows:

$$\mathbf{p} = \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{P}_c \quad . \tag{2.4}$$

Using

$$\mathtt{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

it follows:

$$\mathbf{p} = \mathtt{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{P}_c = \mathtt{K}[\mathtt{E}|\mathbf{0}]\mathbf{P}_c \quad . \tag{2.6}$$

The matrix $\mathtt{K}$ is called calibration matrix and holds the intrinsic camera parameters f (focal length) and $\underline{\mathbf{c}} = (c_x, c_y)^\top$ (principal point). These parameters determine the mapping properties of the camera.

In equation (2.6), $\mathbf{P}_c$ is given in camera coordinates. The world point $\mathbf{P} = (X, Y, Z, 1)^\top$ is calculated by subtraction of $\underline{\mathbf{C}} = (C_x, C_y, C_z)^\top$ and the multiplication with a 3×3 rotation matrix $\mathtt{R}$

$$\mathbf{P}_c = \begin{bmatrix} \mathtt{R} & -\mathtt{R}\underline{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{P} \quad . \tag{2.7}$$

With equation (2.7), the mapping of $\mathbf{P}$ in world coordinates to $\mathbf{p}$ in camera coordinates is:

$$\mathbf{p} = \mathtt{K}[\mathtt{E}|\mathbf{0}] \begin{bmatrix} \mathtt{R} & -\mathtt{R}\underline{\mathbf{C}} \\ 0 & 1 \end{bmatrix} \mathbf{P} \tag{2.8}$$

$$\Leftrightarrow \quad \mathbf{p} = \mathtt{KR}[\mathtt{E}|-\underline{\mathbf{C}}]\mathbf{P} \quad . \tag{2.9}$$

The center of projection $\underline{\mathbf{C}}$ and the rotation matrix $\mathtt{R}$ determine the position and the orientation of the camera. They are called extrinsic parameters.

With the definition of the 3×4 matrix $\mathtt{A}$,

$$\mathtt{A} \;=\; \mathtt{K}\mathtt{R}\left[\mathtt{E}\middle|-\underline{\mathbf{C}}\right] \quad , \tag{2.10}$$

it follows:

$$\mathbf{p} \;=\; \mathtt{A}\mathbf{P} \quad . \tag{2.11}$$

The matrix $\mathtt{A}$ is called camera matrix. It holds all intrinsic and extrinsic camera parameters of the perspective mapping model. If the camera matrix is exactly decomposed into the parts described in equation (2.10) and (2.5), the camera matrix $\mathtt{A}$ determines a metric camera. Otherwise, $\mathtt{A}$ is called projective camera [32, 33, 43]. In the metric case, intrinsic and extrinsic camera parameters can be calculated explicitly from the matrix $\mathtt{A}$.

2.2.2 Geometric Lens Distortion Model

The lens system of a camera causes geometric distortion, which is not described by the perspective mapping model. The main reason is radial distortion. In this work, radial

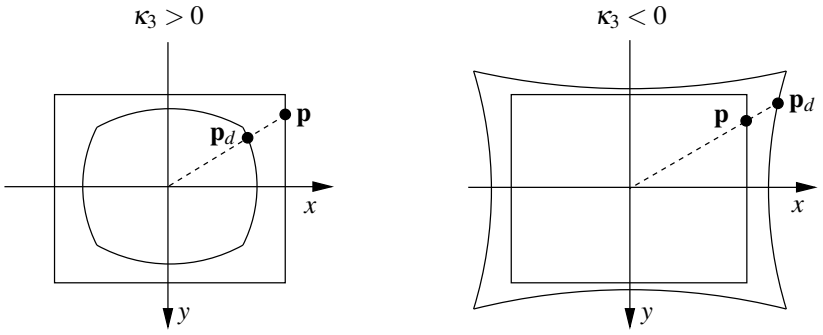


Figure 2.4: Compensation of radial distortion. The distorted point $\mathbf{p}_d$ is mapped to the undistorted point $\mathbf{p}$ (from [90]).

distortion is compensated using the following function

$$\mathbf{p} \;=\; \mathrm{diag}\left[1+\kappa_3\, d(\mathbf{p}_d,\mathbf{0})^2,\; 1+\kappa_3\, d(\mathbf{p}_d,\mathbf{0})^2,\; 1\right]\, \mathbf{p}_d \quad . \tag{2.12}$$

Here, $\mathrm{diag}[\ldots]$ is a diagonal matrix and $d(\ldots,\mathbf{0})$ the Euclidean distance from the origin of the coordinate system. Using equation (2.12), a distorted point $\mathbf{p}_d$ is mapped to the undistorted point $\mathbf{p}$. The parameter κ_3 is usually known from the camera manufacturer or is estimated using a calibration procedure [91, 92, 97]. In Figure 2.4, the compensation of radial distortion is shown for $\kappa_3 > 0$ on the left and for $\kappa_3 < 0$ on the right.

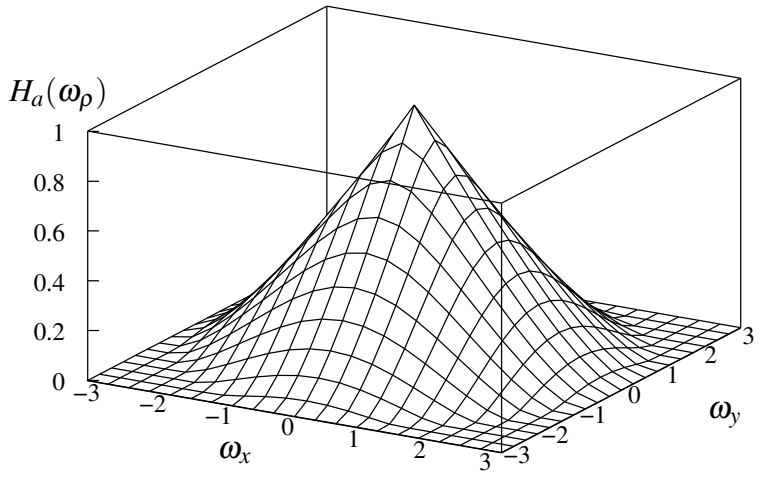


Figure 2.5: Modulation transfer function $H_a(\omega_\rho)$ with $\omega_\rho = \sqrt{\omega_x^2 + \omega_y^2}$ and $\omega_{\rho_c} = \pi$ (from [65]).

2.2.3 Image Signal Transfer Model

The signal transfer of the camera lens system, is determined by the image signal transfer function H_a which is called *modulation transfer function* (MTF) [41, 74]. The MTF is derived in [41]. H_a depends on the spatial frequencies ω_x und ω_y of the input image signal. For radial symmetric lenses, the H_a is rotationally symmetric and depends on the radial displacement $\omega_\rho = \sqrt{\omega_x^2 + \omega_y^2}$ in the frequency plane:

$$H_a(\omega_\rho) = \begin{cases} \frac{2}{\pi}\left(\arccos\left(\frac{\omega_\rho}{\omega_{\rho_c}}\right) - \frac{\omega_\rho}{\omega_{\rho_c}}\sqrt{1-\left(\frac{\omega_\rho}{\omega_{\rho_c}}\right)^2}\right) & , \omega_\rho \leq \omega_{\rho_c} \\ 0 & , \text{otherwise} \end{cases} . \tag{2.13}$$

The function H_a can be interpreted as a low pass filter with cut-off frequency ω_{ρ_c}. This is shown in Figure 2.5 with $\omega_{\rho_c} = \pi$. For the image signal transfer model, a Gaussian approximation of equation (2.13) is used [2, 65, 74]:

$$H_{ag}(\omega_\rho) = \begin{cases} e^{-\frac{\sigma_a^2}{2}\cdot\omega_\rho^2} & , \omega_\rho \leq \omega_{\rho_c} \\ 0 & , \text{otherwise} \end{cases} . \tag{2.14}$$

The variance σ_a^2 controls the low pass character of the MTF. Increasing σ_a^2 leads to a stronger low pass and more blurring in the image. In Figure 2.6, $H_{ag}(\omega_\rho)$ is shown for $\sigma_a^2 = 0.8$ and $\omega_{\rho_c} = \pi$.

The MTF is validated in an experiment using a TFT display showing a black pixel with white background and a real camera [48]. The impulse response captured by the camera is

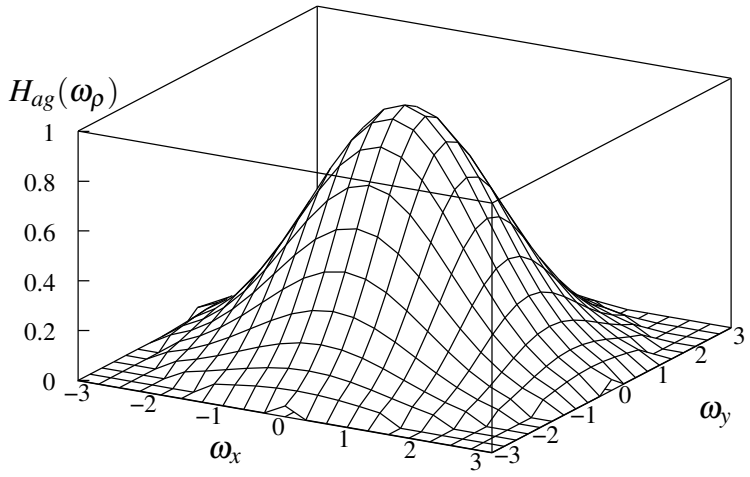


Figure 2.6: Modulation transfer function $H_{ag}(\omega_\rho)$ with $\omega_\rho = \sqrt{\omega_x^2 + \omega_y^2}$ for $\sigma_a^2 = 0.8$ and $\omega_{\rho_c} = \pi$ (from [65]).

shown in x and y direction in Figure 2.7. The Gaussian function is a close approximation for the impulse response of a real camera. Thus, it is considered for highly-accurate position measurements in camera images [17, 71, 72, 77].

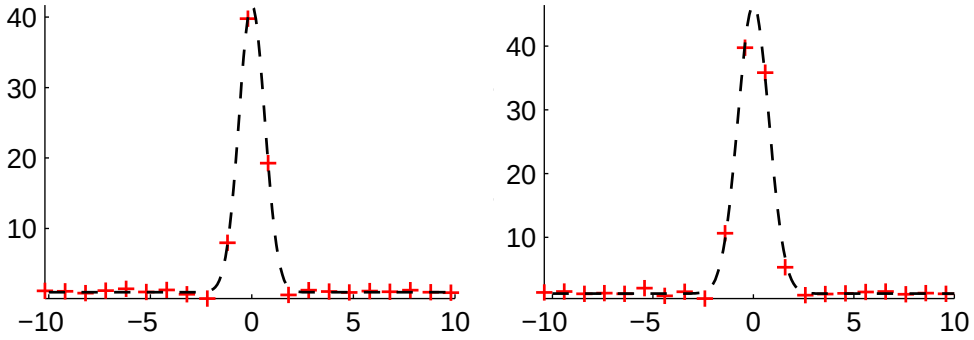
2.2.4 Sensor Model

Most video cameras use a CCD (*charge-coupled device*) chip to measure the ray intensity of the light passing through the camera lens. A CCD chip consists of a two-dimensional array of CCD elements as shown in Figure 2.8. Each of the CCD elements accumulates an electric charge proportional to the light intensity at its location. A control circuit causes each of the elements to transfer its contents to its neighbor. This operation is performed using a shift register. The resulting analogue signal is sampled and quantized. This leads to the discrete luminance signal $\mathtt{I}(n_x, n_y)$ with coordinates (n_x, n_y) in the picture memory of size $N_x \times N_y$.

Usually, a picture coordinate system is defined which measures the position of an image point in picture elements [px]. The origin of this coordinate system is in the top left corner of the picture memory as shown in Figure 2.9. A point $\underline{\mathbf{n}} = (n_x, n_y)^\top$ in the picture coordinate system can be transferred to a point $\mathbf{p}_d$ in the camera coordinate system by using:

$$\mathbf{p}_d = \begin{bmatrix} s_x & 0 & -0,5\,(N_x - 1)\,s_x \\ 0 & s_y & -0,5\,(N_y - 1)\,s_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} n_x \\ n_y \\ 1 \end{pmatrix} \quad . \tag{2.15}$$

In this equation, the point $\mathbf{p}_d$ in the camera coordinate system is given in homogeneous



(a) Horizontal scanline (b) Vertical scanline

Figure 2.7: Real camera (Canon XL-1) measurements of the impulse response using a peak on a TFT display. The red crosses show the sampling points relative to the peak position (x_d, y_d). A regression of the sampling points with a Gaussian function H_{ag} is shown with dotted lines (from [48]).

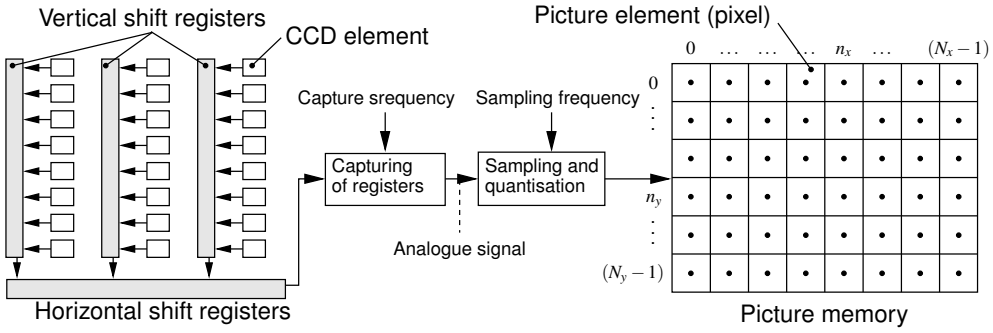


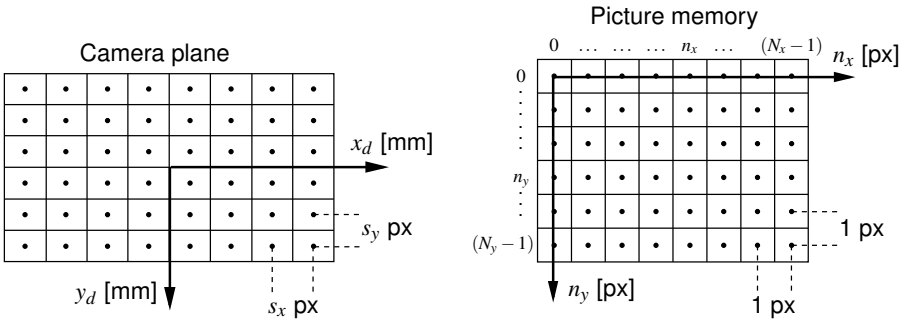
Figure 2.8: CCD-chip and digitizing of camera images (from [90]).

coordinates $\mathbf{p}_d = (x_d, y_d, 1)^\top$. For the computation, the knowledge of the geometric properties of the CCD sensor N_x, N_y, s_x, s_y is required. The scale factors s_x and s_y may be different.

2.3 Object Model

The static objects of a scene are represented by 3D object points as shown in Figure 2.10. They are reconstructed during camera and scene estimation resulting in a point cloud. The sparse representation using points guarantees a limited computational expense.

For the representation, projections of object points with distinct image properties are selected. The projection of the object point $\mathbf{P}_j$ into a camera image at time k is called feature point $\mathbf{p}_{j,k}$.



(a) Picture coordinate system (b) Camera coordinate system

Figure 2.9: Geometric relation between the picture coordinate system and the camera coordinate system (from [90]).

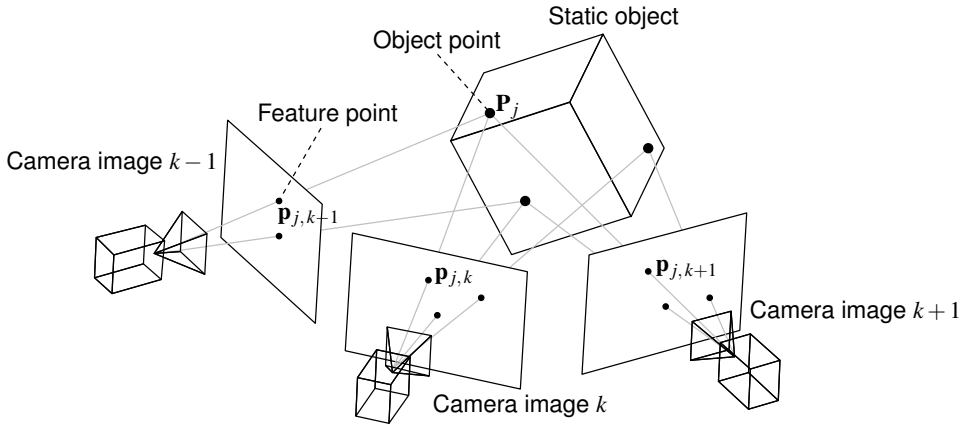


Figure 2.10: Representation of a static object with object points. The feature point $\mathbf{p}_{j,k}$ is the projection of the object point $\mathbf{P}_j$ into the camera image k (from [90]).

2.4 Model for Scale Invariance

Recent feature detection methods [7, 36, 51, 58, 61, 68, 69, 98] use a scale space representation to describe local regions in an image. This allows for the extraction of a distinctive feature description for a detected feature point $\mathbf{p}_{j,k}$. These points are used to establish correspondences between images of cameras with a wide baseline.

In Section 2.4.1, the fundamentals of the scale space are briefly reviewed. A close approximation of the scale space, the *Difference of Gaussians* (DoG) pyramid, is explained in Section 2.4.2.

2.4.1 Scale Space Representation

The scale space representation [55, 57, 58, 59] of an image is an embedding of the image into a derived one-parameter family of smoothed signals, intended to represent the original data at multiple scales. The scale space introduces a scale dimension and is defined [55] as follows:

Given a two dimensional continuous signal $f : \mathbb{R}^2 \to \mathbb{R}$, the scale space $L : \mathbb{R}^2 \times \mathbb{R}_+ \to \mathbb{R}$ is defined as the solution to the diffusion equation

$$\partial_t L = \frac{1}{2}\nabla^2 L \tag{2.16}$$

with initial condition $L(\cdot;0) = f$, or equivalently by convolution with the Gaussian kernel $G : \mathbb{R}^2 \times \mathbb{R}_+ \setminus \{0\} \to \mathbb{R}$:

$$L(\cdot;\sigma) = G(\cdot;\sigma) * f \quad , \tag{2.17}$$

where $*$ is the convolution operation and (cf. equation (3) of [55])

$$G(x,y;\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad . \tag{2.18}$$

The square of the standard deviation $\sigma^2 \in \mathbb{R}_+$ is denoted scale parameter.

In [57], it is shown that the only possible scale-space kernel is the Gaussian function. The main reason for this unique property is that it is the only kernel which does not produces new structures in the transformation from a finer to a coarser scale.

For the localization of scale space extrema, the Laplace operator in equation (2.17) is used:

$$\nabla^2 L = \frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2} \quad . \tag{2.19}$$

It is approximated efficiently by the *Difference of Gaussians* (DoG) pyramid [15].

2.4.2 Difference of Gaussians Pyramid

The approximation of the Laplace operator used for the detection of scale space extrema in [13, 61] is the *Difference of Gaussians* (DoG).

The scale-space of an image $\mathtt{I}(x,y)$ is obtained by the convolution of a variable-scale Gaussian $G(x,y,\sigma)$ (cf. equation (2.17))

$$L(x,y;\sigma) = G(x,y;\sigma) * \mathtt{I}(x,y) \quad , \tag{2.20}$$

where $*$ is the convolution operation and $G(x,y;\sigma)$ is the Gaussian function from equation (2.18).

The DoG pyramid uses neighboring scales which are separated by a constant multiplicative factor k [13]:

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * \mathtt{I}(x,y) \quad . \tag{2.21}$$

It is shown in [61], that the approximation of the Laplacian with equation (2.21) has no significant impact on the stability of the extrema detection and localization. To reduce the

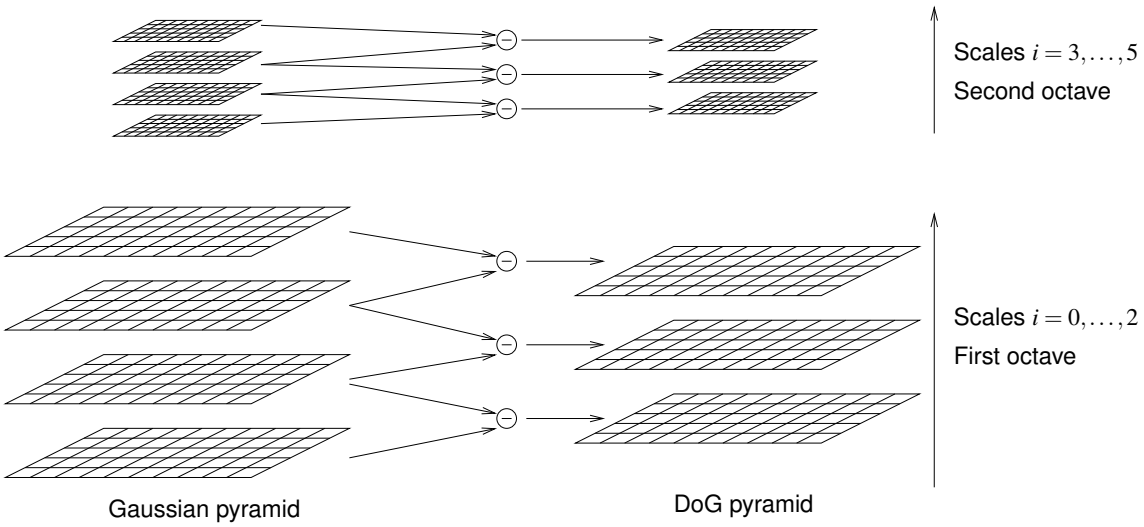


Figure 2.11: Difference of Gaussians for the localization of scale space maxima. The layers of the DoG are obtained by subtracting neighboring scales in an Octave.

noise of the input image, the first scale of each octave is smoothed by a Gaussian filter with the prior smoothing σ_0. The standard deviation σ at a scale i can be calculated as

$$\sigma = \sigma_0 \cdot k^i = \sigma_0 \cdot (2^{\frac{1}{N}})^i = \sigma_0 \cdot 2^{\frac{i}{N}} \quad , \tag{2.22}$$

where N is the number of scales per octave and k determines the separation of neighboring scales. For the input image, an inherent smoothing of σ_{init} is assumed [61]. Thus, the first layer of the first octave is smoothed with the remaining standard deviation $\sqrt{\sigma_0^2 - \sigma_{\text{init}}^2}$ only. To calculate the layer i from a given variance σ, equation (2.22) is transformed to:

$$i = N \cdot (log_2(\frac{\sigma}{\sigma_0})) \quad . \tag{2.23}$$

The Gaussian kernel described in equation (2.18) is uniform, which means that bivariate shapes of features are not necessarily detected. However, it is shown that they are detected up to a certain level of affine distortion which is sufficient for the most applications. The approach of sampling different bimodal scale spaces in [101] leads to a fully affine invariant detection with the limitation of the sampling rate only.

3 Structure and Motion Recovery Reference

In this Chapter, the *structure and motion* (SAM) recovery reference is explained. It uses state of the art algorithms similar to the work of Thormählen [90]. Additionally, the SIFT framework is used as the reference method for wide baseline feature correspondences.

Several reasonable combinations of feature matching and tracking techniques are considered. The resulting feature correspondences provide the input data for the maximum likelihood scene estimation. In Figure 3.1 the workflow of the SAM recovery algorithm

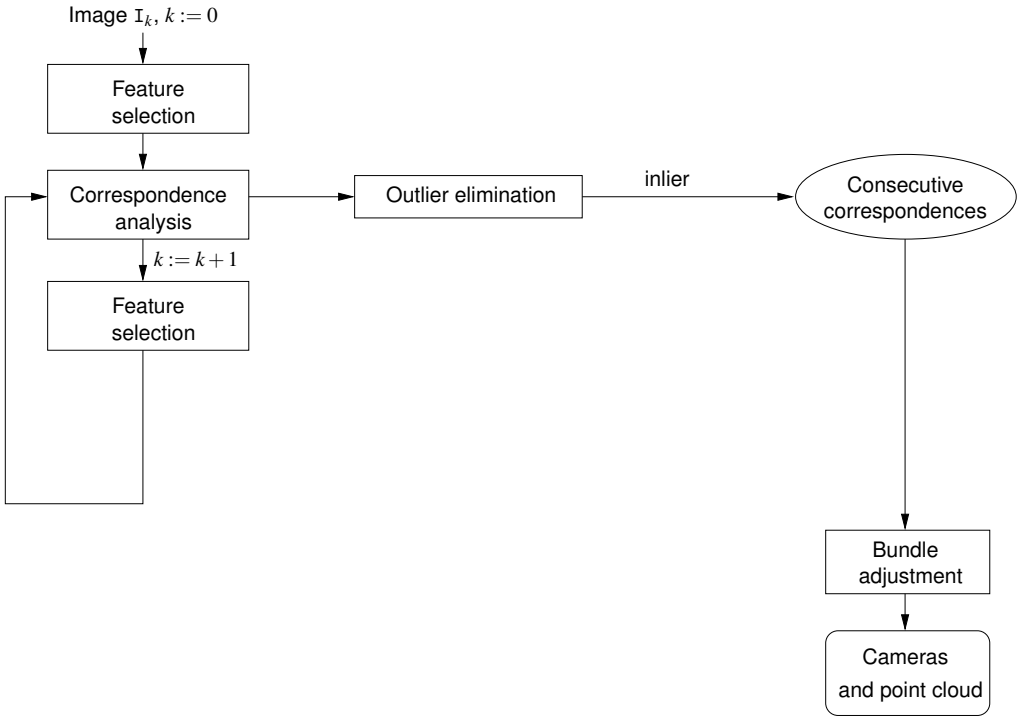


Figure 3.1: The workflow of the *structure and motion* (SAM) recovery reference [25].

is shown. In the first image of the sequence, features are selected (cf. Section 3.1). Corresponding feature points in consecutive images are obtained from a correspondence analysis step (cf. Section 3.2). The correspondences are validated using an outlier elimination method (cf. Section 3.3). The resulting inliers are used for the scene estimation with a bundle adjustment technique (cf. Section 3.4). The final results are the estimated camera parameters and a 3D point cloud which represents the static objects of the scene.

The Sections 3.1.1, 3.2.1, 3.3, and 3.4 are based on [90] since here, a similar reference method for the SAM recovery is employed. The Sections 3.1.2 and 3.2.2 are based on the work in [13, 61].

3.1 Feature Selection

The selection of local features in an images is a fundamental step in image processing. It reduces the large amount of data provided by an image to the most relevant image content. For scene estimation, two methods for feature detection turned out to be the most important techniques. The Harris detector [42, 83] provides feature points which are invariant to translation, rotation and illuminance. The features are located mainly on corners.

The most prominent detector using the scale space for the feature representation is known as the SIFT (*scale invariant feature transform*) detector [61]. Due to the scale space property, SIFT features are additionally invariant to scale compared to Harris corners. The shape of a SIFT feature is characterized as a local extremum with a continuous grey-level function at a certain scale, a so-called blob [55].

While a Harris corner is appropriate as a starting point for frame to frame tracking in video, the SIFT detector provides stable features between images of cameras with a wide baseline. An example for the need of a wide baseline correspondence analysis in video is given when an object point is temporarily occluded and the correspondence between the features before and after the occlusion is to be found.

3.1.1 Harris Corners

The Harris corner detector selects points which are likely to establish a stable correspondence in the next image. This is the case if there are strong gradients in perpendicular directions at the position of the feature point. The *cornerness response function* (CRF) [42] evaluates this property. To select feature points, the CRF for each pixel position $\underline{\mathbf{n}} = (x_n, y_n)^\top$ in the image is calculated,

$$\mathrm{CRF}(\underline{\mathbf{n}}) \quad = \quad \det\left(\mathtt{G}(\underline{\mathbf{n}})\right) + K_{\mathrm{CRF}}\ \mathrm{trace}^2\left(\mathtt{G}(\underline{\mathbf{n}})\right) \quad , \tag{3.1}$$

where $\det(\cdot)$ denotes the determinant and $\mathrm{trace}(\cdot)$ the trace of a matrix. To calculate the 2×2 matrix $\mathtt{G}(\underline{\mathbf{n}})$, a 7×7 sized window centered at $\underline{\mathbf{n}}$ with pixels $\underline{\mathbf{m}} = (x_m, y_m)^\top$ is evaluated:

$$\mathtt{G}(\underline{\mathbf{n}}) = \sum_{x_m} \sum_{y_m} \begin{bmatrix} g_x^2(\underline{\mathbf{n}}+\underline{\mathbf{m}}) & g_x(\underline{\mathbf{n}}+\underline{\mathbf{m}})\, g_y(\underline{\mathbf{n}}+\underline{\mathbf{m}}) \\ g_x(\underline{\mathbf{n}}+\underline{\mathbf{m}})\, g_y(\underline{\mathbf{n}}+\underline{\mathbf{m}}) & g_y^2(\underline{\mathbf{n}}+\underline{\mathbf{m}}) \end{bmatrix} \quad . \tag{3.2}$$

The image signal gradients g_x and g_y are approximated as follows

$$\frac{\partial I(x,y)}{\partial x} \approx g_x = \frac{I(x+1,y) - I(x-1,y)}{2} \quad , \tag{3.3}$$

$$\frac{\partial I(x,y)}{\partial y} \approx g_y = \frac{I(x,y+1) - I(x,y-1)}{2} \quad . \tag{3.4}$$

A threshold S_{CRF} determines the appropriate value for the CRF. All points with a CRF-value larger than S_{CRF} are stored in a list and sorted by their CRF-value. A feature is discarded, if its distance to all previously selected points is smaller than a minimal distance M_{CRF}. This is to achieve a better distribution of the feature points in the image. Now, the best B_{CRF} points of the list are chosen as image features.

Typical parameters are [90]:

$$K_{\text{CRF}} = 0.04,\ S_{\text{CRF}} = 46400,\ M_{\text{CRF}} = 10\,\text{px},\ B_{\text{CRF}} = 1000 \quad . \tag{3.5}$$

3.1.2 SIFT Features

The SIFT feature detector [61] employs a workflow as shown in Figure 3.2. Scale space extrema are detected using the *Difference of Gaussians* (DoG) pyramid (cf. Section 2.4.1). The following localization procedure refines the detected discrete coordinates of the scale space extrema. In order to apply an orientation parameter to a feature, the main orientation of the surrounding image gradients is estimated. If the orientation estimation procedure leads to ambiguous results, multiple features with the same coordinates, but different orientation are constructed. Finally, a 128 dimensional vector is computed using the surrounding gradient orientations. This vector is called descriptor. It is used to establish the correspondence to a feature in a second image. Correspondences between two images are found by associating feature points with a minimal distance between their descriptors.

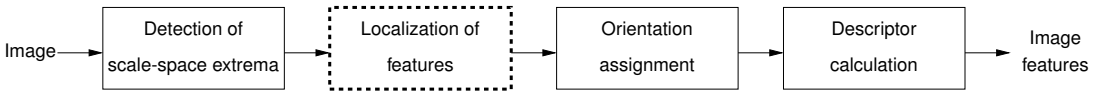


Figure 3.2: Workflow diagram for feature detection with the SIFT detector. The localization accuracy is determined by the localization step marked with a dotted box border (from [19]).

Detection of Scale Space Extrema

For each scale i in every octave o in the DoG pyramid, a feature detection procedure is applied. A feature point is characterized as a pixel position with a DoG value larger

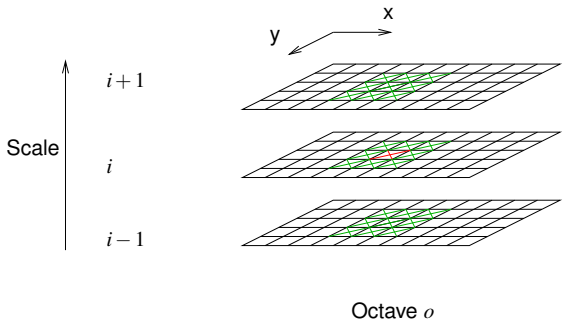


Figure 3.3: Detection of scale space extrema in the DoG pyramid by comparison of a pixel with its 26 neighbors.

or smaller than its 26 neighbors. The evaluated neighborhood is shown in Figure 3.3. The result is an image feature candidate $\delta(\underset{\sim}{\mathbf{n}}) = (\delta(n_x), \delta(n_y), \delta(n_i))^\top \in \mathbb{N}^3$ with discrete pixel and scale coordinates. In the next step, the localization is refined by an interpolation of the 27 grid points with a function model.

Localization of Features

The localization procedure aims at refining the coordinates of the initially detected feature $\hat{\delta}(\underset{\sim}{\mathbf{n}}) = (\hat{\delta}(n_x), \hat{\delta}(n_y), \hat{\delta}(n_i))^\top \in \mathbb{N}^3$ by adding a subpixel and subscale part $\hat{\varepsilon}(\underset{\sim}{\mathbf{n}}) = (\hat{\varepsilon}(n_x), \hat{\varepsilon}(n_y), \hat{\varepsilon}(n_i))^\top \in \mathbb{R}^3$ to estimate the feature position $\hat{\underset{\sim}{\mathbf{n}}} = \hat{\delta}(\underset{\sim}{\mathbf{n}}) + \hat{\varepsilon}(\underset{\sim}{\mathbf{n}})$. The image signal of the DoG pyramid is interpolated with a function model using the 27 sample points surrounding a scale space extremum. The function model is chosen as a 3D quadratic [13]. Thus, the gradient signal with fullpixel sample points $D(\delta(\underset{\sim}{\mathbf{n}}))$ is approximated by the parabolic function $\hat{D}(\delta(\underset{\sim}{\mathbf{n}}))$:

$$\hat{D}(\underset{\sim}{\mathbf{n}}) = D(\hat{\delta}(\underset{\sim}{\mathbf{n}})) + \frac{\partial D(\hat{\delta}(\underset{\sim}{\mathbf{n}}))^T}{\partial \underset{\sim}{\mathbf{n}}} \underset{\sim}{\mathbf{n}}^T + \frac{1}{2} \underset{\sim}{\mathbf{n}}^T \frac{\partial^2 D(\hat{\delta}(\underset{\sim}{\mathbf{n}}))}{\partial \underset{\sim}{\mathbf{n}}^2} \underset{\sim}{\mathbf{n}} \quad . \tag{3.6}$$

The least squares solution of this equation is calculated using the inverse hessian matrix $\underset{\sim}{\mathrm{H}}^{-1}$:

$$\hat{\underset{\sim}{\mathbf{x}}} \;=\; -\begin{pmatrix} D_{xx} & D_{xy} & D_{xi} \\ D_{yx} & D_{yy} & D_{yi} \\ D_{ix} & D_{iy} & D_{ii} \end{pmatrix}^{-1} \begin{pmatrix} D_x \\ D_y \\ D_i \end{pmatrix} \quad . \tag{3.7}$$

The derivatives D_x, D_y, and D_i and the entries of the matrix $\underset{\sim}{\mathrm{H}}$ in equation (3.7) are listed in the Appendix Section 8. The resulting error of the estimation is computed by the residuum $\varepsilon_{\text{SIFT}}$:

$$\varepsilon_{\text{SIFT}} = \sum_i \sum_y \sum_x d(\hat{D}(\hat{\delta}(\underset{\sim}{\mathbf{n}}) - \underset{\sim}{\mathbf{n}}_0), D(\hat{\delta}(\underset{\sim}{\mathbf{n}}) - \underset{\sim}{\mathbf{n}}_0)) \tag{3.8}$$

for all grid points $(\hat{\underset{\sim}{\delta}}(\underline{\mathbf{n}}) - \underline{\mathbf{n}}_0)$ with $\underline{\mathbf{n}}_0 = (x, y, i)$, $x, y, i \in \{-1, 0, 1\}$ in the 27 neighborhood of the fullpixel coordinate $\hat{\underset{\sim}{\delta}}(\underline{\mathbf{n}})$.

The subpixel localization is performed in an iterative procedure. The interpolation is restarted if one of the subpixel or subscale values misses the interval $[-0.5; 0.5]$. Then, the interpolation is reinitialized with the fullpixel position which is nearest to the last computed position. If the number of iterations exceeds $iter_{\text{SIFT}} = 5$, the feature location is regarded as unstable and the feature point is discarded.

The value $|D(\hat{\underline{\mathbf{n}}})| = |D(\hat{\underset{\sim}{\delta}}(\underline{\mathbf{n}}) + \hat{\varepsilon}(\underline{\mathbf{n}}))|$ at the refined position of a feature candidate is called *contrast* of the feature [61]. After the subpixel and subscale localization, feature candidates with a low contrast are rejected. This is evaluated by applying a threshold D_{SIFT}^{min} to the DoG value $|D(\hat{\underline{\mathbf{n}}})|$. Feature candidates with a DoG value smaller than D_{SIFT}^{min} are rejected.

Although feature points located on edges have a large contrast value, they lead to uncertainties when they are used for camera motion estimation (cf. Section 3.1.1). To avoid these feature locations, SIFT uses a selection technique which is similar to the CRF equation (3.1). The hessian matrix $\mathtt{H}$ is considered at the location $\hat{\underline{\mathbf{n}}}$ of the feature point:

$$\mathtt{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \quad . \tag{3.9}$$

The eigenvalues of $\mathtt{H}$ determine the principal curvature r_{SIFT} of the feature point. Assume, that $\alpha_{\mathtt{H}}$ is the larger and $\beta_{\mathtt{H}}$ is the smaller eigenvalue. The sum of the eigenvalues can be computed by the trace of $\mathtt{H}$. The product of the eigenvalues can be computed by the determinant (cf. Section 3.1.1).

$$\begin{aligned} \text{trace}(\mathtt{H}) &= D_{xx} + D_{yy} = \alpha_{\mathtt{H}} + \beta_{\mathtt{H}} && (3.10) \\ \det(\mathtt{H}) &= D_{xx} D_{yy} - (D_{xy})^2 = \alpha_{\mathtt{H}} \beta_{\mathtt{H}} \quad . && (3.11) \end{aligned}$$

To avoid eigenvalues with different signs, a feature point is discarded if the determinant $\det(\mathtt{H})$ is negative. Let r_{SIFT} be the ratio of $\alpha_{\mathtt{H}}$ and $\beta_{\mathtt{H}}$, $r_{\text{SIFT}} = \frac{\alpha_{\mathtt{H}}}{\beta_{\mathtt{H}}}$. It follows:

$$\frac{\text{trace}(\mathtt{H})^2}{\det(\mathtt{H})} = \frac{(\alpha_{\mathtt{H}} + \beta_{\mathtt{H}})^2}{\alpha_{\mathtt{H}} \beta_{\mathtt{H}}} = \frac{(r_{\text{SIFT}} + 1)^2}{r_{\text{SIFT}}} \quad . \tag{3.12}$$

The quantity $\frac{(r_{\text{SIFT}}+1)^2}{r_{\text{SIFT}}}$ adopts its minimum if the two eigenvalues are equal and it increases with r_{SIFT}. Thus, an appropriate threshold evaluation for the ratio of the principal curvatures is:

$$\frac{\text{trace}(\mathtt{H})^2}{\det(\mathtt{H})} < \frac{(r_{\text{SIFT}}^{max} + 1)^2}{r_{\text{SIFT}}^{max}} \tag{3.13}$$

using a threshold value r_{SIFT}^{max}. Like in the Harris approach using the CRF, the eigenvalues of $\mathtt{H}$ must not be computed.

Orientation Assignment

To obtain a rotational invariant feature description, an orientation value Θ is assigned to each feature $\underset{\sim}{\mathbf{n}} = (n_x, n_y, n_i)^\top$. For each sample (x,y) inside a radius of $r_{Ori} * \sigma_{i_n}$ at scale i_n

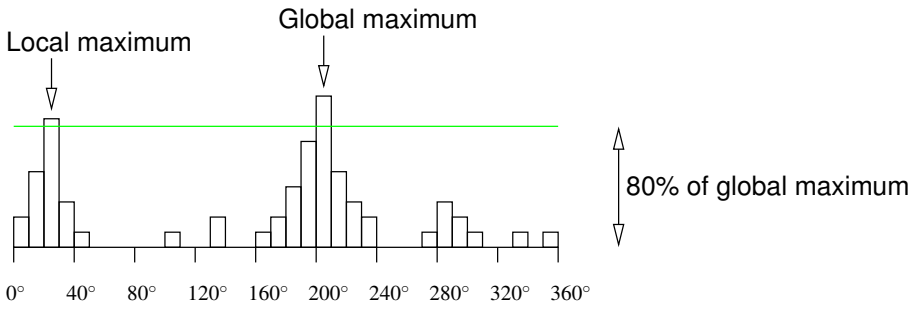


Figure 3.4: Orientation histogram of gradients in a neighborhood of a feature point. If there is a second local maximum with at least 80% magnitude of the global maximum, an additional feature at the same position with a different orientation Θ is created.

of the current octave of the Gaussian pyramid, the gradient magnitude $mag(x,y)$ and the orientation $\theta(x,y)$ is computed using pixel differences:

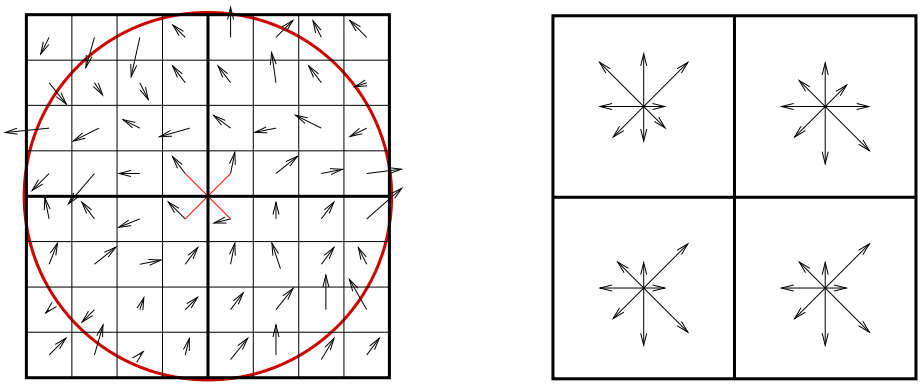
$$mag(x,y) = \sqrt{(G(x+1,y) - G(x-1,y))^2 + (G(x,y+1) - G(x,y-1))^2} \quad (3.14)$$

$$\theta(x,y) = \arctan \frac{G(x,y+1) - G(x,y-1)}{G(x+1,y) - G(x-1,y)} \quad (3.15)$$

with the abbreviation $G(x,y) := G(x,y,\sigma_{i_n})$. The values $\theta(x,y)$ weighted with $mag(x,y)$ form a histogram of gradients with 36 bins as illustrated in Figure 3.4. The highest peak in the histogram determines the orientation of the feature. If there is a second local maximum with at least 80% magnitude of the global maximum, an additional feature at the same position is created with that orientation. Finally, a parabola is fit to interpolate the orientation peak position providing the feature orientation Θ.

Descriptor Calculation

The descriptor of a SIFT feature $\underset{\sim}{\mathbf{n}} = (n_x, n_y, n_i)^\top$ is computed from the gradient magnitudes $mag(x,y)$ and orientations $\theta(x,y)$. The values for $mag(x,y)$ and $\theta(x,y)$ are sampled in a $N_{\text{SIFT}} \times N_{\text{SIFT}}$ neighborhood of the feature point in its detected scale i_n of the current octave of the Gaussian pyramid. The neighborhood patch is compensated for rotation by using the estimated orientation Θ and bilinear interpolation for pixel access. A Gaussian weighting function with variance $\frac{N_{\text{SIFT}}}{2}$ is applied to the magnitude values. This is illustrated in Figure 3.5. On the left, a neighborhood patch of size 8×8 is shown. The



(a) Image gradients (b) SIFT feature descriptor

Figure 3.5: The magnitude $mag(x,y)$ and the orientation $\theta(x,y)$ of the gradients in the neighborhood of a feature point is used to form a SIFT descriptor. The gradients are weighted by a Gaussian window, indicated by the red circle. Using orientation histograms of the weighted $\Theta(x,y)$, the contents of subregions are accumulated as shown on the right. The figure shows a 2×2 descriptor array computed from an 8×8 sample set. The resulting descriptor vector has a dimension of $4 \cdot 8 = 32$.

weighted gradient magnitudes are summed up in histograms to form a 2×2 descriptor array, which is illustrated on the right. Using 8 discrete orientations for each descriptor sample, the resulting descriptor vector has length $4 \cdot 8 = 32$. The reference implementation of SIFT uses a 16×16 patch size and 4×4 descriptors resulting in a vector of length $l_{\text{SIFT}} = 4 \cdot 4 \cdot 8 = 128$.

To obtain a quality measure for detected SIFT features, their contrast values are used [5, 62]. After sorting the resulting feature list by the contrast values, the best B_{SIFT} features are chosen.

The reference implementation uses the parameters [61]:

$$\sigma_0 = 1.6,\ r_{Ori} = 4.5,\ N_{\text{SIFT}} = 16,\ l_{\text{SIFT}} = 128,\ B_{\text{SIFT}} = 3000 \tag{3.16}$$

and the threshold values

$$r_{\text{SIFT}}^{max} = 10,\ D_{\text{SIFT}}^{min} = 0.03 \quad . \tag{3.17}$$

3.2 Correspondence Analysis

The objective of the correspondence analysis is to build a trajectory $\mathbf{t}$ which connects a number of images using feature points. Points of a trajectory $\mathbf{t}_j$ are projections of one

object point $\mathbf{P}_j$ into different camera images k:

$$\mathbf{t}_j = (\mathbf{p}_{j,U}, \ldots, \mathbf{p}_{j,k}, \ldots, \mathbf{p}_{j,K}) \quad . \tag{3.18}$$

The frame $k = U$ denotes the image, in which the feature point $\mathbf{p}_j$ is detected. The frame $k = K$ denotes the current image. The length of a trajectory $|\mathbf{t}_j|$ is number of trajectory elements:

$$|\mathbf{t}_j| = |\{\mathbf{p}_{j,k} | k = U, \ldots, K\}| \quad . \tag{3.19}$$

Two methods for correspondence analysis are explained in Section 3.2.1 and in Section 3.2.2. Feature tracking represented by the KLT tracker and feature matching represented by SIFT.

3.2.1 KLT Feature Tracking

After selecting feature points in an image, the corresponding feature point in the consecutive image is determined. As common video cameras us a frame rate of at least 24 frames per second, a small change in position and perspective is assumed. Thus, the gradient based approach proposed by Kanade, Lucas, and Tomasi (KLT-Tracker) [63, 85, 40] is used.

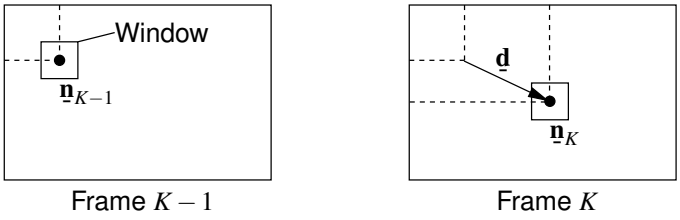


Figure 3.6: Displacement $\underline{\mathbf{d}}$ between corresponding feature points $\underline{\mathbf{n}}_{K-1}$ and $\underline{\mathbf{n}}_K$ (from [90]).

To determine the correspondence between feature point $\underline{\mathbf{n}}_{K-1}$ in the previous image $K-1$ and $\underline{\mathbf{n}}_K$ in the current image K, a 7×7 window with its center at $\underline{\mathbf{n}}_{K-1}$ is evaluated. The displacement $\underline{\mathbf{d}} = (\underline{\mathbf{n}}_K - \underline{\mathbf{n}}_{K-1})$ as shown in Figure 3.6 is calculated by minimizing the *sum of squared distances* (SSD):

$$\mathrm{SSD}(\underline{\mathbf{d}}) = \sum_{x_m} \sum_{y_m} [I_K(\underline{\mathbf{n}}_{K-1} + \underline{\mathbf{d}} + \underline{\mathbf{m}}) - I_{K-1}(\underline{\mathbf{n}}_{K-1} + \underline{\mathbf{m}})]^2 \rightarrow \min \quad . \tag{3.20}$$

By using the approximation

$$I_K(\underline{\mathbf{n}}_{K-1} + \underline{\mathbf{d}} + \underline{\mathbf{m}}) \approx I_K(\underline{\mathbf{n}}_{K-1} + \underline{\mathbf{m}}) + \begin{pmatrix} g_x(\underline{\mathbf{n}}_{K-1} + \underline{\mathbf{m}}) \\ g_y(\underline{\mathbf{n}}_{K-1} + \underline{\mathbf{m}}) \end{pmatrix}^\top \underline{\mathbf{d}} \quad , \tag{3.21}$$

the solution is found by substituting equation (3.21) in (3.20) and setting the derivative of the resulting term to zero:

$$\sum_{x_m}\sum_{y_m} 2\left[I_K(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) + \begin{pmatrix} g_x(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) \\ g_y(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) \end{pmatrix}^\top \underline{\mathbf{d}} - I_{K-1}(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}})\right] \begin{pmatrix} g_x(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) \\ g_y(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) \end{pmatrix} = \mathbf{0} \quad . \tag{3.22}$$

Thus, the displacement $\underline{\mathbf{d}}$ is:

$$\underline{\mathbf{d}} = \mathtt{G}(\underline{\mathbf{n}}_{K-1})^{-1}\, \mathbf{b}(\underline{\mathbf{n}}_{K-1}) \tag{3.23}$$

with

$$\mathtt{G}(\underline{\mathbf{n}}_{K-1}) = \sum_{x_m}\sum_{y_m} \begin{bmatrix} g_x^2(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) & g_x(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}})\, g_y(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) \\ g_x(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}})\, g_y(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) & g_y^2(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) \end{bmatrix} \tag{3.24}$$

and

$$\mathbf{b}(\underline{\mathbf{n}}_{K-1}) \;=\; \sum_{x_m}\sum_{y_m} [I_{K-1}(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) - I_K(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}})] \begin{pmatrix} g_x(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) \\ g_y(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) \end{pmatrix} . \tag{3.25}$$

The displacement $\underline{\mathbf{d}}$ in equation (3.23) is calculated with subpixel accuracy. A bilinear interpolation [9] is incorporated for the calculation of the image intensities.

Because of the approximation in equation (3.21), equation (3.23) is also an approximation of $\underline{\mathbf{d}}$. Thus, the result is improved by calculating equation (3.23) iteratively. The distance $\Delta\underline{\mathbf{d}}$ between two iterations is:

$$\Delta\underline{\mathbf{d}} = \mathtt{G}(\underline{\mathbf{n}}_{K-1})^{-1} \sum_{x_m}\sum_{y_m} [I_{K-1}(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) - I_K(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{d}}'+\underline{\mathbf{m}})] \begin{pmatrix} g_x(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) \\ g_y(\underline{\mathbf{n}}_{K-1}+\underline{\mathbf{m}}) \end{pmatrix} \quad , \tag{3.26}$$

where $\underline{\mathbf{d}}'$ denotes the displacement of the previous iteration. The iterative computation is aborted if $\Delta\underline{\mathbf{d}}$ is smaller than a threshold D_{KLT} or after a maximum number of iteration O_{KLT} is exceeded. A feature correspondence is discarded if after the minimization $\sqrt{\mathrm{SSD}(\underline{\mathbf{d}})}/7^2$ is larger than a threshold E_{KLT}.

The approximation in equation (3.21) is valid for displacements of up to 3 px. For larger displacements, the iterative minimization is likely to diverge. By using a resolution pyramid as shown in Figure 3.7, larger displacements are estimated reliably. Therefore, Υ_{KLT} resolutions of the camera images $K-1$ and K are computed by lowpassfiltering and subsampling with factor 2. At first, the displacement vector is determined in the coarsest scale of the resolution pyramid. This vector is mapped to the next scale of the pyramid and is used as initial value $\underline{\mathbf{d}}'$ for iterative optimization in this scale. By repeating this procedure, the displacement vector in the base image of resolution $N_x \times N_y$ is computed.

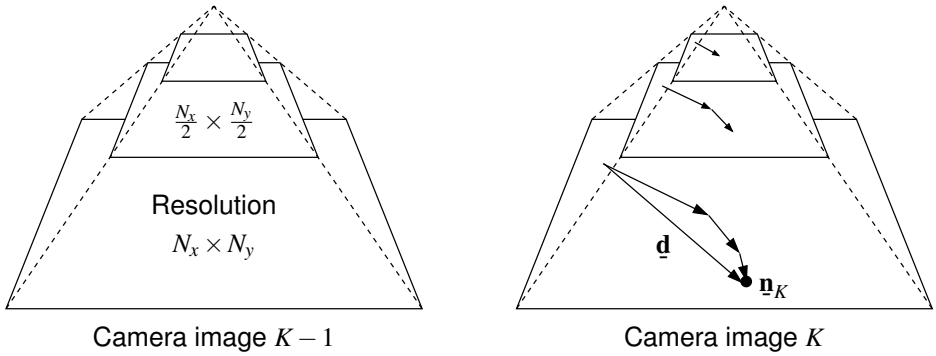


Figure 3.7: Usage of a resolution pyramid for the computation of large displacements $\underline{\mathbf{d}}$ (from [90]).

In the reference algorithm, the following parameters are used for the correspondence analysis [90]:

$$D_{\text{KLT}} = 0.01\,\text{px},\ O_{\text{KLT}} = 10,\ E_{\text{KLT}} = 10.0,\ \Upsilon_{\text{KLT}} = 3 \quad . \tag{3.27}$$

3.2.2 Feature Matching using SIFT

The SIFT correspondence analysis is done by a feature matching technique using the descriptors of the features. The best match for each feature in the first image is found by identifying its nearest neighbor in the set of features detected in the second image. But, the correspondence is discarded, if the second-closest neighbor of the feature set has a similar distance from the observed feature in the first image. If the nearest distance is $d_{closest}$, and the second nearest neighbor has a distance $d_{2closest}$, the correspondence is discarded, if:

$$\frac{d_{closest}}{d_{2closest}} > \tau_{\text{SIFT}} \quad . \tag{3.28}$$

As demonstrated in [61], this increases the distinctiveness of the resulting set of correspondences and avoids mismatches. The parameter τ_{SIFT} is set to [61]:

$$\tau_{\text{SIFT}} = 0.8 \quad . \tag{3.29}$$

3.2.3 Feature Selection in the First Image

In the first camera image, a set of features is selected which is tracked for the next images of the sequence. To obtain a user determined number of features, a quality measure is required to select the best features of the set.

Harris Features

To obtain a feature set in the first image, Harris corners are detected as explained in Section 3.1.1. The feature list is sorted by descending CRF values (from biggest to smallest). The first B_{CRF} features are selected with the constraint that each feature has a minimal distance M_{CRF} to all selected features in the list.

SIFT Features

For SIFT features, a commonly used property determining the quality of the feature is the contrast value $|D(\hat{\underset{\sim}{\mathbf{n}}})|$ at its position $\hat{\underset{\sim}{\mathbf{n}}}$ [5, 62]. Thus, the feature list is sorted by descending contrast values (from biggest to smallest) and the B_{SIFT} features in the list are selected.

3.2.4 Selection of new Features

Due to a limited lifetime of feature trajectories and due to newly appearing scene content, new features are selected in each image to guarantee a constant number of features in each image of the sequence. Like in the first image, a quality measure is required to complete the set of tracked features with the best possible newly detected features.

Harris Features

Like in the first image, newly detected Harris corners are sorted by descending CRF values. The list of tracked features is added with these features until B_{CRF} is achieved considering the minimal distance M_{CRF} to all features in the list.

SIFT Features

Like in the first image, newly detected SIFT features are sorted by descending contrast values. The list of tracked features is added with these features until B_{SIFT} is achieved.

3.3 Outlier Elimination

The outlier elimination step removes erroneous correspondences from a correspondence set. Outliers are correspondences of features which arise from different 3D points. These correspondences result from coincidental similarities in the image signal or if the 3D point is occluded by a foreground object in the second image. Additionally, outliers result from correspondences on moving objects, because our scene model is restricted to static geometry. If the camera parameters are known, the possibility for the position of a corresponding point is limited to a small stripe. Thus, outliers are eliminated by estimating initial camera parameters using the *random sample consensus* (RANSAC) algorithm [34]. RANSAC allows for initial camera parameter estimation although the correspondence set

is spoiled with outliers. The mathematical context is given by the mapping model of the corresponding feature points in the two images. As shown in Figure 3.8, a number β_{RSC}

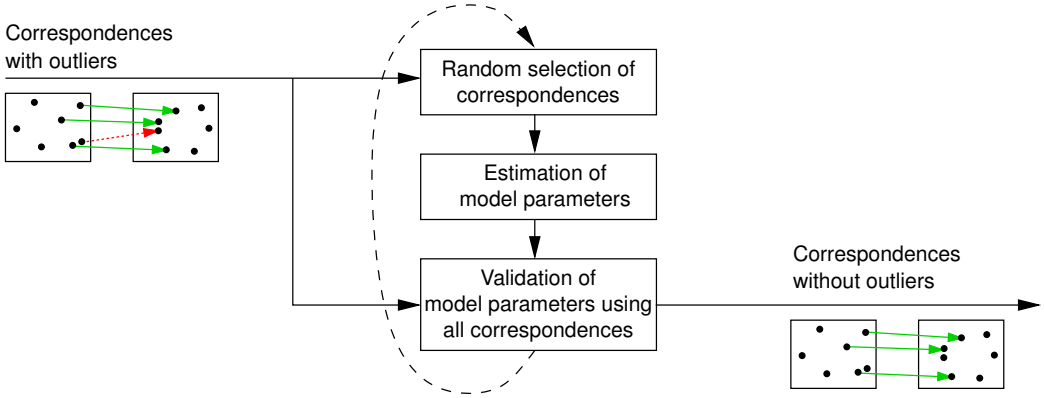


Figure 3.8: Outlier elimination using RANSAC (from [90]).

of correspondences is selected randomly. β_{RSC} is determined by the minimal number of parameters needed for a unique mapping. The model parameters are estimated using the selected correspondences. Then, all correspondences are validated and the number of correspondences supporting the model is counted. A correspondence is assumed to support the model if the distance ε between the corresponding point and the point determined by the mapping is smaller than a maximum distance ε_{max}. The ratio R_{RSC} between the supported correspondences and the total number of correspondences is calculated. If R_{RSC} is below a threshold R_{RSC}, the iterative algorithm is aborted. Otherwise, a new try with randomly selected correspondences is started. After a maximum number N_{RSC} of iterations, the algorithm is aborted. For the application experiments in Sections 6.3 and 6.4, the following parameters are used:

$$R_{\text{RSC}} = 0.95, \quad N_{\text{RSC}} = 4000 \quad . \tag{3.30}$$

For the outlier elimination two different mapping models are considered in this work. If no 3D object points are estimated yet, the fundamental matrix is used only. Otherwise, the fundamental matrix is used followed from the evaluation with the camera matrix model. These models are explained in the following sections 3.3.1 and 3.3.2

3.3.1 Outlier Elimination with Fundamental Matrix

The fundamental matrix $\mathtt{F}$ (F-matrix) determines the mapping of a feature point $\mathbf{p}_1$ in camera image $k = 1$ to the epipolar line $\mathbf{l}_K$ in camera image K (cf. Figure 3.9).

Assuming that $\mathtt{A}_1$ is the camera matrix of image $k = 1$, from equation (2.11) follows

$$\mathtt{A}_1 \mathbf{P} \quad = \quad \mathbf{p}_1 \tag{3.31}$$

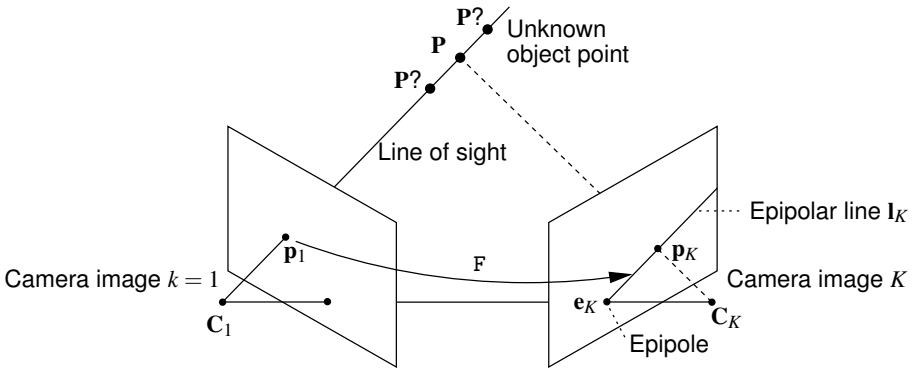


Figure 3.9: Mapping of feature point $\mathbf{p}_1$ of camera image $k = 1$ to the epipolar line $\mathbf{l}_K$ in camera image K by the fundamental matrix $\mathtt{F}$. The epipolar line $\mathbf{l}_K$ is determined by the projection of the line of sight of $\mathbf{p}_1$ into camera image K (from [90]).

for the object point $\mathbf{P}$ located on the line of sight starting at $\mathbf{C}_1$ and passing through $\mathbf{p}_1$. The projection of this line of sight into the camera image K determines the epipolar line $\mathbf{l}_K$. The projection of $\mathbf{C}_1$ into the Camera $\mathtt{A}_K$ determines the epipole $\mathbf{e}_K$, $\mathbf{e}_K = \mathtt{A}_K\mathbf{C}_1$, while the projection of $\mathbf{P}$ into the Camera $\mathtt{A}_K$ is $\mathbf{p}_K$ with $\mathbf{p}_K = \mathtt{A}_K\mathbf{P}$. With equation (3.31), it follows:

$$\mathbf{p}_K = \mathtt{A}_K\mathbf{P} = \mathtt{A}_K\,\mathtt{A}_1^+\,\mathbf{p}_1 \quad , \tag{3.32}$$

where $\mathtt{A}_1^+$ is the Moore-Penrose pseudoinverse of $\mathtt{A}_1$, which is defined by $\mathtt{A}_1\mathtt{A}_1^+ = \mathtt{E}$. Both points, $\mathbf{e}_K$ and $\mathbf{p}_K$ are located on the epipolar line $\mathbf{l}_K$. It follows :

$$\begin{aligned} \mathbf{l}_K &= \mathbf{e}_K \times \mathtt{A}_K\,\mathtt{A}_1^+\,\mathbf{p}_1 && (3.33)\\ &= [\mathbf{e}_K]_\times\,\mathtt{A}_K\,\mathtt{A}_1^+\,\mathbf{p}_1 \quad , && (3.34) \end{aligned}$$

where the vector product $\times$ is determined by the matrix multiplication with

$$[\mathbf{e}_K]_\times = [(e_x, e_y, e_z)^\top]_\times = \begin{bmatrix} 0 & -e_z & e_y \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix} \quad . \tag{3.35}$$

The F-matrix $\mathtt{F}$ is defined as:

$$\mathtt{F} = [\mathbf{e}_K]_\times\,\mathtt{A}_K\,\mathtt{A}_1^+ \quad . \tag{3.36}$$

The required mathematical context between the epipolar line $\mathbf{l}_K$, the F-matrix $\mathtt{F}$, and the feature point $\mathbf{p}_1$ is:

$$\mathbf{l}_K = \mathtt{F}\,\mathbf{p}_1 \quad . \tag{3.37}$$

If a corresponding feature point $\mathbf{p}_K$ is located exactly on the epipolar line $\mathbf{l}_K$, it follows $\mathbf{p}_K^\top \mathbf{l}_K = 0$. With equation (3.37), it follows (cf. equation (9.5) of [43])

$$\mathbf{p}_K^\top \, \mathtt{F} \, \mathbf{p}_1 \quad = \quad 0 \quad . \tag{3.38}$$

The left hand side of equation (3.38) is used to eliminate outliers between the camera images K and 1. The 3×3 fundamental matrix $\mathtt{F}$ is determined by $\beta_{\mathrm{RSC}} = 7$ correspondences [43]. For the estimation of $\mathtt{F}$, equation (3.38) is transformed to the scalar product of two vectors. With $\mathbf{p}_1 = (x_1, y_1, 1)^\top$, $\mathbf{p}_K = (x_K, y_K, 1)^\top$ and $\mathbf{f} = (\mathtt{F}_{11}, \mathtt{F}_{12}, \mathtt{F}_{13}, \mathtt{F}_{21}, \ldots, \mathtt{F}_{33})^\top$, it follows:

$$\mathbf{p}_K^\top \, \mathtt{F} \, \mathbf{p}_1 = (x_K x_1, \; x_K y_1, \; x_K, \; y_K x_1, \; y_K y_1, \; y_K, \; x_1, \; y_1, \; 1) \; \mathbf{f} \quad = \quad 0 \quad . \tag{3.39}$$

This is a linear system of equation in the form $\mathtt{B}\mathbf{f} = \mathbf{0}$, where the 7×9 Matrix $\mathtt{B}$ holds the coordinates of the 14 feature points. This system of equations is solved using the *singular value decomposition* (SVD). Two vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ with corresponding matrices $\mathtt{F}_1$ and $\mathtt{F}_2$ are determined. They belong to the two smallest eigenvalues. The linear combination of $\mathbf{f}_1$ and $\mathbf{f}_2$ span the solution space. The fundamental matrix $\mathtt{F}$ is calculated as $\mathtt{F} = \gamma \mathtt{F}_1 + (1 - \gamma)\mathtt{F}_2$. The value γ is obtained from the side condition $\det(\mathtt{F}) = \det(\gamma \mathtt{F}_1 + (1-\gamma)\mathtt{F}_2) = 0$. This constraint is a polynomial in γ. Thus, one or three solutions are obtained for $\mathtt{F}$.

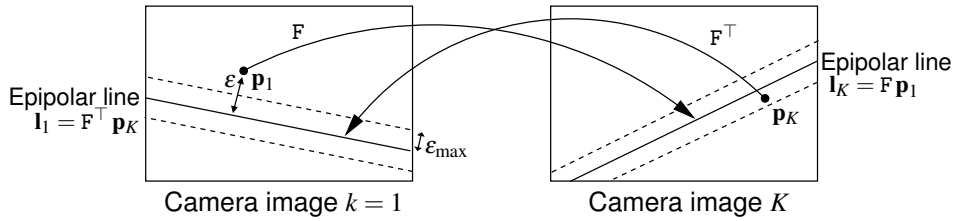


Figure 3.10: The corresponding feature points $\mathbf{p}_1$ and $\mathbf{p}_K$ have a symmetric distance ε to the epipolar lines $\mathbf{l}_1$ and $\mathbf{l}_K$. The correspondences support the estimated F-matrix $\mathtt{F}$ if ε is smaller than $\varepsilon_{\max}$ (from [90]).

After computation of the F-matrix from 7 randomly selected correspondences, the number of supported correspondences is counted. If three solutions result from F-matrix estimation, each of them is considered. For each correspondence, the Euclidean distance $d(\mathbf{p}_1, \mathbf{l}_1)$ between the feature point $\mathbf{p}_1$ and the epipolar line $\mathbf{l}_1$ in image $k = 1$ as well as $d(\mathbf{p}_K, \mathbf{l}_K)$ between $\mathbf{p}_K$ and $\mathbf{l}_K$ in Image K is computed (cf. Figure 3.10). The epipolar line $\mathbf{l}_1$ is (cf. Equation (3.37)):

$$\mathbf{l}_1 = \mathtt{F}^\top \, \mathbf{p}_K \quad . \tag{3.40}$$

The symmetric distance ε is determined by [43]:

$$\varepsilon \quad = \quad \frac{1}{2} \sqrt{d(\mathbf{p}_1, \mathtt{F}^\top \mathbf{p}_K)^2 + d(\mathbf{p}_K, \mathtt{F}\,\mathbf{p}_1)^2} \quad . \tag{3.41}$$

A correspondence supports the F-matrix if the symmetric distance ε is smaller than a threshold $\varepsilon_{\max}$. The value $\varepsilon_{\max}$ is determined by the maximum standard deviation $\sigma_{\max}$ of the location of a feature point which is distorted because of camera noise [90]. In [90], $\varepsilon_{\max} \approx 0.8254\,\text{px}$ is used.

3.3.2 Outlier Elimination with Camera Matrix

After the estimation of 3D object points, an outlier elimination using the camera matrix $\mathtt{A}$ is applied. By multiplication of equation (2.11) with the matrix $[\mathbf{p}]_\times$, it follows:

$$[\mathbf{p}]_\times \mathtt{A}\mathbf{P} = \mathbf{0} \quad , \tag{3.42}$$

because $[\mathbf{p}]_\times \mathbf{p} = \mathbf{0}$. By forming the vector $\mathbf{a} = (a_{11}, \ldots, a_{14}, a_{21}, \ldots, a_{34})^\top$, which consists of the 12 unknown elements of the camera matrix $\mathtt{A}$, it follows:

$$[\mathbf{p}]_\times \mathtt{A}\mathbf{P} = \begin{bmatrix} \mathbf{0}^\top & -\mathbf{P}^\top & y\mathbf{P}^\top \\ \mathbf{P}^\top & \mathbf{0}^\top & -x\mathbf{P}^\top \\ -y\mathbf{P}^\top & x\mathbf{P}^\top & \mathbf{0}^\top \end{bmatrix} \mathbf{a} = \mathbf{0} \quad . \tag{3.43}$$

Each feature point $\mathbf{p} = (x, y, 1)^\top$ leads to this system of three equations in which only two equations are linear independent. Thus, the camera matrix $\mathtt{A}$ is determined by $\beta_{\text{RSC}} = 6$ feature points and their object points. The elements of the camera matrix are obtained by a *singular value decomposition* (SVD) of this linear system of 12 equations.

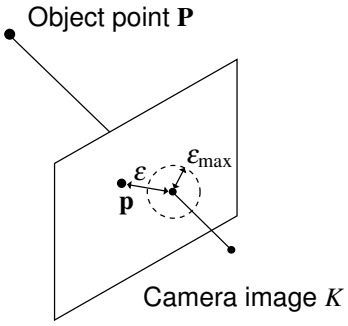


Figure 3.11: The feature point $\mathbf{p}$ supports the estimated camera matrix if the distance ε to the projected object point is smaller than $\varepsilon_{\max}$ (from [90]).

For each feature point $\mathbf{p}_{j,K}$ in camera image K the Euclidean distance to the projected object point $\mathtt{A}_K \mathbf{P}_j$ is calculated:

$$\varepsilon \;=\; d(\mathbf{p}_{j,K},\, \mathtt{A}_K \mathbf{P}_j) \quad . \tag{3.44}$$

A feature point supports the estimated camera matrix $\mathtt{A}$ if its Euclidean distance ε to the projected object point is smaller than a maximum distance $\varepsilon_{\max}$ as shown in Figure 3.11. In [90], a threshold of $\varepsilon_{\max} \approx 3.0\,\text{px}$ is used.

3.4 Scene Estimation Using Incremental Bundle Adjustment

For the scene estimation, a maximum likelihood estimator is derived in [90]. It uses the assumption that the uncertainty of a measured location of a feature point $\tilde{\mathbf{p}}_{j,k} = (\tilde{x}, \tilde{y}, 1)^\top$ is given by a two dimensional Gaussian function with its mean value at the true feature position $\mathbf{p}_{j,k} = (x, y, 1)^\top$. The maximum likelihood estimation of the camera parameters is obtained by minimizing the following cost function:

$$\varepsilon_{\mathrm{BA}} = \sum_k \sum_j d(\tilde{\mathbf{p}}_{j,k}, \hat{\mathtt{A}}_k \hat{\mathbf{P}}_j)^2_{\Lambda_{\mathbf{p}}} \quad , \tag{3.45}$$

with the Mahalanobis distance $d(\cdot)_\Lambda$ [43] and the covariance matrix

$$\Lambda_{\mathbf{p}} \;=\; E\left[\begin{array}{cc} (\tilde{x}-x)^2 & (\tilde{x}-x)(\tilde{y}-y) \\ (\tilde{x}-x)(\tilde{y}-y) & (\tilde{y}-y)^2 \end{array} \right] \quad . \tag{3.46}$$

The estimated values are the camera parameters $\hat{\mathtt{A}}_k$ and the object points $\hat{\mathbf{P}}_j$. The normalization with the number of object points J and the number of images K is called reprojection error (Root Mean Square Error):

$$\varepsilon_{\mathrm{RMSE}} = \sqrt{\frac{\varepsilon_{\mathrm{BA}}}{JK}} \quad . \tag{3.47}$$

The value $\varepsilon_{\mathrm{RMSE}}$ is often used for the evaluation. The minimization of the right hand side of equation (3.45) is called bundle adjustment. The idea of the bundle adjustment is to minimize the distance between the reprojection of an estimated 3D object point $\hat{\mathbf{P}}_j$ and the measured feature point $\tilde{\mathbf{p}}_{j,k}$ for each camera k, in which $\hat{\mathbf{P}}_j$ is visible. The incremental bundle adjustment minimizes (3.45) for each processed image. To limit the computational expense, it is often sufficient to perform a bundle adjustment after a certain number of processed images of the sequence or only after the final image K.

The bundle adjustment cost function is minimized iteratively using the Levenberg Marquardt algorithm [43]. Thus, reasonable initial values are required.

3.4.1 Initialization of the Incremental Bundle Adjustment

Due to the iterative scheme of the bundle adjustment, reasonable initialization values are required to avoid local minima of the cost function (3.45). If the camera is calibrated, which means that the calibration matrices $\mathtt{K}_k$ are known, only the extrinsic camera parameters are estimated. If the camera is uncalibrated, intrinsic and extrinsic camera parameter are determined. In this case, the 12 parameters of the camera matrix $\mathtt{A}_k$ are computed using bundle adjustment. These parameters are called projective camera parameters. The metric camera parameters are derived using a technique called self-calibration [90].

In this work, we focus on the calibrated case. In the calibrated case, for each camera six extrinsic parameters are estimated. These parameters are the projection center $\underline{\mathbf{C}} = (C_x, C_y, C_z)^\top$ and three rotation angles ϕ_x, ϕ_y, ϕ_z which determine the rotation matrix $\mathtt{R}$ as follows:

$$\mathtt{R} = \begin{bmatrix} \cos\phi_z & \sin\phi_z & 0 \\ -\sin\phi_z & \cos\phi_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_x & \sin\phi_x \\ 0 & -\sin\phi_x & \cos\phi_x \end{bmatrix} \begin{bmatrix} \cos\phi_y & 0 & -\sin\phi_y \\ 0 & 1 & 0 \\ \sin\phi_y & 0 & \cos\phi_y \end{bmatrix} \quad .(3.48)$$

For the initialization of the bundle adjustment, appropriate values for camera parameters and object points are required. In this work the incremental bundle adjustment [76] is used. It consists of an initial state and a 3D projection state. The estimator stays in the initial state until the translation between the first and the current camera is sufficiently large. Then, it changes to the 3D projection state and initial parameters $\hat{\mathtt{A}}_k, \hat{\mathbf{P}}_j$ for the bundle adjustment are computed. For the determination of the point of time to change, the *geometric robust information criterion* (GRIC) [94, 95] is used.

Determination of Initial Camera Parameters

Here, we assume the calibrated case, which means that the calibration matrices $\mathtt{K}_1, \mathtt{K}_K$ are known. Furthermore, the GRIC guarantees a sufficiently large translation between the cameras 1 and K. Then, the first camera matrix is chosen as :

$$\mathtt{A}_1 \;=\; \mathtt{K}_1[\mathtt{E}|\mathbf{0}] \quad . \tag{3.49}$$

It follows for the metric camera matrix $\mathtt{A}_K$:

$$\mathtt{A}_K \;=\; \mathtt{K}_K \mathtt{R}_K [\mathtt{E}| - \underline{\mathbf{C}}_K] \quad . \tag{3.50}$$

The objective is now to use the estimated F-matrix to obtain the camera matrix $\mathtt{A}_K$ with equation (3.50). With the definition of the F-matrix (3.36) and equation (2.11), it follows (cf. equation (9.2) of [43]):

$$\mathtt{F} \;=\; \mathtt{K}_K^{-\top} [\underline{\mathbf{T}}_K]_\times \mathtt{R}_K \, \mathtt{K}_1^{-1} \quad , \tag{3.51}$$

with $\underline{\mathbf{T}}_K = -\mathtt{R}_K \underline{\mathbf{C}}_K$. The fundamental matrix corresponding to a pair of calibrated cameras is called *essential matrix* (E-matrix). According to equation (3.51), it has the form:

$$\mathtt{E} \;=\; [\underline{\mathbf{T}}_K]_\times \mathtt{R}_K \quad . \tag{3.52}$$

Then, the $\mathtt{E}$-matrix is calculated as (cf. equation (9.12) of [43]):

$$\mathtt{E} = \mathtt{K}_K^\top \mathtt{F} \, \mathtt{K}_1 \quad . \tag{3.53}$$

Both matrices, $\mathtt{E}$ and $\mathtt{F}$ have rank 2. Two singular values of the $\mathtt{E}$ -matrix are identical and the third is equal to 0 [49]. Therefore, a singular value decomposition can be found as follows:

$$\mathtt{E} = \mathtt{U}\mathrm{diag}[1,\,1,\,0]\mathtt{V}^{\top} \quad . \tag{3.54}$$

Four solutions are possible for the decomposition of the $\mathtt{E}$ -matrix (cf. equation (9.14) of [43]):

$$\mathtt{R}_K = \mathtt{U}\mathtt{M}\mathtt{V}^{\top} \qquad \text{or} \qquad \mathtt{R}_K = \mathtt{U}\mathtt{M}^{\top}\mathtt{V}^{\top} \tag{3.55}$$

$$\underline{\mathbf{T}}_K = \mathbf{u}_3 \qquad \text{or} \qquad \underline{\mathbf{T}}_K = -\mathbf{u}_3 \quad , \tag{3.56}$$

where $\mathbf{u}_3$ is determined as the last column of $\mathtt{U}$. The matrix $\mathtt{M}$ is defined as:

$$\mathtt{M} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad . \tag{3.57}$$

To determine the correct solution, the four possible results are tested using the position of reconstructed 3D object points. A reconstructed object point is only in one of the four solutions in front on both cameras (cf. Figure (9.12) of [43]).

After determining the optimal solutions for $\mathtt{R}_K$ and $\underline{\mathbf{T}}_K$, the metric camera matrix $\mathtt{A}_K$ with $\underline{\mathbf{C}}_K = -\mathtt{R}_K^{\top}\underline{\mathbf{T}}_K$ is derived from equation (3.50).

Determination of Initial Object Points

After the determination of initial camera parameters for the camera images $k = 1$ and $k = K$, initial 3D object points are estimated using triangulation. From equation (3.43), two linear independent equations follow for each camera image and every feature point $\mathbf{p} = (x, y, 1)^{\top}$ corresponding to the 3D object point $\mathbf{P}$:

$$\begin{bmatrix} y\mathbf{a}_3^{\top} - \mathbf{a}_2^{\top} \\ -x\mathbf{a}_3^{\top} + \mathbf{a}_1^{\top} \end{bmatrix} \mathbf{P} = \mathbf{0} \quad , \tag{3.58}$$

with vectors $\mathbf{a}_1^{\top} = (a_{11}, a_{12}, a_{13}, a_{14})$, $\mathbf{a}_2^{\top} = (a_{21}, \ldots, a_{24})$, and $\mathbf{a}_3^{\top} = (a_{31}, \ldots, a_{34})$ derived from the camera matrix. Together with a correspondence of feature points and $\mathtt{A}_1, \mathtt{A}_K$, a linear system of equations is obtained which is solved using the SVD. The solution for the 3D position of the object point is improved by minimizing

$$d(\tilde{\mathbf{p}}_{j,1},\, \mathtt{A}_1\hat{\mathbf{P}}_j)^2 + d(\tilde{\mathbf{p}}_{j,K},\, \mathtt{A}_K\hat{\mathbf{P}}_j)^2 \quad \rightarrow \quad \min \qquad \forall \quad j \tag{3.59}$$

for each object point $\mathbf{P}_j$ using the Levenberg Marquardt algorithm.

Finally, a bundle adjustment with the cameras $\hat{\mathtt{A}}_1$ and $\hat{\mathtt{A}}_K$ as well as all object points $\hat{\mathbf{P}}_j$ is performed. Then, the estimator state changes to the 3D projection state, which is explained in the next section.

3.4.2 3D Projection State of the Incremental Bundle Adjustment

Since 3D object points are already reconstructed in the 3D projection state, they are used together with the new feature points in the current image for the estimation of the camera parameters. Then, the positions of the object points are improved and new object points are generated.

Determination of Initial Camera Parameters

The initial camera parameters $\mathtt{A}_K$ of the current image K are determined using the already known object points $\mathbf{P}_j$. The following cost function is minimized using the Levenberg Marquardt algorithm:

$$\sum_j d(\tilde{\mathbf{p}}_{j,K}, \hat{\mathtt{A}}_K \mathbf{P}_j)^2 \quad \rightarrow \quad \min \quad . \tag{3.60}$$

As initial values, the camera parameters $\mathtt{A}_{K-1}$ of the previous image $K-1$ are used.

Improvement and Reconstruction of Object Points

For each corresponding feature point in the current image K which has an already reconstructed object point $\mathbf{P}_j$ in the previous image, the position of $\mathbf{P}_j$ is improved by minimizing the cost function:

$$\sum_{k=U}^{K} d(\tilde{\mathbf{p}}_{j,k}, \mathtt{A}_k \hat{\mathbf{P}}_j)^2 \quad \rightarrow \quad \min \tag{3.61}$$

using the Levenberg Marquardt algorithm. Here, every camera image k, in which the object point is visible after being selected in image U is used for the optimization. For the initialization, the position of $\mathbf{P}_j$ in image $K-1$ is used.

If the corresponding feature point in the previous image $K-1$ has no valid object point, and the corresponding trajectory has a length larger than V_{new}, it is a candidate for the creation of a new object point. The linear system of equations (3.58) is solved for the candidate object point using SVD. The coordinates are optimized using equation (3.60) with the Levenberg Marquardt algorithm. The candidate point is used as a new 3D object point $\mathbf{P}_j$ if:

$$d(\mathbf{p}_{j,k}, \mathtt{A}_k \mathbf{P}_j) \quad < \quad W_{\text{new}} \qquad \forall \quad k = U, \ldots, K \quad . \tag{3.62}$$

The following parameters are used [90]:

$$V_{\text{new}} = 3, \quad W_{\text{new}} = 1.94 \quad . \tag{3.63}$$

Finally, a bundle adjustment is done for each camera $\hat{\mathtt{A}}_k$, $k = 1,\ldots,K$ and all object points $\mathbf{P}_j$. The initial values of the optimization are taken from the previous bundle adjustment together with the initial camera parameters $\mathtt{A}_K$ and the newly generated object points in the current image K.

4 Improvement of Feature Localization Accuracy

The state of the art subpixel feature localization technique [13, 61] incorporates no shape information of the image gradient signal. The localization scheme minimizes the distance to a function model which approximates the gradient signal of the scale space. Introduced in [13], the function model is chosen as a 3D quadratic [61]. The same feature localization method is used in many other scale invariant feature detectors, e.g. [7, 36, 51, 69].

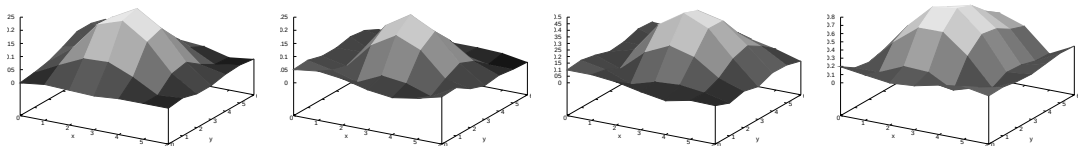
(a) Input image (b) 1st octave, $i = 0$ (c) 2nd octave, $i = 3$ (d) 3rd octave, $i = 6$

Figure 4.1: Input image and the first layer of the first three octaves in the Difference of Gaussians pyramid. The images (c) and (d) are rescaled to the original image size of 800×600 pixels (input image from [70]).

However, it is known that the image gradients surrounding a feature in an image do not have parabolic shape. Moreover, the shape in scale direction is unknown. In the scale space literature [55, 57, 58, 59], features are modeled as Gaussian blobs. These Gaussian blobs should lead to a *Difference of Gaussians* (DoG) shape in the DoG pyramid. In Figure 4.1, the first scales of the first three octaves are shown exemplarily. In each octave, the extrema in these images appear to adopt the DoG shape as shown for some features in Figure 4.2.

While the assumption using a Gaussian approximation function for Harris corners or Canny edges is analytically and experimentally justified in literature [17, 41, 65, 71, 77], a shape assumption for SIFT features has not been considered so far. It is expected that the approximation with a 3D quadratic function leads to suboptimal solutions.

In this chapter, it is shown that the feature localization accuracy of SIFT can be increased using a model for the shape of the image signal in the neighborhood of a feature [18, 19]. The model is motivated by the point spread function of the aperture lens system of a real camera as explained in Section 2.2.3. The spatial neighborhood of a



(a) (b) (c) (d)

Figure 4.2: Selected examples for the gradient signal shape of the neighborhood of a detected feature.

feature in an image is assumed to have Gaussian shape. The shape in scale direction is modeled implicitly by following the signal transfer of the Difference of Gaussians filter.

The approach followed here is the generalization of the Gaussian model for the image signal surrounding a feature point [71, 72, 73, 77] to the detected features in the scale space pyramid. In [71], it is shown that the localization accuracy for a detected image point can be increased by up to 0.025 px for a Canny Edge. As scale invariant features, such as SIFT features, are detected in higher pyramid levels, we expect more gain in accuracy for two reasons: (1) errors of features detected in a higher pyramid level lead to even larger errors in the base layer of the pyramid, and (2) the erroneous signal approximation in scale direction is assumed to influence the magnitude of the spatial localization error. The increase in localization accuracy should lead to an increase in reconstruction accuracy for the application of structure and motion recovery.

In Section 4.1, a systematic error is derived and demonstrated for the SIFT detector. In Section 4.2, the DoG SIFT function model is introduced. The feature localization procedure of SIFT is exchanged with DoG SIFT to eliminate the systematic error. Using images of Gaussian blobs, the accuracy of this model is compared to the accuracy of the original SIFT detector in Section 4.3.

4.1 Analysis of SIFT Feature Localization

The objective of a feature localization procedure is to estimate the true position $\underset{\sim}{\mathbf{n}}$ of a feature. The desired result is the estimated feature position $\hat{\underset{\sim}{\mathbf{n}}} = (\hat{n}_x, \hat{n}_y, \hat{n}_i)^\top$ with minimal estimation error $|\hat{\underset{\sim}{\mathbf{n}}} - \underset{\sim}{\mathbf{n}}|$ using a distance measure $|\cdot|$. In most applications, a high spatial localization accuracy is needed. The estimation error in scale direction $|n_i - \hat{n}_i|$ is of minor importance, because most applications use the computed scale only for the distinctive scale invariant representation of the feature with a descriptor. For the descriptor, a highly accurate localization of the scale layer is not required. However, using a suboptimal model for the scale in the estimation of subpixel and subscale may spoil the results of the subpixel localization, as well.

The feature localization of the SIFT detector [13, 61] consists of two parts (cf. Figure 3.2):

(1) a fullpixel and fullscale detection, which localizes the feature in the scale space with discrete coordinates $\hat{\underset{\sim}{\delta}}(\underset{\sim}{\mathbf{n}}) = (\hat{\delta}(n_x), \hat{\delta}(n_y), \hat{\delta}(n_i))^\top \in \mathbb{N}^3$

(2) a gradient signal approximation scheme resulting in subpixel and subscale coordinates $\hat{\varepsilon}(\underset{\sim}{\mathbf{n}}) = (\hat{\varepsilon}(n_x), \hat{\varepsilon}(n_y), \hat{\varepsilon}(n_i))^\top \in \mathbb{R}^3$ with $-0.5 \leq \hat{\varepsilon}(n_x), \hat{\varepsilon}(n_y), \hat{\varepsilon}(n_i) \leq 0.5$. The proposed approximation function in [13, 61] is a 3D quadratic.

The resulting detected feature position $\hat{\underset{\sim}{\mathbf{n}}}$ is

$$\hat{\underset{\sim}{\mathbf{n}}} = \hat{\underset{\sim}{\delta}}(\underset{\sim}{\mathbf{n}}) + \hat{\varepsilon}(\underset{\sim}{\mathbf{n}}) = \begin{pmatrix} \hat{\delta}(n_x) + \hat{\varepsilon}(n_x) \\ \hat{\delta}(n_y) + \hat{\varepsilon}(n_y) \\ \hat{\delta}(n_i) + \hat{\varepsilon}(n_i) \end{pmatrix} = \begin{pmatrix} \hat{n}_x \\ \hat{n}_y \\ \hat{n}_i \end{pmatrix} \quad , \tag{4.1}$$

while the true feature position $\underset{\sim}{\mathbf{n}}$ is

$$\underset{\sim}{\mathbf{n}} = \underset{\sim}{\delta}(\underset{\sim}{\mathbf{n}}) + \varepsilon(\underset{\sim}{\mathbf{n}}) = \begin{pmatrix} \delta(n_x) + \varepsilon(n_x) \\ \delta(n_y) + \varepsilon(n_y) \\ \delta(n_i) + \varepsilon(n_i) \end{pmatrix} = \begin{pmatrix} n_x \\ n_y \\ n_i \end{pmatrix} \quad . \tag{4.2}$$

In equation (4.1) and (4.2), the scale component of the feature is given in terms of a scale layer numbering with index i.

Assuming that the magnitude of the localization error is lower than 0.5 for each of the three components of $\hat{\varepsilon}(\underset{\sim}{\mathbf{n}}) - \varepsilon(\underset{\sim}{\mathbf{n}})$, with

$$\hat{\varepsilon}(\underset{\sim}{\mathbf{n}}) - \varepsilon(\underset{\sim}{\mathbf{n}}) = \begin{pmatrix} \hat{\varepsilon}(n_x) - \varepsilon(n_x) \\ \hat{\varepsilon}(n_y) - \varepsilon(n_y) \\ \hat{\varepsilon}(n_i) - \varepsilon(n_i) \end{pmatrix} \quad , \tag{4.3}$$

it is determined by its estimated and true subpixel and subscale coordinates only.

We split the localization error into spatial and scale localization error, denoted ξ^{E} and σ^{E}, respectively. The spatial localization ξ^{E} error is defined as:

$$\xi^{\mathrm{E}} = \begin{pmatrix} \xi_x^{\mathrm{E}} \\ \xi_y^{\mathrm{E}} \end{pmatrix} = 2^{n_{\mathrm{OCT}}} \begin{pmatrix} \hat{\varepsilon}(n_x) - \varepsilon(n_x) \\ \hat{\varepsilon}(n_y) - \varepsilon(n_y) \end{pmatrix} \quad , \tag{4.4}$$

where $n_{\mathrm{OCT}} \in \mathbb{N}_0$ denotes the detected octave of the feature. The factor $2^{n_{\mathrm{OCT}}}$ projects the feature position to the ground plane of the image pyramid. Thus, the spatial localization error of a feature increases by this factor if it is detected in a higher octave.

To obtain the standard deviation n_σ corresponding to a feature $\underset{\sim}{\mathbf{n}}$ located in layer n_i, equation (2.22) is used. The scale localization error is defined as:

$$\sigma^{\mathrm{E}} = \hat{\varepsilon}(n_\sigma) - \varepsilon(n_\sigma) = \sigma_0 \cdot ((2^{\frac{\hat{n}_i}{N_{\mathrm{SIFT}}}}) - (2^{\frac{n_i}{N_{\mathrm{SIFT}}}})) \quad . \tag{4.5}$$

For better comparability, the scale layer error $i^{\mathrm{E}} = \hat{\varepsilon}(n_i) - \varepsilon(n_i)$ is used for demonstrating the results. To calculate the scale layer error i^{E} from a given standard deviation n_σ of a feature, equation (2.23) is used:

$$i^{\mathrm{E}} = \hat{\varepsilon}(n_i) - \varepsilon(n_i) = N_{\mathrm{SIFT}} \cdot (log_2 \hat{\varepsilon}(n_\sigma) - log_2 \varepsilon(n_\sigma)) \quad . \tag{4.6}$$

4.1.1 Analytic Analysis of SIFT Feature Localization

The reference feature localization method of SIFT is analyzed using Gaussian features as input data. An univariate, two-dimensional Gaussian input signal $G_\sigma(n_x, n_y)$ with variance σ is defined as follows:

$$G_\sigma(\delta_x, \delta_y) = \frac{1}{2\pi \cdot \sigma^2} \cdot \exp(-\frac{(\delta_x - \varepsilon_x)^2 + (\delta_y - \varepsilon_y)^2}{2\sigma^2}) \quad . \tag{4.7}$$

Here, $(\delta_x, \delta_y) := (\delta(n_x), \delta(n_y))$ is the fullpixel position of the feature point $\underline{\mathbf{n}} = (n_x, n_y)$ and $(\varepsilon_x, \varepsilon_y) := (\varepsilon(n_x), \varepsilon(n_y))$ is its subpixel position.

For this input signal, the output of the *Difference of Gaussians* (DoG) filter $D_\sigma(\delta_x, \delta_y)$ with a distance k between neighboring scales can be described by:

$$D_\sigma(\delta_x, \delta_y) \quad = \quad G_{k\sigma}(\delta_x, \delta_y) - G_\sigma(\delta_x, \delta_y) \quad . \tag{4.8}$$

The Gaussian input signal f with $f(\delta_x, \delta_y) = G_{\sigma_f}(\delta_x, \delta_y)$ and variance σ_f leads to an extremum in the DoG pyramid:

$$\begin{aligned} D_\sigma(\delta_x, \delta_y) &= (G_{k\sigma} * f)(\delta_x, \delta_y) - (G_\sigma * f)(\delta_x, \delta_y) \\ &= G_{\sqrt{k^2\sigma^2 + \sigma_f^2}}(\delta_x, \delta_y) - G_{\sqrt{\sigma^2 + \sigma_f^2}}(\delta_x, \delta_y) \quad . \end{aligned} \tag{4.9}$$

We assume that G_{σ_f} is the function that adopts its extremum in the DoG pyramid at scale σ. To obtain the relation between σ and σ_f, the derivation of $D(\delta_x, \delta_y, \sigma) := D_\sigma(\delta_x, \delta_y)$ at the fullpixel position $(n_x, n_y) = (\delta_x, \delta_y) = (0,0)$ is calculated:

$$D(0,0,\sigma) = \frac{1}{2\pi \cdot (k^2\sigma^2 + \sigma_f^2)} - \frac{1}{2\pi \cdot (\sigma^2 + \sigma_f^2)} \quad . \tag{4.10}$$

For the extremum, it follows:

$$\begin{aligned} \frac{\partial D(0,0,\sigma)}{\partial \sigma} &= \frac{-k^2\sigma}{\pi(k^2\sigma^2 + \sigma_f^2)^2} - \frac{-\sigma}{\pi(\sigma^2 + \sigma_f^2)^2} = 0 \\ \Leftrightarrow \quad \sigma(k^2\sigma^2 + \sigma_f^2) &= k^2\sigma(\sigma^2 + \sigma_f^2) \\ \overset{\sigma \geq 0}{\Leftrightarrow} \quad k^4\sigma^4 + \sigma_f^4 &= k^2\sigma^4 + k^2\sigma_f^4 \\ \Leftrightarrow \quad k^2\sigma^4 &= \sigma_f^4 \quad . \end{aligned} \tag{4.11}$$

Thus, the relation between σ and σ_f is:

$$\sigma = \frac{1}{\sqrt{k}}\sigma_f \quad . \tag{4.12}$$

In the following, the systematic localization error ξ^{E} using a parabolic interpolation for the DoG signal [13] is derived. For simplification here as well as in the following experimental evaluations we set $\varepsilon_y := 0$ and investigate the systematic error ξ_x^{E} in x-direction. As the two dimensional Gaussian distribution is rotationally symmetric, the same behavior can be expected for the systematic error ξ_y^{E} in y-direction. We investigate $D_\sigma(x)$ with:

$$D_\sigma(x) := D_\sigma(x,0) \quad . \tag{4.13}$$

The function $D_\sigma(x-\varepsilon_x)$ which depicts the gradient image $\mathtt{I}$ is approximated with a parabolic function

$$\mathtt{I}_{grad}(x) = a\cdot x^2 + b\cdot x + c \quad . \tag{4.14}$$

This function is shown in Figure 4.3. It adopts its extremum at $-\frac{b}{2a}$. For the approxima-

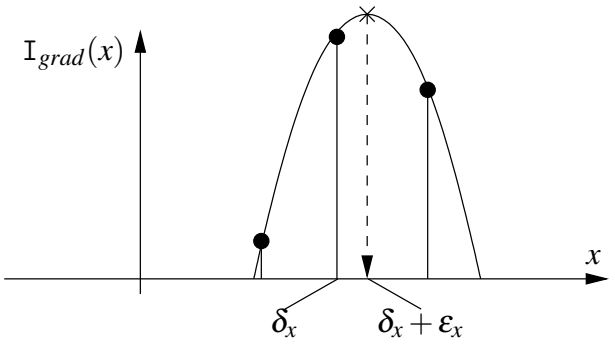


Figure 4.3: Interpolation of $\mathtt{I}_{grad}(x)$ using a parabola. The maximum determines the subpixel coordinate ε_x corresponding to the fullpixel coordinate δ_x (from [71]).

tion of $D_\sigma(x-\varepsilon_x)$ with equation (4.14) the sample points $x_1,\cdots,x_{w_{fit}}$ are used. As SIFT uses a window size of 3 pixels [61], the analysis is done with a window of $w_{fit}=3$. It follows the linear system of equations:

$$A\cdot\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} D_\sigma(x_1-\varepsilon_x) \\ D_\sigma(x_2-\varepsilon_x) \\ D_\sigma(x_3-\varepsilon_x) \end{pmatrix} \quad , \tag{4.15}$$

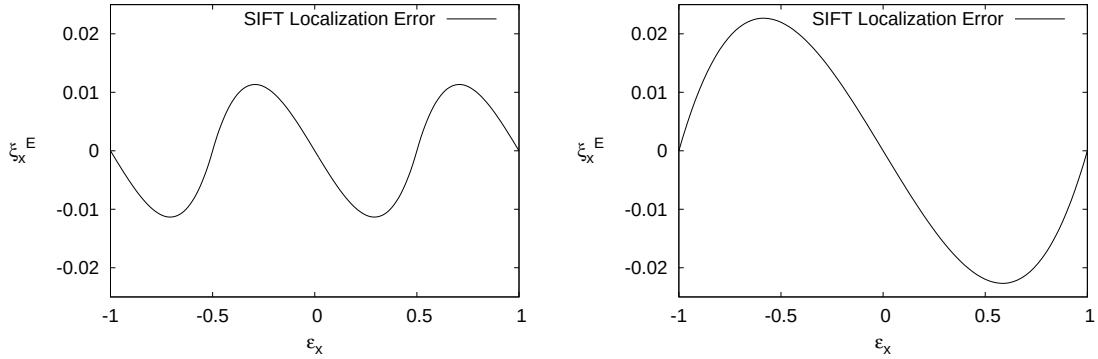
with

$$A = \begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{pmatrix} \quad . \tag{4.16}$$

Assuming a feature point with fullpixel coordinate $\delta_x = 0$ and the sampling points $x_1=-1, x_2=0, x_3=1$, it follows:

$$A = \begin{pmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \Rightarrow A^{-1} = \begin{pmatrix} \frac{1}{2} & -1 & \frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{pmatrix} \quad . \tag{4.17}$$

Now, a, b, and c are determined:



(a) Features in the first octave (b) Features in the second octave

Figure 4.4: Expected systematic error ξ_x^{E} in x-direction of the SIFT localization technique using a parabolic interpolation of the gradient signal. The diagram depicts the expected errors for $\sigma_0 = 1.6$ in the first octave on the left and in the second octave on the right.

$$\begin{aligned}
\begin{pmatrix} a \\ b \\ c \end{pmatrix} &= A^{-1} \cdot \begin{pmatrix} D_\sigma(-1-\varepsilon_x) \\ D_\sigma(0-\varepsilon_x) \\ D_\sigma(1-\varepsilon_x) \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2} & -1 & \frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} D_\sigma(-1-\varepsilon_x) \\ D_\sigma(0-\varepsilon_x) \\ D_\sigma(1-\varepsilon_x) \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2}D_\sigma(-1-\varepsilon_x) - D_\sigma(0-\varepsilon_x) + \frac{1}{2}D_\sigma(1-\varepsilon_x) \\ -\frac{1}{2}D_\sigma(-1-\varepsilon_x) + \frac{1}{2}D_\sigma(1-\varepsilon_x) \\ D_\sigma(0-\varepsilon_x) \end{pmatrix} \quad . \qquad (4.18)
\end{aligned}$$

The systematic error ξ_x^{E} in x-direction is derived as the difference of the extremum of the parabolic function in equation (4.14) and the true subpixel coordinate:

$$\begin{aligned}
\xi_x^{\mathrm{E}}(\sigma, \varepsilon_x) &= \hat{\varepsilon}_x - \varepsilon_x \\
&= -\frac{b}{2a} - \varepsilon_x \\
&= -\frac{1}{2}\frac{D_\sigma(1-\varepsilon_x) - D_\sigma(-1-\varepsilon_x)}{D_\sigma(1-\varepsilon_x) - 2D_\sigma(0-\varepsilon_x) + D_\sigma(-1-\varepsilon_x)} - \varepsilon_x \quad . \qquad (4.19)
\end{aligned}$$

Equation (4.19) shows the systematic error of the parabolic regression in case of a discrete scale, $\varepsilon(n_i) = 0$. This means that the influence of a possibly erroneous function model in scale direction is not considered in equation (4.19). For $\sigma_0 = 1.6$ [61], the error distribution ξ_x^{E} is shown in Figure 4.4 for the first and second octave. As described in [13],

the magnitude of the error is expected to be doubled due to the subsampling of the image from one octave to the next. Resulting from the subsampling, the period length of ξ_x^{E} is doubled from one octave to the next as well.

4.1.2 Experimental Analysis of SIFT Feature Localization

To evaluate the accuracy of the SIFT feature localization, aliasing-free images with Gaussian blobs are synthesized. As we are using rotationally invariant Gaussians, we can limit the test scenario to the spatial localization error ξ_x^{E} in x-direction and the scale localization error σ^{E}. The subpixel value in y-direction is set to $\varepsilon_y := 0$.

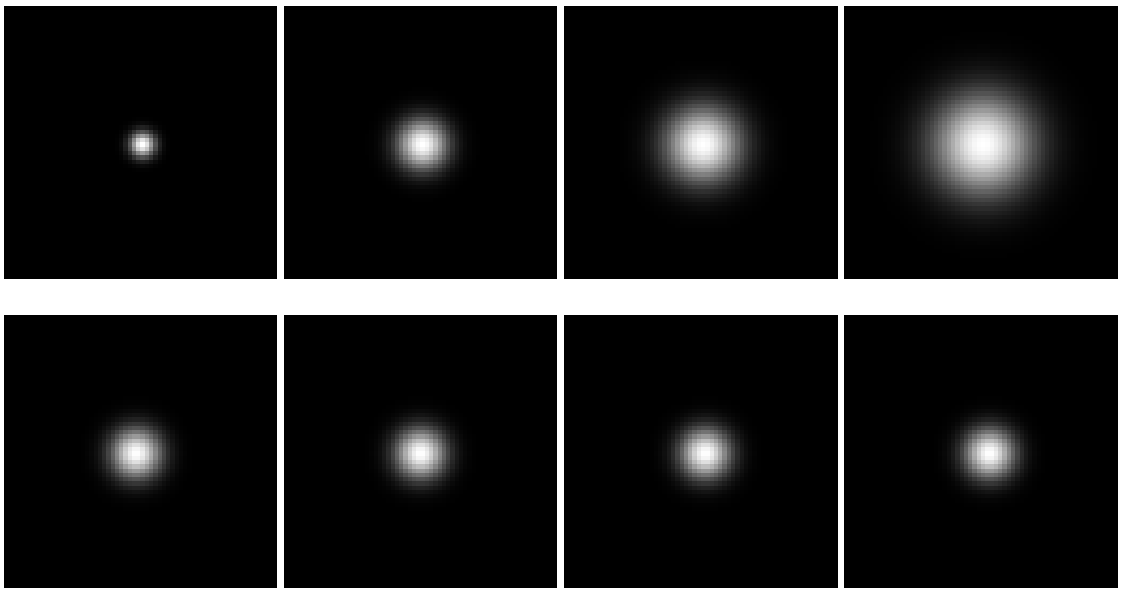


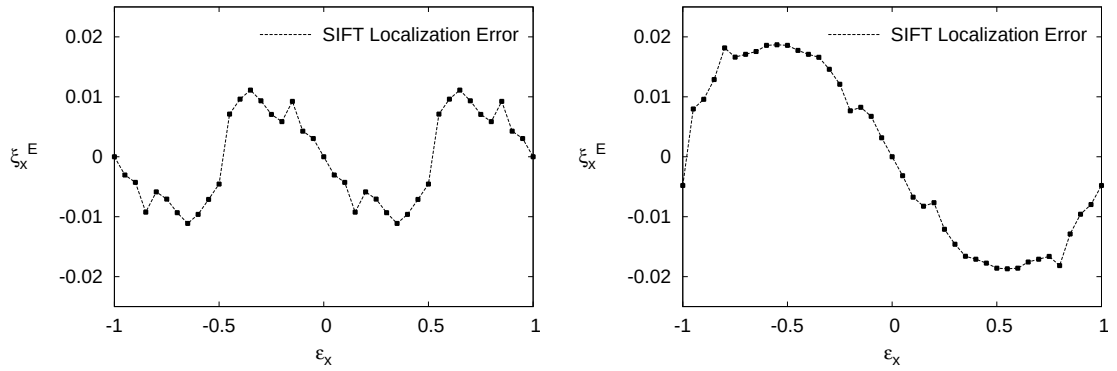
Figure 4.5: Some examples of the test images with varying standard deviation σ_f (top row, $\sigma_f = 2.0, 4.0, 6.0, 8.0$) and varying subpixel position ε_x in x-direction (bottom row, $\sigma_f = 4.0$, $\varepsilon_x = -1.0, -0.5, 0.5, 1.0$). Images from [18].

Each of the images provides one scale space extremum with ground truth subpixel and subscale values. The Gaussian blobs have a varying localization in x-direction and a varying standard deviation σ_f which determines the standard deviation $\sigma = \frac{1}{\sqrt{k}}\sigma_f$ of the detected scale n_i (cf. equation (4.12)). The ground truth localizations ε_x in x-direction are within the interval $[-1.0; 1.0]$ with a step distance of 0.05 px. The used standard deviations σ_f are within the interval $[1.6; 7.9]$ with a step distance of 0.06. With these values, the first two pyramid octaves are covered. The image size is 64×64. Some examples are shown in Figure 4.5. The spatial ground truth feature position $\underline{\mathbf{n}} = (n_x, 0)$ is the center of the Gaussian blob. The ground truth scale n_i is represented by the standard deviation σ_f

of the Gaussian. The relation between n_i and σ_f follows from equation (2.23):

$$n_i = N_{\text{SIFT}} \cdot (log_2(\frac{\sigma_f}{\sigma_0})) \quad . \tag{4.20}$$

The systematic error $\xi_x^{\text{E}} = \hat{\varepsilon}_x - \varepsilon_x$ of the subpixel localization in x-direction and the systematic error of the subscale localization $\sigma^{\text{E}} = \hat{n}_\sigma - n_\sigma$ of the SIFT detector are extracted by detecting the features in the images with SIFT. If more than one feature is detected, the feature in the center is chosen. The results are shown in Figures 4.6 - 4.9. The Figures 4.6 and 4.7 show ξ_x^{E} for selected values of σ_f in the first and second octave. The Figures 4.8 and 4.9 show all results for ξ_x^{E} and i^{E} for the first two octaves.



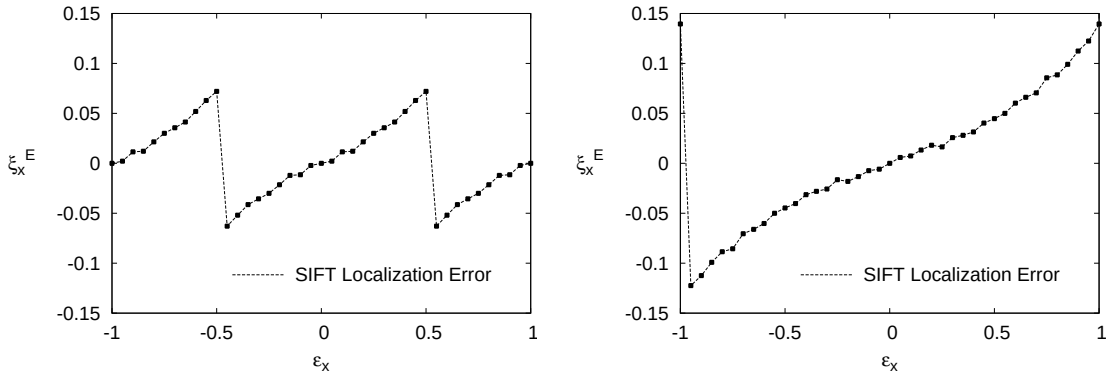
(a) Features in the 1st octave, $\sigma_f = 2.36$ (b) Features in the 2nd octave, $\sigma_f = 4.62$

Figure 4.6: Measured systematic error ξ_x^{E} in x-direction of the SIFT localization technique [46]. The diagram depicts the results for $\sigma_f = 2.36$ in the left diagram and results for $\sigma_f = 4.62$ in the right diagram. The values for σ_f are selected such that the influence of the wrong signal model in scale direction is small.

The Figure 4.6 shows the measured localization error ξ_x^{E} of SIFT for two opportunely selected values for σ_f. The values $\sigma_f = 2.36$ and $\sigma_f = 4.62$ are located nearby a fullscale in the first and the second octave, respectively. The magnitude and shape of the error distribution is similar to the expected distribution (cf. Figure 4.4). As the values for σ do not match a fullscale layer perfectly, small differences to Figure 4.4 are visible.

Conversely, Figure 4.7 shows the measured localization error of SIFT for two values for σ_f, where the influence of the localization error in scale direction is high. Here, the error distributions ξ_x^{E} adopt their maxima at different positions for ε_x compared to the values in Figure 4.6. Their magnitude is about 2.8 times higher.

The four curves shown in Figure 4.6 and in Figure 4.7 are to be found as two slices in the diagram in Figure 4.8 which gives a complete visualization of the localization error ξ_x^{E}. Here, the first two octaves are covered. In the first octave ($\sigma_f < 4.03$), the error is periodic with period length 1 px, because features with a subpixel value $\varepsilon_x > 0.5$ or



(a) Features in the 1st octave, $\sigma_f = 2.6$

(b) Features in the 2nd octave, $\sigma_f = 4.1$

Figure 4.7: Measured systematic error ξ_x^E in x-direction of the SIFT localization technique [46]. The diagram depicts the results for $\sigma_f = 2.6$ in the left diagram and results for $\sigma_f = 4.1$ in the right diagram. The values for σ_f are selected such that the influence of the wrong signal model in scale direction is high. Note, that the y-axis has a different scale compared to Figure 4.6.

$\varepsilon_x < -0.5$ are detected in the left or right adjacent pixel by the fullpixel detection scheme of SIFT. While the systematic error is $\xi_x^E = 0$ on fullpixel positions ($\varepsilon_x = 0$), its magnitude increases in both directions.

In the second octave ($\sigma_f \geq 4.03$), the period length in ε_x direction is doubled because here, the feature is detected in a pyramid level with images of half the size of the original image. However, the systematic error increases significantly compared to the first octave, because the resulting image coordinates are projected into the base image (i.e multiplied by 2 for the second octave). Thus, the error increases. As pointed out in [13], this effect makes accurate subpixel estimation schemes especially important for feature detectors using the scale space, such as SIFT. The results are verified with a second implementation of the SIFT detector in [23].

The Figure 4.9 illustrates the scale layer localization error i^E. It shows the same period lengths compared to ξ_x^E. The magnitude of i^E is similar to the magnitude of the localization error ξ_x^E. But, it does not increase with the octave, because in this diagram, these errors are not referenced to the base layer resolution like ξ^E. However, if the application requires the detected scale layer for a circular region description [29, 70], the corresponding error σ^E (cf. equation (4.5)) will increase with the octave like in ξ^E leading to larger errors in higher octaves. For the application of scene reconstruction, the localization in scale direction is of less importance, because the spatial feature localization is required only. However, the localization error in the scale influences and increases the spatial localization error. This results from the erroneous assumption in [13] that a SIFT feature has parabolic shape in scale direction.

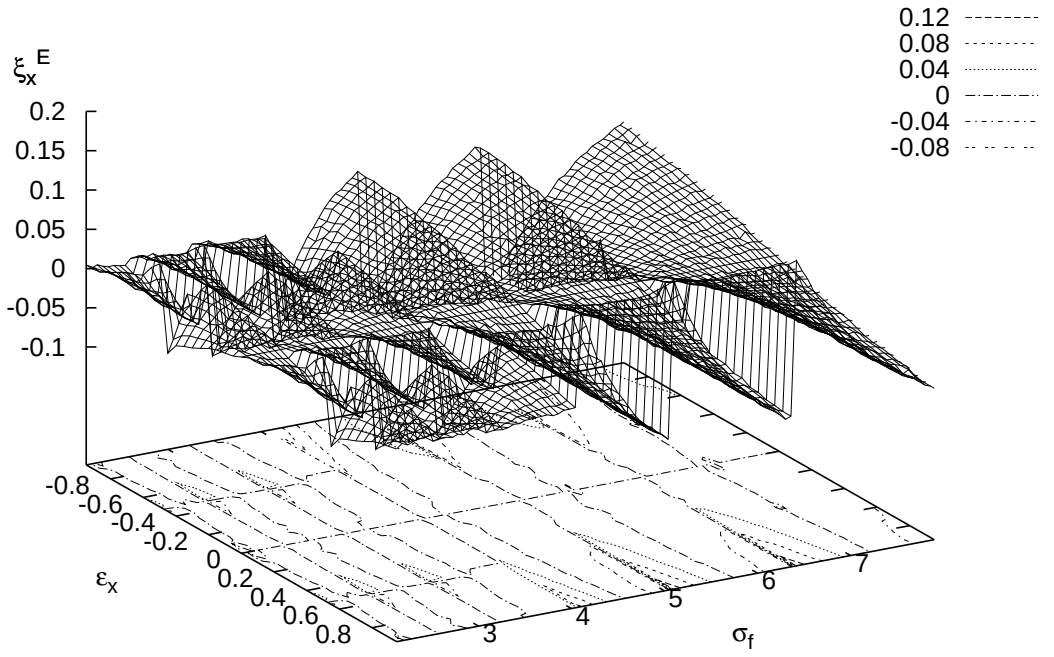


Figure 4.8: Spatial error distribution of the reference SIFT feature localization of the x-position showing the first two octaves. The contour lines in the ground plane indicate the decrease in periodicity by a factor of two from one octave to the next [18].

4.2 Feature Localization Using the Image Signal Model

Scale invariant feature detectors determine a feature point at a specific scale in the scale space. Initially, the feature is localized with fullpixel accuracy as described in Section 3.1.2. The subpixel coordinates are calculated in a second step using the neighboring pixel values and a function which approximates the image gradient signal.

Assuming a camera lens system with a Gaussian transfer function and a scale invariant feature shape [55, 57, 58, 59] as spatial impulse function, the neighborhood of a feature can be described by a Gaussian function. The response of the *Difference of Gaussians* (DoG) filter to this distribution leads to a Difference of Gaussians function.

In order to extract an accurate subpixel and subscale localization of a feature point, the DoG function model is introduced. A member of the function model is determined by a parameter vector $\mathbf{p}$. The parameter vector $\mathbf{p}$ is identified through a regression analysis which minimizes the distances between the sampling points surrounding a feature point in

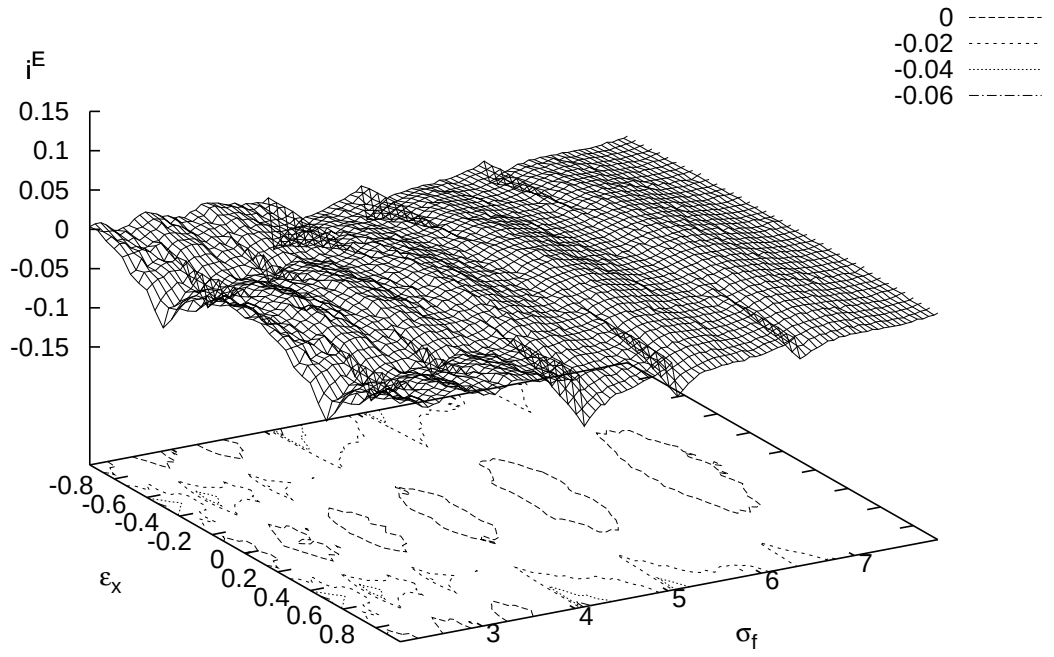


Figure 4.9: Scale layer error distribution of the reference SIFT feature localization of the layer i^{E} showing the first two octaves [18]. Again, the contour lines indicate the decrease in periodicity by a factor of two from one octave to the next (cf. Figure 4.8.

the DoG pyramid and the function model. For the optimization, the Levenberg-Marquardt algorithm is used [43].

The presented localization scheme allows for arbitrary sample point neighborhood sizes. For the detection of features in natural images, the enlargement of the sample point neighborhood, e.g. $5 \times 5 \times 3$, is not beneficial because of the interaction with neighboring features. Experiments have shown that the localization accuracy decreases. But, for applications using images with well separated feature shapes, a larger sample neighborhood stabilizes the localization as shown in [27]. For the results shown here, the regression uses the same $3 \times 3 \times 3$ sample point neighborhood as in the SIFT reference method (cf. Section 3.1.2).

4.2.1 DoG SIFT: Difference of Gaussians Function Model

To build the *Difference of Gaussian* (DoG) function model, we first denote a Gaussian function. To allow for an affine shape of the neighborhood of the feature, an elliptical

region is assumed for its description. A Gaussian function with the covariance matrix $\Sigma = \begin{pmatrix} a^2 & b \\ b & c^2 \end{pmatrix}$ and its determinant $\det(\Sigma)$ can be written as:

$$G_\Sigma(\underline{\mathbf{x}}_\delta) = \frac{l}{\sqrt{\det(\Sigma)}} \cdot e^{-\frac{1}{2}((\underline{\mathbf{x}}_\delta - \underline{\mathbf{x}}_\varepsilon)^\top \Sigma^{-1} (\underline{\mathbf{x}}_\delta - \underline{\mathbf{x}}_\varepsilon))} \quad . \tag{4.21}$$

Here, $\underline{\mathbf{x}}_\delta = (\delta_x, \delta_y)^\top \in \mathbb{N}^2$ is the fullpixel position of the feature point and $\underline{\mathbf{x}}_\varepsilon = (\varepsilon_x, \varepsilon_y)^\top \in \mathbb{R}^2_{[-0.5;0.5]}$ is its subpixel position. The parameters a, b, c define the surrounding elliptical region and l is the peak value parameter.

For the DoG function model, a Gaussian image signal G_Σ as derived in equation (4.21) is used. The DoG function is based on the response of a DoG filter to a Gaussian input:

$$\begin{aligned} D_\sigma(\underline{\mathbf{x}}_\delta) &= l \cdot (G_{\Sigma_\sigma}(\underline{\mathbf{x}}_\delta) - G_{\Sigma_{k\sigma}}(\underline{\mathbf{x}}_\delta)) * G_\Sigma(\underline{\mathbf{x}}_\delta) \\ &= l \cdot (G_{\Sigma_\sigma + \Sigma}(\underline{\mathbf{x}}_\delta) - G_{\Sigma_{k\sigma} + \Sigma}(\underline{\mathbf{x}}_\delta)) \quad , \end{aligned} \tag{4.22}$$

with $\Sigma_\sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$ and the standard deviation $\sigma = \sigma_0 \cdot 2^{\frac{i_n}{N_{\mathrm{SIFT}}}}$ of the current scale i_n.

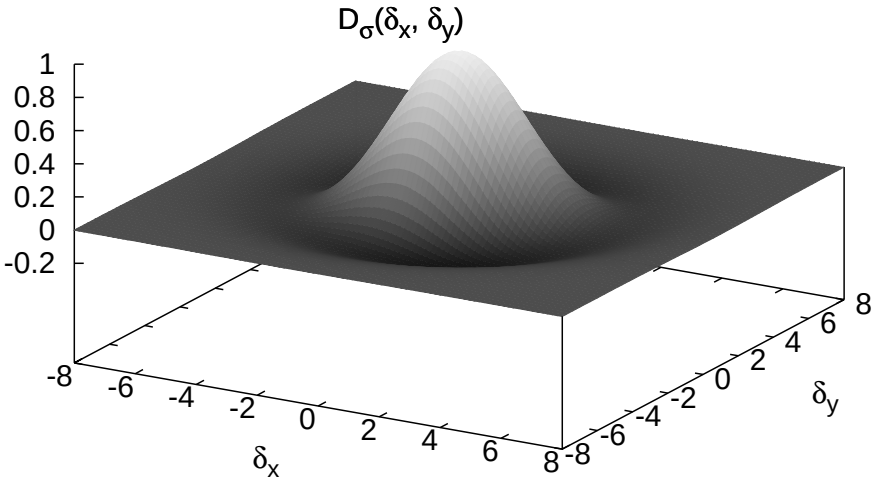


Figure 4.10: Plot of the Difference of Gaussians function model

The DoG function can be described by a six-dimensional parameter vector $\mathbf{p} = (\varepsilon_x, \varepsilon_y, a, b, c, l)$. An example is shown in Figure 4.10. The parameter vector $\mathbf{p}$ is found by minimizing the Euclidean distance $d(\,\cdot\,)$ between the DoG function model $D_\sigma(\underline{\mathbf{x}}_\delta)$ (cf.

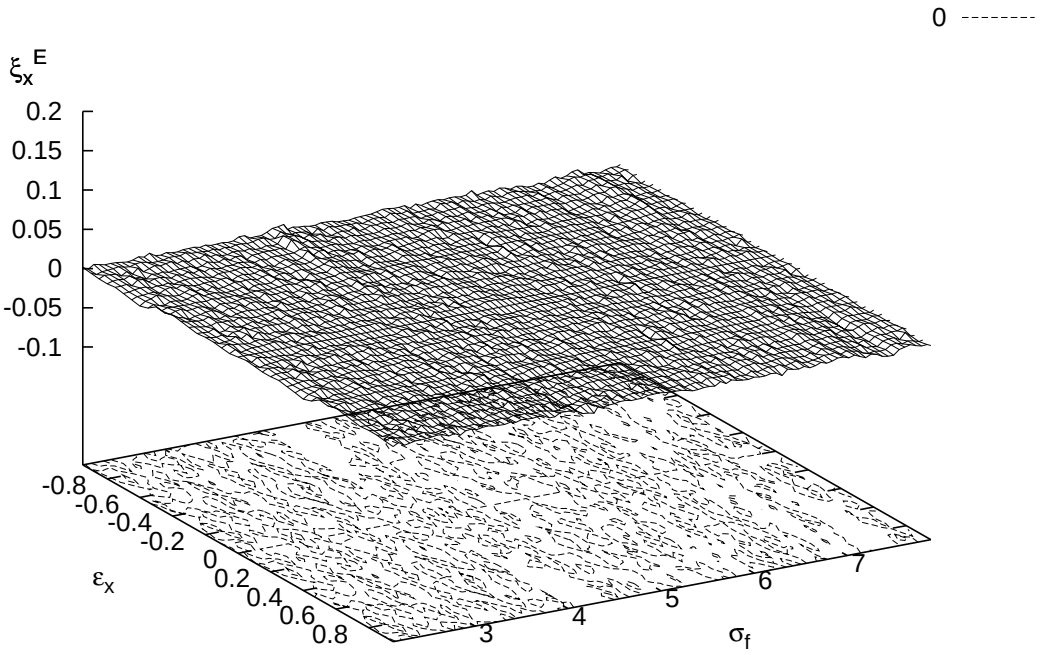


Figure 4.11: Error distribution ξ_x^{E} of the proposed Difference of Gaussians regression DoG SIFT of the x-position showing the first two octaves [18]. The contour lines in the ground plane indicate the small variance of the resulting error near the zero plane.

equation (4.22)) and the 27 sample points in the DoG pyramid surrounding the detected feature $\hat{\underset{\sim}{\mathbf{x}}}_\delta$ with fullpixel and fullscale coordinates. The sum of these distances is called residuum $\varepsilon_{\mathrm{DoG}}$ with:

$$\varepsilon_{\mathrm{DoG}} = \sum_i \sum_y \sum_x d(D_\sigma(\underset{\sim}{\mathbf{x}}_\delta - \underset{\sim}{\mathbf{x}}), D(\underset{\sim}{\mathbf{x}}_\delta - \underset{\sim}{\mathbf{x}})) \quad \rightarrow \quad \min \tag{4.23}$$

for all grid points $\underset{\sim}{\mathbf{x}} = (x,y,i),\ x,y,i \in \{-1,0,1\}$ in the $3\times3\times3$ neighborhood of $\underset{\sim}{\mathbf{x}}_\delta$.

The optimal parameter vector $\mathbf{p}$ in terms of minimizing $\varepsilon_{\mathrm{DoG}}$ is found using the Levenberg Marqardt optimization. The Levenberg Marqardt algorithm is chosen for the optimization because appropriate initial values are known:

$$x_0 = y_0 = i_0 = 0, \quad l = \sigma_0^2 \cdot k \cdot \frac{k+1}{k-1}, \quad a = c = \sigma_0 \cdot \sqrt{k}, \quad b = 0 \quad . \tag{4.24}$$

The localization method using the regression with the function model derived in this Section will be denoted as DoG SIFT. The residuum value $\varepsilon_{\mathrm{DoG}}$ determines the accuracy

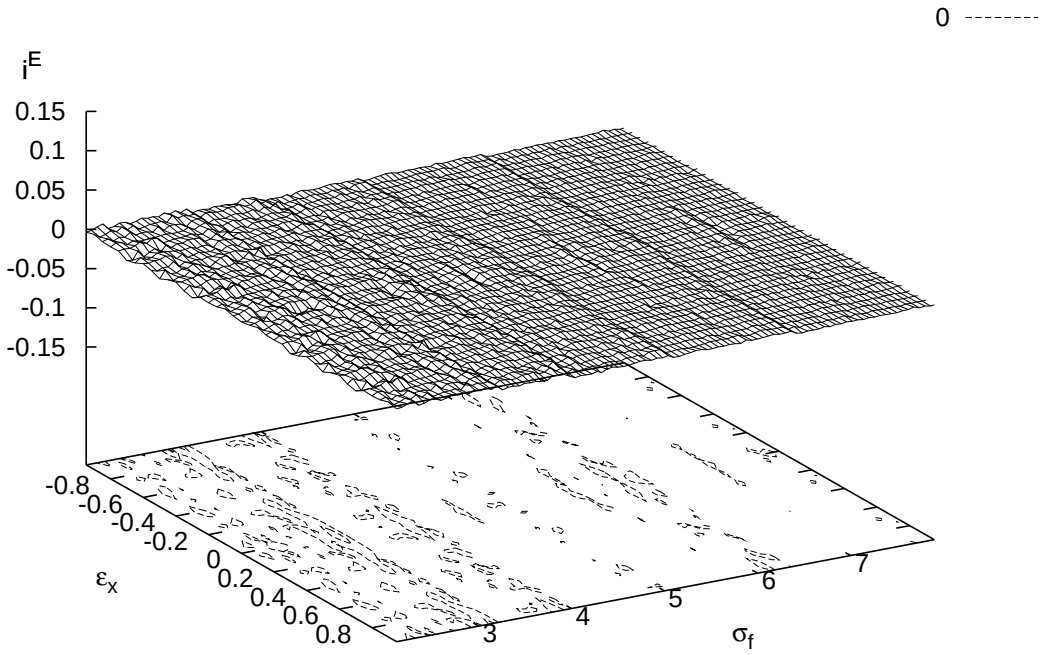


Figure 4.12: Error distribution i^{E} of the proposed Difference of Gaussians regression DoG SIFT of the layer showing the first two octaves [18]. Again, the contour lines in the ground plane indicate the small error.

of the regression. Thus, it provides a measure for the quality of the resulting feature localization. The smaller the residuum $\varepsilon_{\mathrm{DoG}}$, the higher is the probability of obtaining a more accurate feature localization [23]. For the application of feature detection in natural images, the resulting feature list is sorted by their residuum values and the B_{SIFT} features with smallest residuum are selected.

4.2.2 Experimental Analysis of DoG SIFT

The localization accuracy of the DoG SIFT method for the approximation of the gradient image signal is evaluated with the same test scenario as in Section 4.1.2. The evaluation uses the synthesized Gaussian blobs as shown in Figure 4.5. The resulting spatial localization error ξ_x^{E} in x-direction is shown in Figure 4.11. The error distribution shows no dependency on ε_x and σ_f, respectively. Its magnitude is low. The resulting scale localization error regarding the layer error i^{E} is shown in Figure 4.12. This error distribution shows no dependency on ε_x or σ_f. Again, the magnitude is low.

Table 4.1: Maximum errors for the first two octaves of the Gaussian blobs. The maximum scale $n_i = 6$ is limited by the boundaries of the 64×64 image. Features with a σ_f smaller than 2.12 are not detected by the fullpixel localization scheme.

		$Max(\lvert\xi_x^{\mathrm{E}}\rvert)$		$Max(\lvert i^{\mathrm{E}}\rvert)$	
σ_f	n_i	SIFT	DoG SIFT	SIFT	DoG SIFT
2.12..3.98	1..3	0.0721	0.0062	0.0723	0.0111
4.04..7.92	4..6	0.1424	0.0091	0.0829	0.0087

4.3 Comparison of SIFT with Image Signal based Method

In Section 4.1, the reference feature localization SIFT is shown and analyzed in detail. In Section 4.2, the improved feature localization technique DoG SIFT using the image signal model is demonstrated and evaluated. In the following, the results of the evaluations are summarized.

The maximum spatial localization errors $Max(\lvert\xi_x^{\mathrm{E}}\rvert)$ and the maximum scale layer localization errors $Max(\lvert i^{\mathrm{E}}\rvert)$ of SIFT and DoG SIFT for the first two octaves are subsumed in Table 4.1. The systematic error which occurred in the SIFT reference method (cf. Figure 4.8 and Figure 4.9) is eliminated by DoG SIFT. Due to the projection of the spatial localization error to the base layer by the factor of $2^{n_{\mathrm{OCT}}}$, the maximal spatial localization error $Max(\lvert\xi_x^{\mathrm{E}}\rvert)$ increases with the octave by a factor of approximately 2. This is not the case for the scale layer localization error. The maximum errors $Max(\lvert i^{\mathrm{E}}\rvert)$ for the scale is comparable for both octaves.

The magnitude of the localization error (spatial and scale) within an octave is drastically reduced by the proposed DoG SIFT method. The maximum spatial localization error $Max(\lvert\xi_x^{\mathrm{E}}\rvert)$ in x-direction for SIFT is much larger than the maximal localization error found in the analysis for the Canny Edge detector [71]. In [71], maximum error is 0.025 px in maximum. The maximum error for the first octave of the SIFT localization method is 0.0721 px. Interestingly, the large error values for $Max(\lvert\xi_x^{\mathrm{E}}\rvert)$ are obtained for the feature locations $\varepsilon_x \in \{-0.5, 0.5\}$ in the first octave and $\varepsilon_x \in \{-1.0, 1.0\}$ in the second octave. The largest errors for the Canny Edges are located at subpixel positions $\varepsilon_x \in \{-0.25, 0.25\}$ [71].

The maximal error values are verified for the feature location $\varepsilon_x \in \{-0.25, 0.25\}$ for selected values σ_f, where the scale influence is low (cf. Figure 4.6). Nevertheless, the largest localization errors in the SIFT detector are obtained in those positions, where the largest errors for the scale localization occur. These are the subpixel positions $\varepsilon_x \in \{-0.5, 0.5\}$ in the first octave and $\varepsilon_x \in \{-1.0, 1.0\}$ in the second octave(cf. Figure 4.7). This gives the explanation for the increase in error compared to the Canny Edge detector:

the error resulting from a wrong signal approximation in scale direction influences and increases the resulting localization error. The maximum localization error is about 2.8 times higher.

The comparison of SIFT and DoG SIFT using natural image sequences in a structure and motion recovery framework is provided in Section 6.1. An accuracy comparison of DoG SIFT and SIFT using the *repeatability* measure [70] is shown in the Appendix Section 8.

5 Occlusion Handling in Structure and Motion Recovery

Most sequential approaches for structure and motion recovery rely on feature correspondences in consecutive frames [76, 90] as described in Chapter 3. Thus, temporarily occluded scene content causes broken trajectories. A reappearing feature is regarded as a newly appearing 3D point. The resulting estimated new object point and the object point which has been generated before its occlusion adopt different 3D positions. As a consequence, errors accumulate and noticeable drift occurs [28]. The problem of drift arises from foreground occlusion, moving objects, repeated texture, image noise, motion blur, or because tracked points temporarily leave the camera's field of view [93].

Figure 5.1: *Playground* sequence with temporarily occluded scene content resulting from static and moving foreground objects [25]. Feature trajectories discontinue and their features reappear after being occluded. The images show the frames $k = 10, 42, 53, 85$ of the video.

A typical input example is shown in Figure 5.1. In this sequence, the background scene is temporarily occluded by a part of the swing rack and the swinging child. If only consecutive correspondences were used for the scene estimation, many 3D points in the background would be described by ambiguous 3D object point positions, because for each reappearance of a 3D point after its occlusion, a new 3D object point is generated.

In this chapter, it is shown how the reappearing feature points are assigned reliably to the correct previously discontinued trajectories and their 3D object points. This additional information is incorporated into the bundle adjustment and leads to an improved scene reconstruction [20, 25]. Firstly, the reconstruction accuracy is increased. Secondly, the generation of superfluous and erroneous 3D object points is avoided.

The presented correspondence analysis employs the combination of KLT tracking for frame to frame correspondences and SIFT feature matching when wide baseline correspondences are required. In contrast to [28, 93], the assignment of discontinued trajectories is integrated into the sequential camera and scene estimation.

The reassignment of discontinued trajectories not only improves the scene reconstruction, but also provides useful information about the scene. Discontinued and reassigned trajectories are used for the generation of small regions in the images which can be identified as foreground or background. These regions are extracted by reprojecting the reconstructed 3D object points. If a 3D object point is occluded in the current frame, its reprojection leads to an image position which belongs to the foreground.

Assume that a trajectory which discontinued after frame $k-l$ is retrieved in frame k. Then, the two corresponding feature positions in the frames $k-l$, k and the points of the trajectory in the images $k-l-1, k-l-2, \ldots$ are assumed to belong to the background. Candidates for occluded image positions for the 3D object point are found by examining its reprojection in the frames $k-l+1, \ldots, k-1$. But, not every reprojection of an object point into these frames leads to an occluded image position. Therefore, the neighborhood of a reprojected point is compared to the neighborhood of a feature of the non-consecutive correspondence. If these neighborhoods are similar, the reprojected location is not foreground. If they are different, a foreground region is found. The extracted foreground regions have properties which are beneficial for an image segmentation approach: (1) they are located inside the objects and not on the boundaries and (2) each of the reprojection of the 3D object point on a foreground region provides additional color information. These properties are not provided by many state of the art algorithms which extract foreground regions by tracking and clustering features on the foreground [1, 30, 84].

The foreground and background regions collected from all non-consecutive correspondences are used for the generation of a foreground and a background model. The models are generated using the *Gaussian mixture model* (GMM) [78] which is built from the color information of the foreground and background regions, respectively. Here, we make use of a special property of the generated foreground regions. While the color information extracted from a tracked point is approximately the same for each frame, the projections of a background object point into foreground image regions provide new color information for each frame. These image positions do not result from tracked features, but from the projection of scene points *behind* the foreground object. Thus, it is beneficial to collect the colors describing the foreground model from each foreground region of the sequence [24]. We call this method *appearance learning from occlusions* (ALO).

The information about foreground and background resulting from the occlusion of static scene parts is called *occlusion information* [24, 25]. The occlusion information is used to initialize an efficient segmentation algorithm which automatically separates foreground from background in the video. The video segmentation enables visual effect (VFX) creation, such as the automatic occlusion of integrated virtual objects with foreground objects of the captured scene. More applications in VFX creation make use of the video segmentation by applying various filters to the fore- or the background, such as the background blur effect, which focuses the observers attention on the foreground object(s).

In Section 5.1, the feature tracking scheme is presented. It uses the combination of KLT tracking for frame to frame and SIFT feature matching for the non-consecutive correspondences. The extension of the bundle adjustment is explained in Section 5.2. The

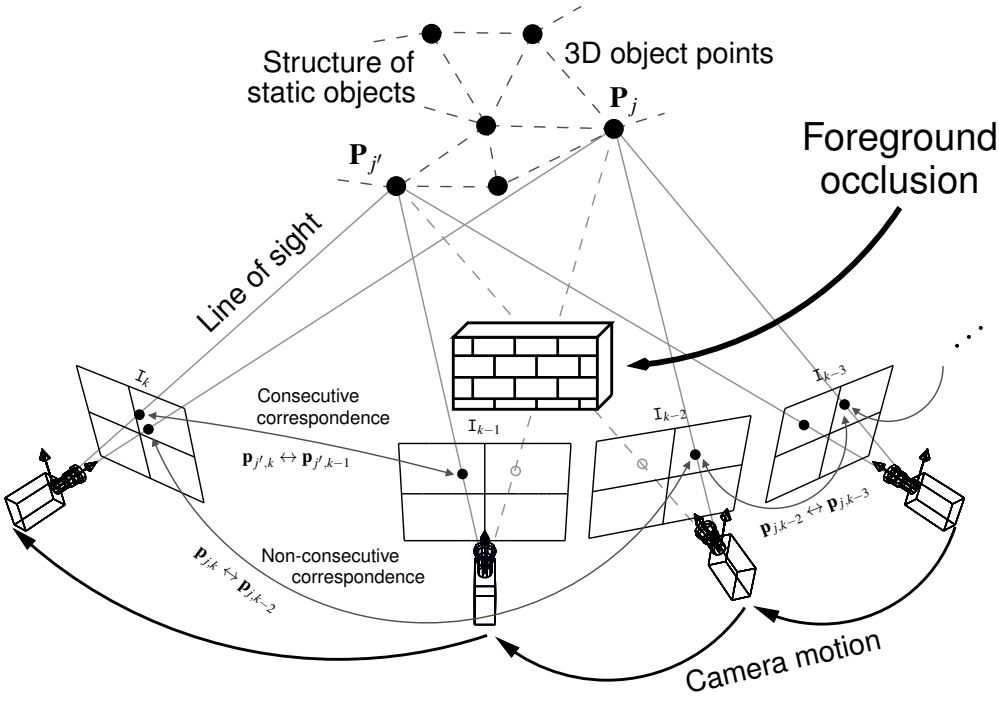


Figure 5.2: Common structure and motion estimation techniques use corresponding feature points in consecutive images only, for example $\mathbf{p}_{j',k} \leftrightarrow \mathbf{p}_{j',k-1}$ [25]. Due to foreground occlusion, trajectories discontinue and the corresponding scene content reappears in a later image. These trajectories are connected using a wide-baseline correspondence analysis, for example $\mathbf{p}_{j,k} \leftrightarrow \mathbf{p}_{j,k-2}$. A real world example is shown in Figure 5.1.

learning of foreground and background appearance from reconnected feature trajectories is shown in Section 5.3. The application of the automatic occlusion of integrated virtual objects is presented in Section 5.4.

5.1 Combination of Feature Matching and Tracking

The combination of frame to frame correspondences and non-consecutive correspondences in the structure and motion recovery approach is illustrated in Figure 5.2. Consecutive correspondences are established using feature tracking while non-consecutive correspondences are obtained by feature matching.

Like in the example in Figure 5.1, the foreground occlusion causes the discontinuation of the trajectory $\mathbf{t}_j = (\mathbf{p}_{j,k-2}, \mathbf{p}_{j,k-3}, \dots)$ in image $k-1$. Assume, that the corresponding 3D object point $\mathbf{P}_j$ has already been reconstructed by the structure and motion recovery algorithm (cf. Section 3.4). Then, the image feature of $\mathbf{P}_j$ reappears in image k and a non-consecutive correspondence $\mathbf{p}_{j,k} \leftrightarrow \mathbf{p}_{j,k-2}$ can be established. The object point $\mathbf{P}_j$

is invisible in image $k-1$ because of occlusion, but the position in image $k-1$ can be estimated by projecting $\mathbf{P}_j$ into this image. We denote this position $\mathbf{p}_{j,k-1}^{\text{invisible}}$ and the visible position in the current image $\mathbf{p}_{j,k}^{\text{visible}}$.

The trajectory $\mathbf{t}_j$ of the 3D object point $\mathbf{P}_j$ in Figure 5.2 is denoted:

$$\mathbf{t}_j = (\mathbf{p}_{j,k}^{\text{visible}}, \mathbf{p}_{j,k-1}^{\text{invisible}}, \mathbf{p}_{j,k-2}^{\text{visible}}, \mathbf{p}_{j,k-3}^{\text{visible}}, \dots) \quad . \tag{5.1}$$

The object point $\mathbf{P}_j$ is visible in the camera images k, $k-2$, and $k-3$ and invisible in the frame $k-1$. The length of a trajectory $|\mathbf{t}_j|$ which contains at least one non-consecutive correspondence is defined as the number of visible trajectory elements:

$$|\mathbf{t}_j| = |\{\mathbf{p}_{j,k}, k = U, \dots, K | \mathbf{p}_{j,k} = \mathbf{p}_{j,k}^{\text{visible}}\}| \quad . \tag{5.2}$$

This definition is consistent with the previous notation of the length of a trajectory in equation (3.19). For establishing a non-consecutive correspondence in $\mathbf{t}_j$, feature matching is utilized. The reconstructed 3D object point $\mathbf{P}_j$ is used for the projection into the following images to estimate the invisible positions of $\mathbf{P}_j$ and to enable guided matching.

To assign the newly appearing feature in image k to the correct previously discontinued trajectory of $\mathbf{P}_j$, a wide-baseline correspondence analysis is required because of a possibly strong viewpoint change between the camera images before and after the occlusion. For this task, the SIFT correspondence analysis is used (cf. Section 3.2.2). The optimal correspondence analysis in consecutive frames remains the KLT feature tracking as presented in Section 3.2.1. The combination of both enables the handling of discontinued trajectories.

The tracking workflow is shown in Figure 5.3. It is extended from the workflow shown in Figure 3.1. In the first image of the input sequence, SIFT features are selected. They are tracked in consecutive images using KLT. The SIFT descriptor is updated for all tracked points. The KLT tracked features are validated and outliers are determined (cf. Section 3.3). The inliers are used for the bundle adjustment leading to the estimation of the current camera $\mathtt{A}_k$ as well as to an update of the reconstructed point cloud $\{\mathbf{P}_j, j = 1, \dots, J\}$. Outliers as well as lost tracks with an already reconstructed and stable 3D object point are stored for a later match with the possibly reappearing feature. In the next frame, SIFT features are selected. They are compared to the stored discontinued trajectories using the descriptors of their last occurrence. In case of a successful match to a feature in image $k-l$, $l > 1$, a candidate for a non-consecutive correspondence is obtained. Candidates for non-consecutive correspondences are validated using outlier elimination and the epipolar geometry of the camera images $\mathtt{A}_k$ and $\mathtt{A}_{k-l}$.

The non-consecutive correspondences avoid the construction of superfluous and therefore erroneous object points. This improves the result of the scene reconstruction. But, because of the more constrained bundle adjustment, the resulting value for the reprojection error in equation (3.47) is likely to increase [53].

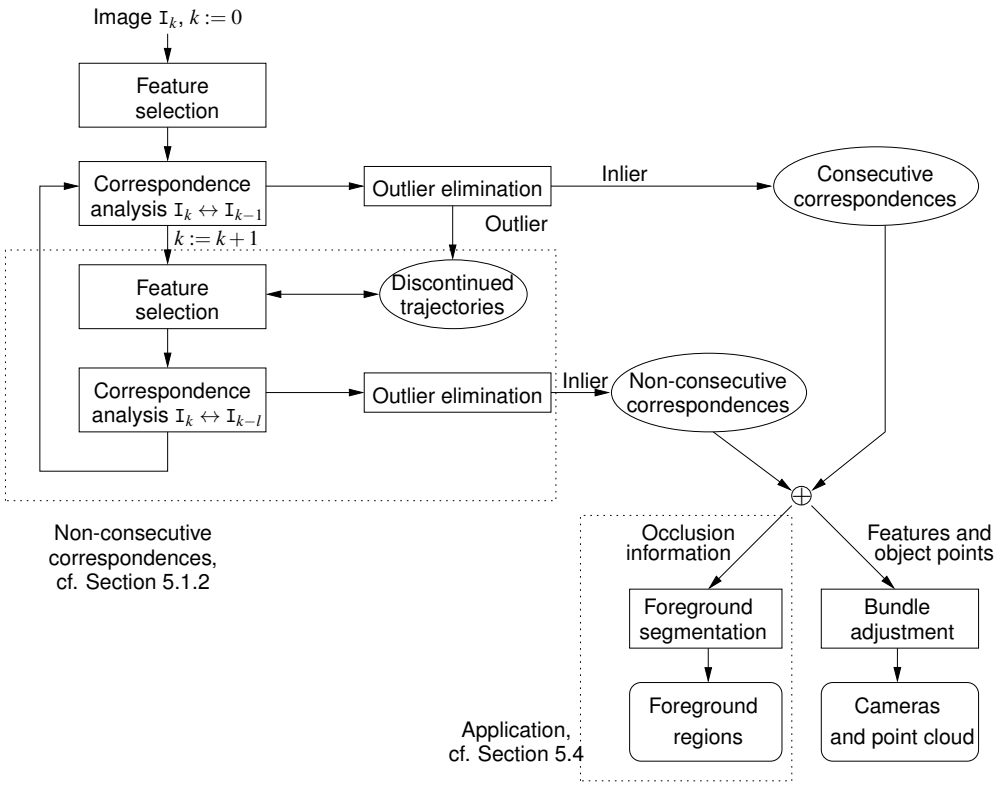


Figure 5.3: Workflow overview: features are tracked in consecutive frames by feature tracking while non-consecutive correspondences are established by feature matching [25]. Features of the current frame k are matched to features of previously discontinued trajectories in the images $k-l$, $l = 2,\ldots,L_{\text{MAX}}$. The bundle adjustment is based on consecutive and non-consecutive correspondences. For the new application as described in Section 5.4, occlusion information is extracted from the non-consecutive correspondences and their trajectories.

5.1.1 Reliable and Accurate Frame to Frame Correspondences

The correspondences in consecutive frames are computed with the KLT tracker (cf. Section 3.2.1). The starting points for the tracking are selected using the SIFT feature selection scheme (cf. Section 3.1.2) because a reappearing feature after its occlusion may be detected in an image with significantly different angle of perspective. To account for the limited localization accuracy of the SIFT features, the improved feature localization method DoG SIFT as presented in Section 4.2 is used. With the current image k, the SIFT descriptor is updated for each of the KLT tracked features. For the validation of a tracked feature, the outlier elimination as explained in Section 3.3 is used together with an initial estimation of the fundamental matrix (cf. Section 3.3.1).

Like in the reference algorithm, the inliers provide the input data for the scene estima-

tion using incremental bundle adjustment. Outliers with an already reconstructed, stable 3D object point are stored in a list, called *discontinued trajectories*. A 3D object point $\mathbf{P}_j$ is regarded as stable if its trajectory $\mathbf{t}_j$ has a length larger than L_{MIN}. Additionally, trajectories of stable 3D object points which are discarded by the KLT tracker because of a large SSD (cf. equation (3.20)) are stored in the discontinued trajectories list. Entries in this list are the candidates for establishing a non-consecutive correspondence in a later image. To limit the required amount of storage for the list, its memory is limited to the discontinued trajectories of the last L_{MAX} images. For each new image k, the discontinued trajectories with its last occurrence in the image $k - L_{\text{MAX}}$ are deleted from the list.

The following parameters are used:

$$L_{\text{MAX}} = 50,\ L_{\text{MIN}} = 4 \quad . \tag{5.3}$$

5.1.2 Reliable and Accurate Non-Consecutive Correspondences

For the reliable correspondence analysis in non-consecutive frames, the SIFT descriptor (Section 3.1.2) is used. Newly appearing features are at first compared to the list of discontinued trajectories. Like in Section 5.1.1, features are selected using SIFT and the improved feature localization method DoG SIFT. For the validation of a feature correspondence, an outlier elimination technique is employed.

Correspondence Analysis in Non-Consecutive Frames

For each image k, the newly detected features are compared to the candidate list of discontinued trajectories using the SIFT correspondence analysis (cf. Section 3.2.2). The possible location of the correspondence of an element $\mathbf{t}_j$ of the candidate list in the current image k is estimated using the reconstructed camera parameters of the previous image $k-1$ and the coordinates of the estimated 3D object point $\hat{\mathbf{P}}_j$.

The distance d_{GUIDED} between the reprojected 3D object point of the discontinued trajectory and the measured feature point $\tilde{\mathbf{p}}_{k,j}$ in the current image is calculated (cf. equation (2.11)),

$$d_{\text{GUIDED}} = d(\hat{\mathtt{A}}_{k-1}\hat{\mathbf{P}}_j, \tilde{\mathbf{p}}_{k,j}) \quad , \tag{5.4}$$

using the Euclidean distance $d(\cdot)$. If d_{GUIDED} is larger than $d^{\max}_{\text{GUIDED}}$, the feature is discarded from the current candidate set. This technique is called guided matching. It decreases the number of possible candidate features for a correspondence.

Due to the uniqueness constraint in the SIFT correspondence analysis determined by the parameter τ_{SIFT} (cf. Section 3.2.2), the guided matching greatly increases the possibility of a successful match because the number of discontinued trajectories may be large. If a correspondence is established, it is validated using an outlier elimination technique (cf. Section 3.3).

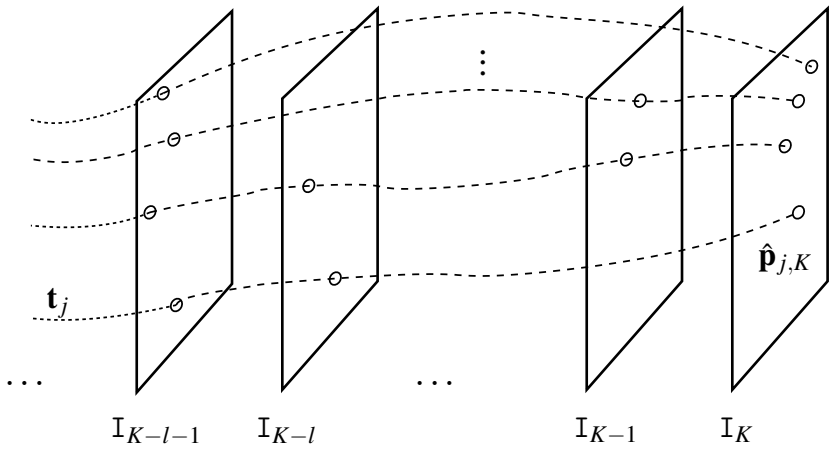


Figure 5.4: Schematic feature tracking situation [20]. The newly detected feature $\hat{\mathbf{p}}_{j,K}$ has no correspondence in the previous frame $K-1$, but a corresponding previously discontinued trajectory $\mathbf{t}_j$ in image $K-l$.

Additionally to the parameters of the SIFT correspondence analysis, the following parameter for the guided matching is used:

$$d_{\text{GUIDED}}^{\max} = 50\,\text{px} \quad . \tag{5.5}$$

The combination of frame to frame tracking and non-consecutive feature correspondences leads to trajectories as shown in Figure 5.4. Trajectories now consist of chains of feature points which are allowed to disappear and reappear arbitrarily.

Outlier Elimination

After the assignment of non-consecutive correspondences for each image $k-l$, $l>1$ to the current image k, outliers are eliminated using the RANSAC approach and the F-matrix mapping model (cf. Section 3.3.1). The outlier elimination is applied to each image $k-l$. To obtain a reliable initial estimation of the F-matrix, the trajectories of the consecutive correspondence are traced to the image $k-l$. The features in this image and their correspondences to image K are used together with the non-consecutive correspondences for the estimation of the initial F-matrix as shown in Figure 5.5. A minimum number $\eta_{\mathtt{F},\min}$ of correspondences, consecutive as well as non-consecutive, is required between image k and $k-l$. Additionally, the camera matrix model (cf. Section 3.3.2) is used to detect outliers in the set of non-consecutive correspondences if its remaining number is larger than $\eta_{\mathtt{A},\min}$.

The following parameters are used:

$$\eta_{\mathtt{F},\min} = 14,\ \eta_{\mathtt{A},\min} = 6 \quad . \tag{5.6}$$

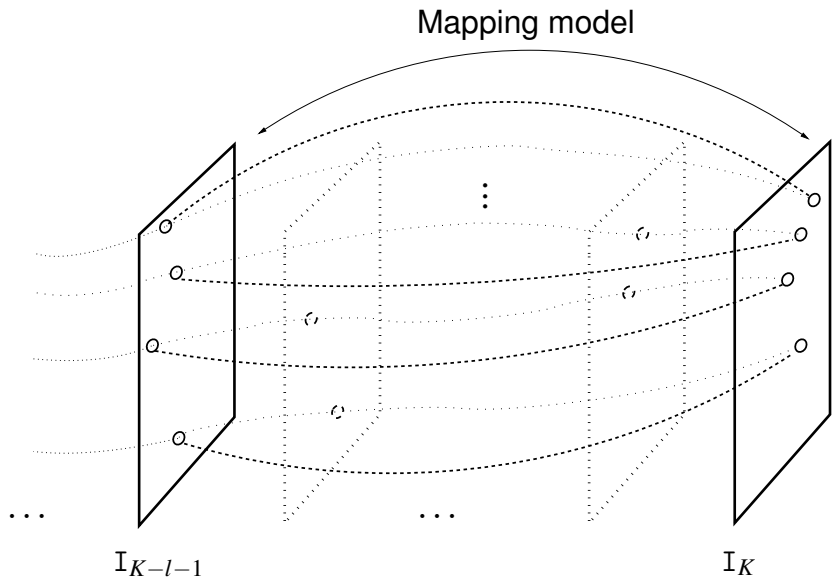


Figure 5.5: Outlier elimination for the non-consecutive correspondences between the images $K-l-1$ and K using the F-matrix mapping model. For the initially estimated F-matrix, the trajectories consisting of consecutive correspondences are used additionally.

5.2 Extension of the Bundle Adjustment

By using the new information of connected object points and trajectories, the incremental bundle adjustment (cf. Section 3.4) is extended. This only affects the bundle adjustment in the 3D projection state as explained in Section 3.4.2, because 3D object point are required to establish the non-consecutive correspondences. While the initialization of the incremental bundle adjustment is not changed, equations (3.60) and (3.61) are now minimized using the connections between non-consecutive frames additionally. This leads to an improved scene reconstruction. In addition, new scene knowledge is induced by trajectories which discontinue because of occlusion. This new scene knowledge is derived in the next section.

5.3 Object Appearance Learning from Occlusions

Often, non-consecutive correspondences provide information about the occlusion of background with foreground objects. A trajectory $\mathbf{t}_j$ with a non-consecutive correspondence from the current image k to a previous image $k-l-1$,

$$\mathbf{t}_j = (\mathbf{p}_{j,k}^{\text{visible}}, \mathbf{p}_{j,k-1}^{\text{invisible}}, \ldots, \mathbf{p}_{j,k-l}^{\text{invisible}}, \mathbf{p}_{j,k-l-1}^{\text{visible}}, \ldots) \quad , \tag{5.7}$$

provides the visibility information of a part of a background object represented by the object point $\mathbf{P}_j$. This object point $\mathbf{P}_j$ is not seen by the camera for the frames $k-1,\ldots,k-l$. Thus, if it is assured that the object point is not seen because of occlusion, the projections of $\mathbf{P}_j$ in the images $k-1,\ldots,k-l$ provide image regions on a foreground object. These regions are the cues for obtaining an automatic object segmentation. By collecting image regions from the whole sequence, the appearance of the foreground objects is learned.

While the neighborhood of projections of an object point on visible image content $\mathbf{p}_{j,k}^{\text{visible}}$ show nearly the same texture for each frame (these points result from tracking), the projections on occluding objects provide new foreground color information for every new frame. As the object appearance does not change throughout the sequence, the foreground color values from each available image can be collected to initialize the representation of the foreground with a *Gaussian mixture model* (GMM) [78].

5.3.1 Occlusion Information

A successfully established non-consecutive correspondence $\mathbf{p}_{j,k} \leftrightarrow \mathbf{p}_{j,k-l-1}$ in the current frame k is a part of a feature trajectory $\mathbf{t}_j$ as denoted in equation (5.7). The object point $\mathbf{P}_j$ of $\mathbf{t}_j$ is invisible in l frames $k-1,\ldots,k-l$. It is visible in the current image k and in some previous images $k-l-1,k-l-2,\ldots$. It may has been invisible several times before. The coordinates of each of the invisible image locations $\hat{\mathbf{p}}_{j,k}^{\text{invisible}}$ of $\mathbf{t}_j$ are estimated as follows (cf. equation (2.11)):

$$\hat{\mathbf{p}}_{j,k}^{\text{invisible}} = \hat{\mathtt{A}}_k \hat{\mathbf{P}}_j \quad . \tag{5.8}$$

The positions $\hat{\mathbf{p}}_{j,k}^{\text{invisible}}$ are used to extract occlusion information which provides the initialization of the automatic foreground segmentation.

The foreground is defined as image regions which occlude the background scene temporarily. The background scene is represented by reconstructed 3D object points. For the occlusion property of a region surrounding a reprojected object point, a verification procedure is required for the invisible part of the trajectory of a non-consecutive feature correspondences, which is (cf. equation (5.7)) $(\mathbf{p}_{j,k-1}^{\text{invisible}},\ldots,\mathbf{p}_{j,k-l}^{\text{invisible}})$. This step is important because the correspondence may be established several frames after the 3D point reappears. In this case, either the reappearing 3D point is not immediately recognized as a feature by the detector, or the correspondence analysis in the previous frames failed due to ambiguities in the image signal (e.g. repeated texture patterns, noise). Furthermore, non-consecutive correspondences are established without occlusion. The main reason for this is motion blur (examples are shown later in Figure 6.16 and 6.17). Thus, a verification of the occlusion property is required.

5.3.2 Verification of Occlusion Property

If the object point $\mathbf{P}_j$ is invisible in the current image k because of occlusion, its reprojection $\hat{\mathbf{p}}_{j,k}^{\text{invisible}}$ belongs to the foreground region in k. However, experiments have shown, that

Figure 5.6: Visualization of the occlusion information of the *Playground* sequence (images $k = 12, 34, 45, 77$) from Figure 5.1: occluded (white) and not occluded (black) object points interpreted as foreground and background regions [25]

non-consecutive correspondences are established without occluded scene content as well. To verify the occlusion property, a similarity constraint between each invisible point of $\mathbf{t}_j$ and the reappearing point $\mathbf{p}_{j,k}^{\text{visible}} = \mathtt{A}_k\mathbf{P}_j$ is evaluated. If the similarity constraint is fulfilled, the object point is deemed not occluded in the camera view. Otherwise, the reprojection is an occluded image position. As similarity measure, the color histogram in a $N_{\text{hist}} \times N_{\text{hist}}$ window around each reprojection of $\mathtt{A}_{k-1}\mathbf{P}_j, \mathtt{A}_{k-2}\mathbf{P}_j, \cdots = \hat{\mathbf{p}}_{j,k-1}^{\text{invisible}}, \hat{\mathbf{p}}_{j,k-2}^{\text{invisible}}, \ldots$ is computed. For the measurement, the Bhattacharyya histogram distance metric is chosen. This metric provides best results for comparing histograms [93] of image regions. The validation with the Bhattacharyya distance also avoids that points which are located on foreground object boundaries are assigned to the foreground. A point on the object boundary would lead to a region which contains pixels from the foreground and the background. These regions would spoil the color model and lead to suboptimal segmentation results. Consequently, the size of the region N_{hist} for the histogram is based on the neighborhood size used for the SIFT descriptor (cf. Section 3.1.2) and is chosen to $N_{\text{hist}} = N_{\text{SIFT}} = 16$ px.

A visualization of the occlusion information is shown in Figure 5.6. The occluded image locations are visualized as white discs, the visible locations of the feature points are black. The diameter of a disc is set to N_{occl}, $N_{\text{occl}} = N_{\text{SIFT}} = 16$ px. These images provide a reasonable initialization of the segmentation procedure as explained in Section 5.4.1.

5.3.3 Color Representation

The state of the art model for the efficient description of an image region by its color values is the Gaussian mixture model (GMM) model [75]. It has been successfully applied to the application of image segmentation in [78]. The color information of the observed image region is represented by a number N_{GMM} of d dimensional Gaussian distributions. As RGB color values are used for the representation, the dimension d is set to 3. Like in [78], the number of models is chosen to $N_{GMM} = 5$.

The occlusion information introduced in Section 5.3.1 is used to obtain a GMM for the

Figure 5.7: *Playground* sequence from Figure 5.1, As shown in the top row, for integrating virtual objects, it is essential to handle foreground occlusions in the composition of virtual and real scenes. The desired result should look like in the bottom row: the integrated virtual objects are occluded accurately and reliably [25].

foreground object(s) from all frames of the sequence. The underlying assumption is that the objects appearance does not change significantly throughout the input image sequence. Then, each of the occluded image positions is used together with their neighborhood to build the GMM of the foreground.

5.4 Application: Integration of Virtual Objects between Scene Elements

An often used technique in movie production is the integration of virtual objects into a video [47]. The perspectively correct integration requires the accurate estimation of the camera parameters for each camera view. A point cloud representation of the observed scene is helpful for the orientation, but not crucial. By using a 3D modeling tool, such as 3D Studio Max® [4], Maya® [3], or Blender [37], synthetic 3D content is added to the reconstructed scene. Then, the virtual scene is rendered and added to the natural image sequence in a step called compositing. An example is shown in Figure 5.7. If the integrated virtual scene elements should be occluded by foreground scene content, a video segmentation is required, which separates the input video into foreground and background (cf. Figure 5.8). This enables the occlusion of the integrated objects in a second compositing step as shown in Figure 5.9. This results in the final video with occluded virtual scene content as shown in Figure 5.7, bottom row.

The task of video segmentation is usually done manually [47] or in a semi-automatic process which requires user interaction [10, 11, 78]. In our approach, the video segmentation results are obtained automatically using the generated occlusion information.

5.4.1 Interactive Video Segmentation

The segmentation of an image is a labeling problem which assigns a label to each pixel. Pixel of the same label form a class in which the elements have coherent properties, e.g. visual similarity. Common approaches for video segmentation separate the images of the video into two temporarily consistent classes, which are interpreted as foreground and background as shown in Figure 5.8. This technique is also called video matting [86].

The *binary* segmentation assigns to each pixel a label $S \in \mathscr{L} = \{0,1\}$, where 0 is interpreted as background and 1 is interpreted as foreground. The binary segmentation can be extended to general alpha mattes, where a smooth transition between the labels 0 (background) and 1 (foreground) is created [45, 78, 86]. For these alpha mattes, the binary segmentation is used as initialization.

Figure 5.8: Binary segmentation of the input image (left) into two classes: Foreground and background, illustrated with the colors white and black, respectively.

State of the art methods for binary segmentation minimize an energy term $E(f)$ consisting of regional and boundary costs [10, 81]. The idea is to determine the optimal segmentation as the minimum of the discrete energy function $E : \mathscr{L}^n \rightarrow \mathbb{R}$:

$$E(x) = \sum_{i \in \mathscr{V}} \varphi_i(x_i) + \sum_{(i,j) \in \mathscr{E}} \varphi_{i,j}(x_i, x_j) \quad , \tag{5.9}$$

where $\mathscr{V}$ corresponds to the set of all image pixels and $\mathscr{E}$ is the set of all edges between neighboring pixels. The label set $\mathscr{L}$ consists of a foreground and a background label. The unary term φ_i is given as the negative log likelihood using a Gaussian mixture model (GMM) model [78], defined by

$$\varphi_i(x_i) = -\log Pr(I_i \mid x_i = S) \quad , \tag{5.10}$$

where S is either foreground or background and I_i describes the feature vector of pixel i. Each GMM, one for the foreground and one for the background, consists of a number of N_{GMM} components. Each component is three-dimensional to represent the RGB color space. The pairwise term $\varphi_{i,j}$ of equation (5.9) takes the form of a contrast sensitive Ising model [52] and is defined as:

$$\varphi_{i,j}(x_i, x_j) = \gamma \cdot [x_i \neq x_j] \cdot \exp(-\beta \|I_i - I_j\|^2) \quad . \tag{5.11}$$

Here, $[\cdot]$ denotes the indicator function. The indicator function is 1 if its argument is true and 0 otherwise. The parameter γ weights the impact of the pairwise term and β is determined by the distribution of noise among all neighboring pixels. It has been shown that the energy function (5.9) is submodular and can be represented as a graph [10]. Represented as a graph, the minimum cut minimizes the given energy function. With this technique, it is ensured to reach the global minimum of the cost function [10, 11, 12] which represents the optimal binary segmentation for two labels. Furthermore, the globally minimum cut can be computed on any graph. Thus, the approach is applicable for data with higher dimensionality, e.g. a 3D graph. An example for a 3D graph is the three dimensional volume of a video [10, 81].

Figure 5.9: Combination of input and augmented sequence (cf. Figure 5.7 using the segmentation in Figure 5.8)

Due to the complexity of the automatic separation of foreground and background, practical approaches to binary segmentation [10, 78] allow for user interaction. Usually the user guides the algorithm by marking the desired foreground and background regions with a bounding box [78] or some strokes [10]. However, many user strokes are required to obtain a result of good quality as illustrated in Figure 5.8 on the right.

We use the energy minimization approach as proposed in [10]. The GMM's are computed from the RGB color space as presented in [78] using N_{GMM} mixture components. To reduce the computational expense, the graph sparsification as proposed in [80] is used. This technique contracts the nodes of the graph while preserving the global minimum of the cost function (5.9). To reduce the computational expense for the 3D graph, the grouping of pixels to regions called *superpixels* is incorporated [81]. The generated superpixels are designed for the image segmentation approach using graph cuts. The pixels which

belong to one superpixel are assigned to the same label. This assumption decreases the complexity of the graph significantly.

For the energy minimization, the following parameters are used [10, 78]:

$$N_{\text{GMM}} = 5,\ \gamma = 60 \quad . \tag{5.12}$$

5.4.2 Automatic Video Segmentation

The workflow of the automatic video segmentation procedure is shown in Figure 5.10. The extracted occlusion information as described in Section 5.3.1 and illustrated in Figure 5.6 provides the initialization for the graph cut based segmentation. The automatically generated strokes exchange the manually drawn strokes given by the user [10].

From the regions determined with these foreground and background strokes the Gaussian mixtures models are generated. Then, the graph representation is built by computing the regional and boundary costs from the images using equations (5.10) and (5.11), respectively. Finally, the minimum cut is computed which determines the resulting segmentation of the corresponding image(s).

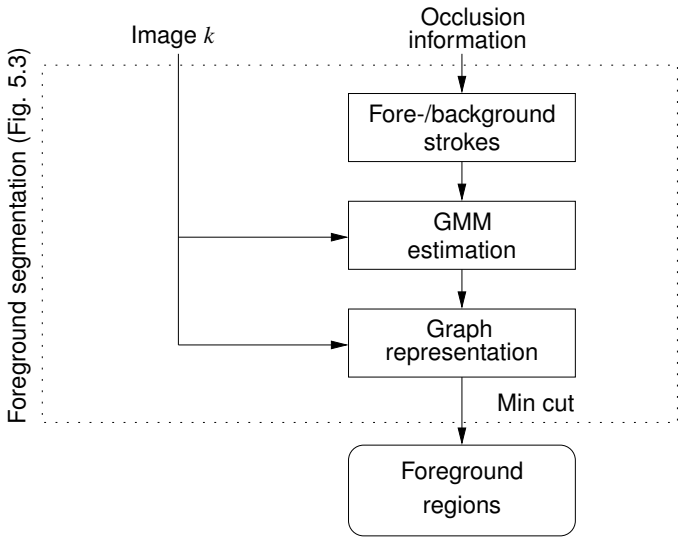


Figure 5.10: Foreground segmentation in detail (refer to Figure 5.3): The occlusion information of the current frame k provides strokes associated to foreground or background. Their Gaussian mixture model (GMM) is obtained by extracting the corresponding color information of image k [26].

The proposed method for *appearance learning from occlusions* (ALO) (c.f. Section 5.3.3) can be used for the 2D grid segmentation for each image separately as well as for the 3D grid segmentation. The 3D grid segmentation treats the image sequence as a

single 3D volume and minimizes the energy function for the whole sequence. This leads to an increased computational expense [10], but to more consistent result for the video segmentation. To reduce the computational expense two approaches are incorporated [80, 81], which reduce the complexity of the graph and, therefore the computational time. Both of them are applicable to any graph cuts based energy minimization algorithm. In contrast to existing approaches, the information required for initializing the energy minimization is derived automatically from the extracted occlusion information.

6 Experimental Results

In this chapter, the presented approaches are evaluated with standard test sequences and self created natural image data in the framework of structure and motion recovery as described in Chapter 3.

The results are grouped in the following test scenarios:

1. Accuracy evaluation (Section 6.1): the proposed feature localization method DoG SIFT is compared with the reference localization method of SIFT. For the evaluation, the reprojection error is used as error measure.

2. Feature trajectory retrieval evaluation (Section 6.2): the proposed feature trajectory retrieval (FTR) is compared with the reference SAM recovery regarding the lengths of the resulting trajectories and the accuracy of the reconstruction.

3. Occlusion information evaluation (Section 6.3): the reliability of the proposed generation of occlusion information is demonstrated based on the number of connected tracks.

4. Application demonstration (Section 6.4): the application of integrating virtual objects between scene elements is demonstrated using natural image sequences.

6.1 Accuracy Evaluation of Feature Localization

The new approach for the feature localization of SIFT features, called DoG SIFT, has been presented and validated with synthetically constructed Gaussian blobs in Section 4.2 and 4.2.2. To evaluate the usability of the presented technique, DoG SIFT is compared with the reference SIFT localization method using natural image sequences and the application of structure and motion (SAM) recovery.

This section contains accuracy evaluations in terms of the (SAM) recovery setup. An accuracy comparison of DoG SIFT and SIFT using the *repeatability* measure [70] on benchmark data is shown in the Appendix Section 8. The repeatability measure incorporates the circular or elliptical shape of a scale invariant feature by computing the *overlap error* [70] between corresponding features.

6.1.1 Input Data and Experimental Setup

To ensure the compatibility of the results, official data of high spatial resolution is used [1]. Each of the image sequences contains a mostly static scene captured by a Canon D60 digital camera. The sequences consist of still images corrected for radial distortion with ground truth data computed from laser scanning (LIDAR) [89]. Together with the images, the intrinsic camera parameters are provided by the authors of [89]. Two examples are demonstrated in this section. The *Fountain* sequence and the *Herzjesu* sequence. The camera positions are located on a nearly circular path around the captured object. Example images are shown in Figure 6.1(a) and in Figure 6.1(b).

(a) *Fountain* sequence (3072 × 2048, 9 frames)

(b) *Herzjesu* sequence (3072 × 2048, 13 frames)

(c) *Elefant* sequence (3456 × 2304, 36 frames)

Figure 6.1: Example frames of the publicly available input sequences *Fountain* (a), *Herzjesu* (b), and our sequence *Elefant* (c).

An additional sequence is captured by a Canon EOS 350d camera on a tripod. The camera observes a scene on a turntable. The camera captures an image after each turn of 5°. For the evaluation, 36 images are used, resulting in a 180° turn. Like in the first

[1] `http://cvlab.epfl.ch/~strecha/multiview/denseMVS.html`

two sequences, the intrinsic camera parameters are determined in a preprocessing step. For the *Elefant* sequence, the intrinsic parameters are computed using the Tsai calibration method with a calibration pattern [97]. The estimated radial distortion coefficients compensate the radial distortion in the images. Example images of this sequence are shown in Figure 6.1(c).

For each of the three sequences, the intrinsic camera parameters are fixed. A reasonable threshold for the guided matching is chosen to obtain as much feature correspondences as possible for the accuracy evaluation.

For the evaluation of the localization approaches, the structure and motion recovery algorithm as explained in Section 3 is used. In this experiment, the SIFT framework for extracting feature points and correspondences is selected. For the comparison of SIFT and DoG SIFT, only the feature localization methods are exchanged. Both methods use the same fullpixel feature locations selected in the scale space pyramid as input. The image feature positions are then refined by the reference localization of SIFT and the proposed localization method DoG SIFT, respectively. Although the detected fullpixel positions are equal for both methods, some features which are rejected by the SIFT subpixel localization are considered valid by DoG SIFT and vice versa. This is mainly due to the slight localization change resulting in different curvature and contrast values (cf. Section 3.1.2). Some feature candidates are discarded by SIFT or DoG SIFT because the iterative optimization of the localization does not converge.

The remaining structure and motion estimation pipeline in the experiment is identical for both compared methods: correspondences are established by the SIFT descriptor and guided matching with an appropriate search range. Due to the large baselines between the images, the search range of the guided matching has to be set to a relatively large value. The selected search regions are 500×500 pixels for *Fountain*, 800×800 pixels for *Herzjesu*, and 200×200 pixels for *Elefant*.

For the scene reconstruction and camera motion estimation, the fundamental matrix is computed using RANSAC with very many iterations $N_{\text{RSC}} = 10^6$ and $R_{\text{RSC}} = 1.0$ (cf. equation (3.30)) to guarantee invariant results. Outliers are eliminated by thresholding the epipolar distance $\varepsilon_{\max}$. If the distance ε (cf. equation (3.41)) of a corresponding feature to the epipolar line is above $\varepsilon_{\max}$, the correspondence is regarded as an outlier and eliminated from the set of correspondences. Initial camera parameters are estimated after a sufficiently large translation of the camera as explained in Section 3.4.1. As the neighboring camera in these sequences show a large baseline, the initial camera parameters are estimated immediately after the first three images are processed (cf. Section 3.4.2). The final camera parameters and object points are obtained by minimizing the bundle adjustment cost function as explained in Section 3.4. The value $\varepsilon_{\text{RMSE}}$ resulting from equation (3.47) gives the reprojection error of a feature. The resulting reprojection error of a reconstructed scene is dependent on the number of constraints used in the bundle adjustment. More constraints restrain the bundle adjustment, and the reprojection error increases [53]. In this experiment, the number of constraints is mainly determined by the number of reconstructed object points. Thus, the evaluation with the reprojection error

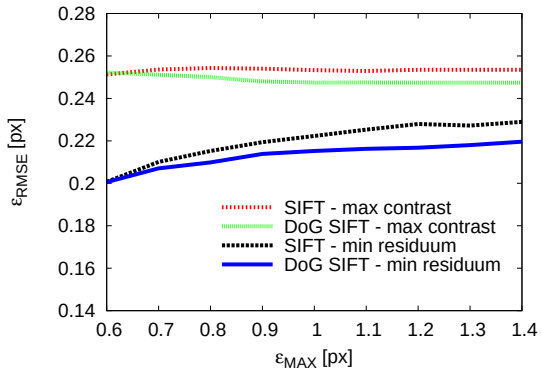


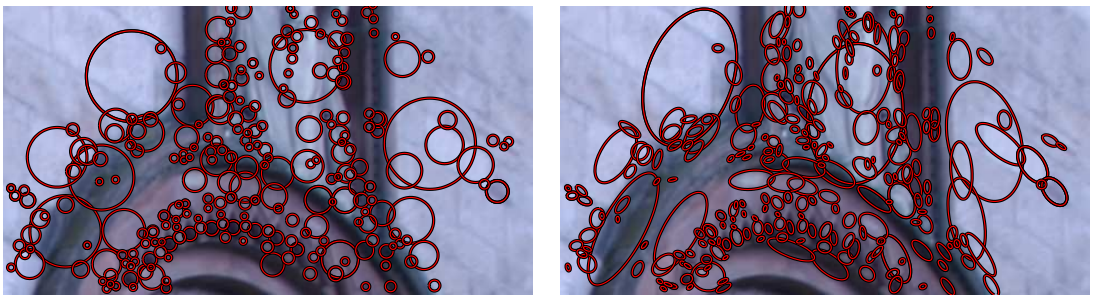
Figure 6.2: Comparison of the results obtained if the feature selection criterion is changed from maximum contrast to minimum residuum (*Herzjesu* sequence, 3000 feature points). The reprojection error is significantly lower for the feature selection using a minimum residuum.

has to be related to the resulting number of object points.

In this experiment, no covariance information is used in the bundle adjustment, because the SIFT reference method only provides features of circular shape and no reasonable accuracy measure for the detected features. Thus, the distance $d(\cdot)$ in equation (3.45) is the Euclidean distance.

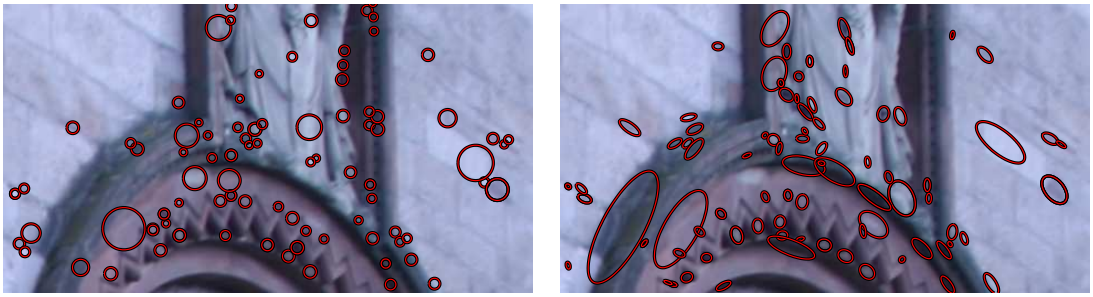
An important issue is the selection of a subset of all automatically detected features in an image. In [5, 62], the contrast values $|D(\underset{\sim}{\mathbf{n}})|$ are used to select the best B_{SIFT} features (cf. Section 3.1.2). The authors assume that the quality of a feature increases with its contrast value. Thus, features with largest contrast are selected. In Section 4.2.1, the residuum value resulting from equation (4.23) is proposed as quality measure for a DoG SIFT feature. The corresponding residuum value for a reference SIFT feature is derived from equation (3.8). As shown in the example in Figure 6.2, this characteristic for feature selection leads to smaller reprojection errors for SIFT as well as for DoG SIFT. More experiments show that better results are obtained when using the residuum feature selection criterion for both methods SIFT and DoG SIFT. An example in Figure 6.3 showcases the resulting features for a small image part of *Herzjesu*. About 300 are detected (top row of Figure 6.3). The selected 100 features are illustrated in the bottom row.

To achieve a fair comparison regarding the reprojection error, for both methods the same number of features are selected in each image. For the evaluations in Figures 6.4-6.6, the features are sorted by increasing residuum values for SIFT and DoG SIFT and the first B_{SIFT} features are chosen.



(a) 292 points detected by SIFT

(b) 299 points detected by DoG SIFT



(c) The best 100 points of SIFT

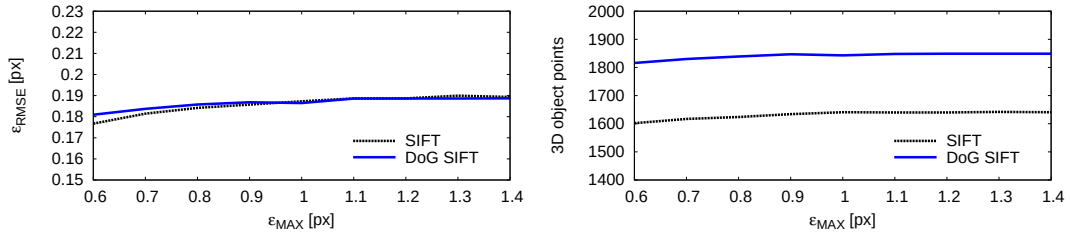
(d) The best 100 points of DoG SIFT

Figure 6.3: Comparison of detector results using a part of an image of the *Herzjesu* sequence. The elliptic shape of the features detected by DoG SIFT show a reasonable structure. Some features which are detected by SIFT are not detected by DoG SIFT and vice versa as shown images (a) and (b). The best 100 points selected with the minimal residuum criterion can be located on significantly different positions for the two approaches which is shown in the images (c) and (d).

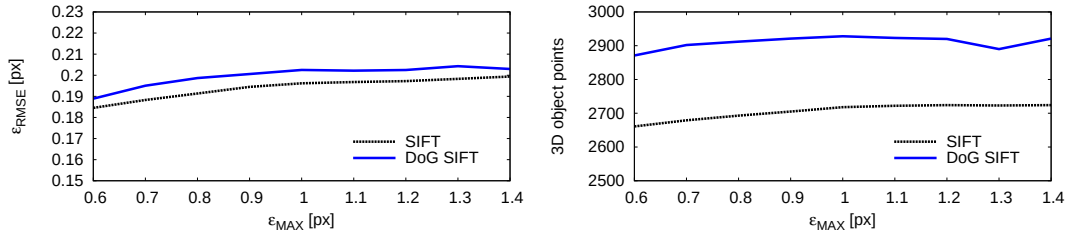
6.1.2 Results

The results for the accuracy evaluation using the structure and motion recovery setup are demonstrated in Figure 6.4 - Figure 6.6. The diagrams on the left show the reprojection error $\varepsilon_{\text{RMSE}}$, while the diagrams on the right show the number of reconstructed 3D object points from the sequence. On the x-axis, the threshold for the maximal epipolar distance ε_{max} is varied. The maximal epipolar distance determines the threshold for the removal of outliers from the structure and scene estimation. The larger the ε_{max}, the more frame to frame correspondences are used for the estimation. If ε_{max} is low, only the more accurate feature correspondences are used. It follows that the reprojection error usually increases with increasing ε_{max}. However, the number of reconstructed object points has to be taken into account, because a lower reprojection error must not lead to a better reconstruction if the number of reconstructed object points is smaller.

The resulting diagrams for the *Fountain* sequence (Figure 6.4, 2000 and 3000 detected

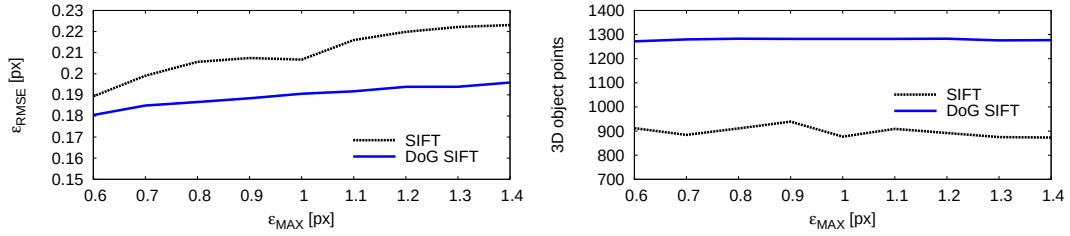


(a) *Fountain* sequence, 2000 detected points per image

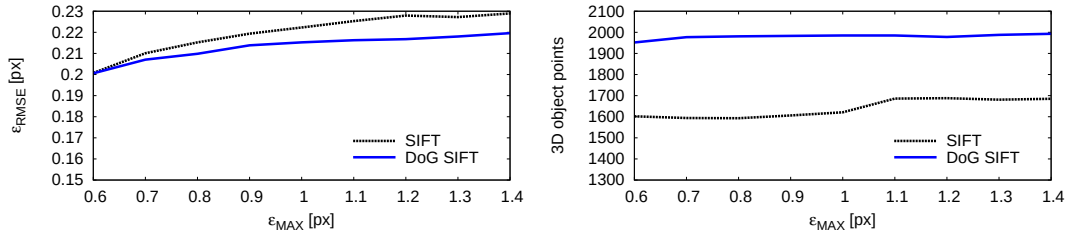


(b) *Fountain* sequence, 3000 detected points per image

Figure 6.4: Accuracy evaluation of the *Fountain* sequence (cf. Figure 6.1(a)): for 2000 and 3000 points in each image, the proposed DoG SIFT is shown with the blue line; the SIFT reference localization method is shown in black.

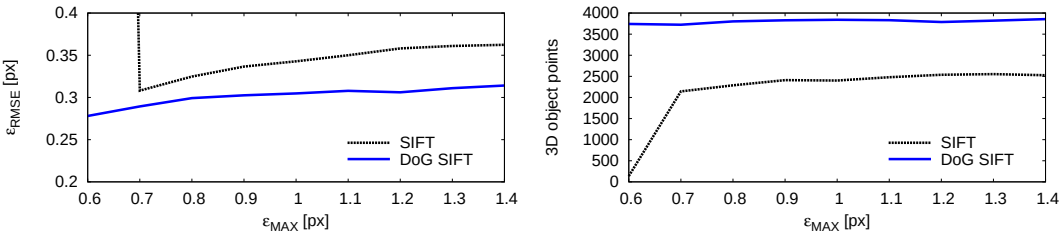


(a) *Herzjesu* sequence, 2000 detected points per image

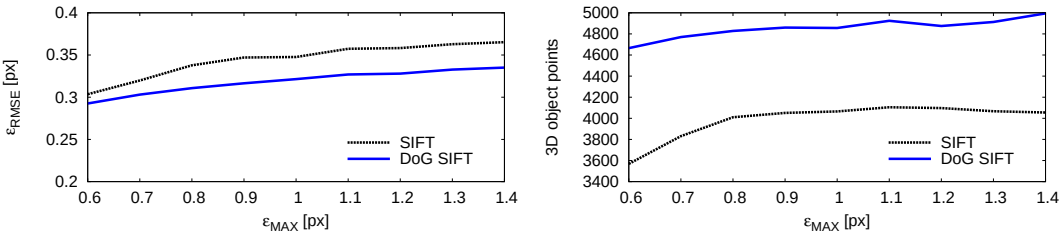


(b) *Herzjesu* sequence, 3000 detected points per image

Figure 6.5: Accuracy evaluation of the *Herzjesu* sequence (cf. Figure 6.1(b)): the DoG SIFT is shown with the blue line; the SIFT localization is shown in black.



(a) *Elefant* sequence, 3000 detected points per image



(b) *Elefant* sequence, 4000 detected points per image

Figure 6.6: Accuracy evaluation of the *Elefant* sequence (cf. Figure 6.1(c)): the DoG SIFT is shown in blue; the SIFT localization is shown in black.

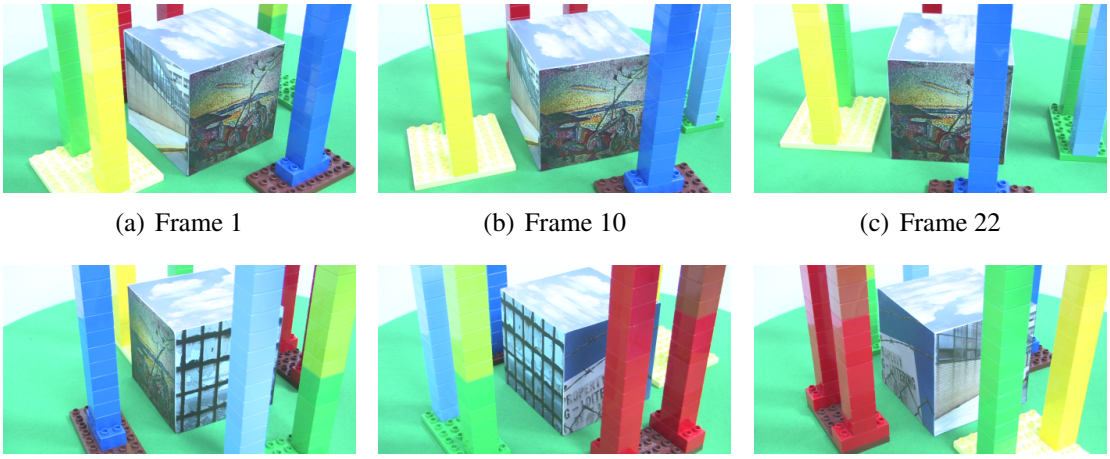
feature points per image) show no improvement of DoG SIFT compared to SIFT regarding the reprojection error. But, many more 3D object points are generated while reconstructing the scene. More object points usually lead to an increasing reprojection error. In this sequence, the reprojection errors of both methods are approximately the same, but DoG SIFT results in more object points.

The resulting diagram for the *Herzjesu* sequence (Figure 6.5, 2000 and 3000 feature points per image), and *Elefant* (Figure 6.6, 3000 and 4000 feature points per image) show an improvement of DoG SIFT compared to SIFT regarding the reprojection error and the number of reconstructed object points. The presented DoG SIFT feature localization results in a lower reprojection error and in more reconstructed 3D object points. Thus, the method clearly improves the reconstruction. In the *Elefant* sequence with 3000 detected points, the reference SIFT method fails for a maximal epipolar distance $\varepsilon_{\max}$ of 0.6 due to too few accurate corresponding features in the images.

To summarize, the reprojection error $\varepsilon_{\text{RMSE}}$ decreases for most of the sequences using DoG SIFT instead of the reference SIFT feature localization. It improves by 14.5% in maximum (*Elefant* sequence, 3000 detected points) while increasing the number of reconstructed object points significantly by 49.2%. The maximum improvement for the *Herzjesu* sequence is 11.9% while increasing the number of reconstructed object points by 43.8%. The maximal improvements are obtained for reasonable values for the outlier elimination threshold between $\varepsilon_{\max} = 0.8$ px and $\varepsilon_{\max} = 1.2$ px. For the *Fountain* sequence, the reprojection error does not improve, it slightly increases (cf. Figure 6.4).

But, many more 3D object points (8.7% in maximum) are reconstructed for *Fountain*, which leads to an increase of the reprojection error in general. Thus, we can suppose that the scene reconstruction improves by using DoG SIFT due to the high number of object points, but the reprojection error provides not necessarily any evidence.

By comparing Figure 6.5(a) with Figure 6.5(b) and Figure 6.6(a) with Figure 6.6(b), it turns out that the advantage of DoG SIFT decreases when selecting more features from the images. The reason is that there are many image features detected by the fullpixel localization procedure which can not be described by neither a 3D quadratic nor a Difference of Gaussians shape in the gradient pyramid. As expected, for the less accurate features a more sophisticated localization method is useless.



(a) Frame 1 (b) Frame 10 (c) Frame 22

(d) Frame 53 (e) Frame 96 (f) frame 145

Figure 6.7: Example images of the *Occluded Cube* turntable sequence with 191 frames showing a full 360° turn. The occlusion of background starts at frame 10. The first non-consecutive correspondences can be established in frame 22. From this image on, temporarily occluded scene content reappears.

6.2 Evaluation of Feature Trajectory Retrieval (FTR)

This experiment evaluates the combined feature tracking and matching approach as described in Section 5.1. The feature tracking connects image features in consecutive frames and builds trajectories. The feature matching, called *feature trajectory retrieval* (FTR) retrieves previously discontinued trajectories and builds connections between non-consecutive frames during sequential structure and motion recovery. In case of a successful feature match, the reappearing feature is assigned to the correct temporarily not visible 3D object point. This new constraint is incorporated in the bundle adjustment to improve the scene reconstruction. The remaining scene reconstruction pipeline is like in

Table 6.1: In the *Occluded Cube* sequence, several Lego™ towers in the foreground occlude the background. In the table, the textures of the visible cube planes are shown in the first column. The frames, in which the foreground starts to occlude the cube, are shown in the second column. The third column denotes the frame intervals in which non-consecutive correspondences are expected to be generated by the proposed FTR approach. Some images are shown in Figure 6.7. The top plane of the cube (*Clouds0000*) induces non-consecutive correspondences nearly throughout the sequence.

Texture	Occlusion starting frame	Frames with non-consecutive correspondences
Paintings1	10	22...49
PrisonWindow0010	45	53...94
FenceSign0001	88	96...143
Building0007	136	145...191
Clouds0000	10	24...185

the previous section. The sequences examined here are captured by video cameras and consist of many more frames than the sequences in Section 6.1. Thus, to guarantee maximal stability and comparability of the results, a bundle adjustment is performed after each frame.

To demonstrate the performance of the proposed FTR method and the new feature localization method DoG SIFT, three different sequences are presented in this section.

- *Occluded Cube*: captured by a professional camera in the studio using a tripod and a slowly turning turntable
- *Playground*: captured by a hand-held consumer camera on a playground
- *Sprayed Wall*: captured by a hand-held consumer camera on a street from a moving car

Two sequences (*Occluded Cube* and *Playground*) are used to compare the SIFT feature matching methods with the KLT feature tracking method. The feature matching methods were demonstrated in Section 6.1 for comparing the localization accuracy of feature detectors. The KLT feature tracking together with initially detected SIFT features is proposed in Section 5 for the retrieval of discontinued trajectories. KLT is expected to provide longer trajectories.

The scenes in *Occluded Cube* and *Sprayed Wall* mainly consist of known geometry, a cube (cf. Figure 6.7) and two planes (cf. Figure 6.8), respectively. This scene knowledge is used to evaluate the accuracy of the reconstructed point cloud in Section 6.2.3. The

comparison demonstrates the differences regarding the reprojection error and the point cloud accuracy by incorporating the proposed *feature trajectory retrieval* (FTR).

Figure 6.8: Example images 1, 16, 32, and 56 of the *Sprayed Wall* sequence with 56 frames taken from a slow driving car. The scene shows the walkway and a wall. Foreground occlusion is mainly induced by a tree.

6.2.1 Input Data and Experimental Setup

Example frames of the presented sequences are shown in Figure 6.7 (*Occluded Cube*), Figure 6.8 (*Sprayed Wall*), and Figure 5.1 (*Playground*). For each of the sequences, radial distortion is removed in a preprocessing step.

- The *Occluded Cube* turntable sequence is captured by a Cannon XL-H1 video camera on a tripod under controlled lighting conditions. The video resolution is 1920×1080 pixel in progressive recording mode. The sequence is captured with constant focal length. The intrinsic camera parameters are computed using a calibration pattern and the Tsai calibration method [97]. The image sequence contains 191 images. Example frames are shown in Figure 6.7. The scene consists of a textured cube which is temporarily occluded by towers of Lego™ bricks passing by in the foreground. The textures are taken from the VisTex [54] library and are listed in Table 6.1.

- The *Playground* sequence is captured outdoors on a playground. As shown in the example frames in Figure 5.1, the hand-held camera is moving from the right to the left. The background scene shows the static playground. The foreground consists of a swinging child and the swing rack. Both foreground elements temporarily occlude the background. Due to the varying motion while walking with the hand-held camera, many frames contain motion blur. The sequence is recorded with constant focal length. As the focal length was unknown for this sequence, it is estimated during sequential structure and motion recovery [90]. The scene is captured by a photo camera (Panasonic Lumix DMC TZ-81) using its video mode in a resolution of 1280×720 pixel.

- The *Sprayed Wall* sequence is again captured by the Panasonic Lumix DMC TZ-81 with the resolution of 1280×720 pixel and constant focal length. The scene is recorded from a slowly driving car. The camera is oriented to the left and observes

Table 6.2: Overview of evaluated methods. The methods SIFT and DoG SIFT use feature matching for frame to frame correspondences (like in the previous results Section 6.1). The others employ KLT tracking for frame to frame correspondences.

Method	Description
Harris KLT	Harris features with KLT tracking (cf. Section 3.1.1 and 3.2.1)
SIFT	SIFT features (cf. Section 3.1.2), frame to frame
DoG SIFT	DoG SIFT features (cf. Section 4.2.1), frame to frame
SIFT KLT FTR	SIFT with KLT tracking and FTR (cf. Section 5.1)
DoG SIFT KLT FTR	DoG SIFT with KLT tracking and FTR

the walkway together with a wall in the background. A tree and a bike in the foreground temporarily occlude the background scene. Example frames are shown in Figure 6.8.

The foreground objects are expected to cause broken trajectories leading to redundant and possibly erroneous 3D object points using the reference structure and motion recovery method. The FTR method presented in Section 5.1 should retrieve many of these discontinued trajectories.

This experiment is split into two parts which are presented in Section 6.2.2 and in Section 6.2.3. The first part focuses on the impact of the compared methods on the reprojection error $\varepsilon_{\text{RMSE}}$ and the mean trajectory length $|\bar{\mathbf{t}}|$ of the feature tracks. The mean trajectory length is determined by the ratio of the sum of trajectory lengths $|\mathbf{t}_j|$ and the number of reconstructed object points J:

$$|\bar{\mathbf{t}}| = \frac{1}{J} \sum_{j=1, \mathbf{P}_j \in \mathscr{J}}^{J} |\mathbf{t}_j| \quad . \tag{6.1}$$

The set $\mathscr{J}$ contains trajectories with a reconstructed object point $\mathbf{P}_j$. Additionally, the number of trajectories T with at least one established non-consecutive correspondence is counted. The results of the first part are presented in Section 6.2.2

The second part of the experiment in Section 6.2.3 examines the accuracy of the reconstructed point cloud. The objective is to demonstrate that the reprojection error $\varepsilon_{\text{RMSE}}$ not necessarily provides a suitable quality measure for comparing the presented approaches. This comes from the differently constrained bundle adjustment leading to (1) more object points if the features are more accurate on the one hand and (2) less object points if the FTR method performs well on the other hand. It is probable that the merging of feature tracks leads to an increase of the reprojection error.

Table 6.3: Results of the *Occluded Cube* sequence with 191 frames regarding trajectory lengths and reprojection errors $\varepsilon_{\text{RMSE}}$. The number of selected points per image is 1000; the maximal epipolar distance $\varepsilon_{\max}$ is set to 0.8 px. The table shows the number of reconstructed 3D objects points J, the number of trajectories T with at least one non-consecutive correspondence, and the resulting mean trajectory length $|\bar{\mathbf{t}}|$.

Feature localization	SIFT	DoG SIFT	Harris	SIFT	DoG SIFT
Correspondences	Feature matching		Feature tracking		
	SIFT	SIFT	KLT	KLT FTR	KLT FTR
T	0	0	0	289	312
J	14484	15354	16879	13748	13587
$\lvert\bar{\mathbf{t}}\rvert$	6.63	6.29	8.08	11.44	11.58
$\varepsilon_{\text{RMSE}}$	0.316	0.326	0.295	0.319	0.317

The maximal epipolar distance $\varepsilon_{\max}$ is set to the reasonable value of 0.8 px. For the methods SIFT and DoG SIFT, B_{SIFT} features are detected in each image. For the methods including tracking Harris KLT, SIFT KLT FTR, and DoG SIFT KLT FTR, $B_{\text{KLT}} = B_{\text{SIFT}}$ features are detected in the first image. If feature tracks get lost in the following frames, the amount of features is filled up again to $B_{\text{KLT}} = B_{\text{SIFT}}$ with newly detected features. For the methods SIFT KLT FTR, and DoG SIFT KLT FTR, only newly appearing and reappearing feature points are localized using SIFT and DoG SIFT, respectively. Most of the features are localized with the KLT tracker. Thus, the impact of the differing localization methods on the reconstruction accuracy is expected to be small.

The number of selected points per image is 1000 for *Occluded Cube* and 6000 for *Playground*. For the *Sprayed Wall* sequence, a varying number of points between 1000 and 7000 per image is chosen. This enables a more detailed evaluation depending on the number of tracked features. We select more features for *Playground* to obtain many discontinued trajectories which provide occlusion information as required for the application of automatic video segmentation which is demonstrated in Section 6.4.

6.2.2 Results: Trajectory Length and Reprojection Error

The evaluation results regarding the mean trajectory length is shown in Table 6.3 for *Occluded Cube* and in Table 6.4 for the *Playground* sequence. The presented methods for tracking and matching and their abbreviations are overviewed in Table 6.2. In this experiment, feature matching as used in Section 6.1 is compared to feature tracking using the structure and motion recovery reference (Harris and KLT, cf. Section 3) and the presented FTR method (cf. Section 5). The number of non-consecutive correspondences are

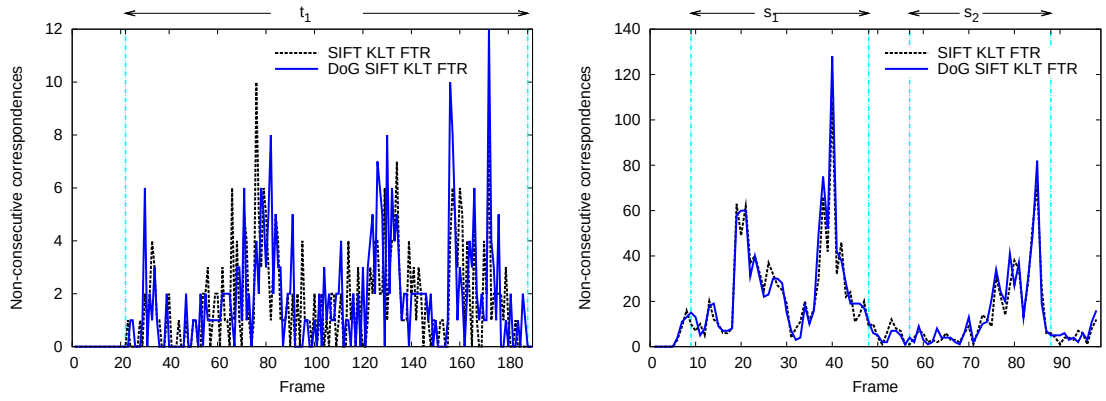
Table 6.4: Results of the *Playground* sequence with 99 frames regarding the mean trajectory length $|\bar{\mathbf{t}}|$. The number of selected points per image is 6000; the maximal epipolar distance $\varepsilon_{\max}$ is set to 0.8 px. The table shows the number of objects points J, the number of trajectories T with at least one non-consecutive correspondence.

Feature localization	SIFT	DoG SIFT	Harris	SIFT	DoG SIFT
Correspondences	Feature matching		Feature tracking		
	SIFT	SIFT	KLT	KLT FTR	KLT FTR
T	0	0	0	1635	1718
J	36638	37616	30600	25338	24165
$\|\bar{\mathbf{t}}\|$	9.58	8.73	15.47	19.39	20.35
$\varepsilon_{\text{RMSE}}$	0.222	0.220	0.250	0.226	0.215

shown in detail in Figure 6.9. The corresponding reprojection error $\varepsilon_{\text{RMSE}}$ after the bundle adjustment in each sequentially processed frame is shown in Figure 6.10.

- *Occluded Cube* (Table 6.3): compared to the methods using consecutive correspondences only, the number of reconstructed object points is reduced by the FTR methods. Consequently, the mean trajectory length $|\bar{\mathbf{t}}|$ increases. This is for two reasons: (1) the KLT tracking in consecutive frames provides longer trajectories as verified with the Harris KLT method; (2) the FTR method leads to non-consecutive correspondences merging about 300 3D object points for SIFT and DoG SIFT, respectively. The reprojection error $\varepsilon_{\text{RMSE}}$ is nearly the same for all methods except for Harris KLT which has lower $\varepsilon_{\text{RMSE}}$.

  A comparison regarding the number of non-consecutive correspondences is presented in Figure 6.9(a). As listed in Table 6.1, non-consecutive correspondences caused by foreground occlusion are expected from frame 22 until the end of the sequence. This is denoted with the interval t_1 and verified by the data shown in the diagram. The non-consecutive correspondences result in reasonable connections between trajectories before and after the occlusion of the cube. The varying amount of correspondences in the sequence is caused by the differently textured cube planes. Although the interpretability of the resulting reprojection error is limited for the methods presented here, it is shown in detail in Figure 6.10(a). While DoG SIFT improves slightly compared to SIFT, the difference diminishes when using the combination with KLT tracking. This is because too few features are localized by the detector. Most of the point positions result from tracking with KLT. Due to the many more 3D object points generated by the Harris KLT method, it results in a lower reprojection error. In this case, the bundle adjustment is less constrained because of redundant object points. Nevertheless, the trajectory length is



(a) *Occluded Cube* sequence

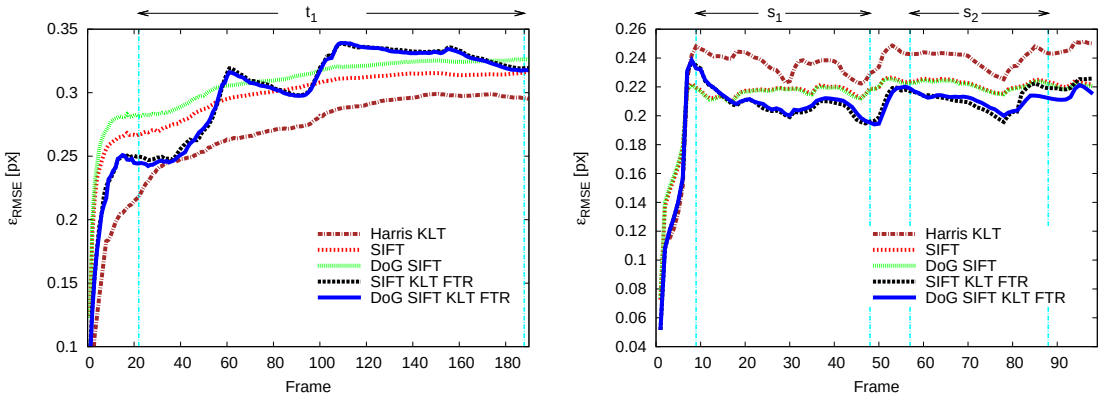
(b) *Playground* sequence

Figure 6.9: Validated non-consecutive correspondences for *Occluded Cube* (191 frames, 1000 features per image) and *Playground* (99 frames, 6000 features per image). The presented methods are subsumed in Table 6.2. The intervals t_1, s_1, and s_2 denote the frames where background reappears after being temporarily occluded.

higher than for feature matching using SIFT and DoG SIFT. Thus, Harris KLT should lead to a better reconstruction. An accuracy comparison between Harris KLT and DoG SIFT KLT FTR can not be derived from reprojection error and trajectory length. Harris KLT has lower reprojection error but the combined tracking and matching DoG SIFT KLT FTR results in longer trajectories.

- *Playground* (Table 6.4): in the *Playground* sequence, more feature points are selected in the images. Thus, the results in Table 6.4 show more object points J and more merged trajectories T. The trajectory lengths are larger in general because of the more translational camera movement compared to the turntable sequence *Occluded Cube*. As before, a large gain in trajectory length $|\bar{\mathbf{t}}|$ is achieved for the combined KLT tracking and FTR matching methods. Like in *Occluded Cube*, the resulting reprojection error $\varepsilon_{\text{RMSE}}$ is similar for all presented methods. Although slightly higher, Harris KLT provides a comparable reprojection error. The best results regarding the trajectory length are again achieved for the combined tracking and matching method DoG SIFT KLT FTR.

  The Figure 6.9(b) shows the number of established non-consecutive correspondences per frame. Like in the *Occluded Cube* sequence, most of the non-consecutive correspondences are established in the frame intervals s_1 and s_2, where they are expected. Inside these intervals the swinging child and the swing rack temporarily occlude the background. Only some non-consecutive correspondences are established outside s_1 and s_2, which is mainly due to discontinued KLT tracks be-



(a) *Occluded Cube* sequence (b) *Playground* sequence

Figure 6.10: Comparison of the reprojection errors for each frame in sequential scene reconstruction. The presented methods are subsumed in Table 6.2.

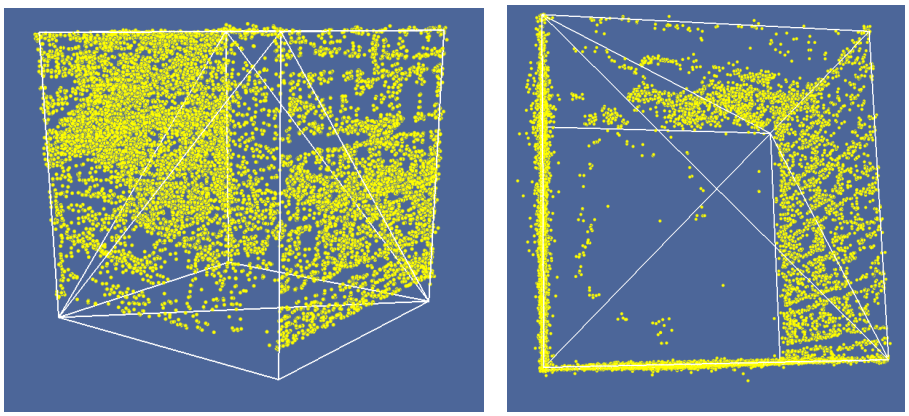
cause of motion blur (cf. Section 6.3). The diagram in Figure 6.10(b) reveals only slight differences between SIFT KLT FTR and DoG SIFT KLT FTR.

Overall, the trajectory length is significantly increased by the combined feature tracking and matching method using KLT and FTR while preserving a small reprojection error. The reprojection error of Harris KLT is smaller for *Occluded Cube*, but larger for *Playground*. Due to the smaller number of feature positions computed by the feature localization methods DoG SIFT and SIFT, the difference of exchanging these methods in the combined tracking and matching approach is relatively small. It cannot be observed that the feature trajectory length increases significantly with using the proposed DoG SIFT approach instead to SIFT feature localization. The best results regarding the trajectory length are achieved for the combined tracking and matching method DoG SIFT KLT FTR.

6.2.3 Results: Accuracy of the Reconstructed Point Cloud

The sequences *Occluded Cube* and *Sprayed Wall* are evaluated using the reconstructed point cloud resulting from the structure and motion estimation. The localization accuracy of the point cloud is measured using a 3D CAD object which represents the observed scene. For the *Occluded Cube* sequence the CAD object is a cube. For *Sprayed Wall*, the scene consists of two planes assembled to an edge. Both models and the corresponding correctly aligned reconstructed 3D point cloud are shown in Figure 6.11 and in Figure 6.12. The point cloud is aligned to the CAD model using an ICP (*iterative closest point*) based algorithm which is robust to outliers [17, 99]. Outliers are 3D object points which have a large distance to the CAD model, e.g. 3D object points which belong to the foreground. These points are removed from the point set. Thus, for the evaluation only

object points which are located nearby the surface of the CAD model are considered. The localization accuracy of the point cloud is determined by the distances between the remaining 3D object points and the CAD model. The workflow in this experiment is as follows:



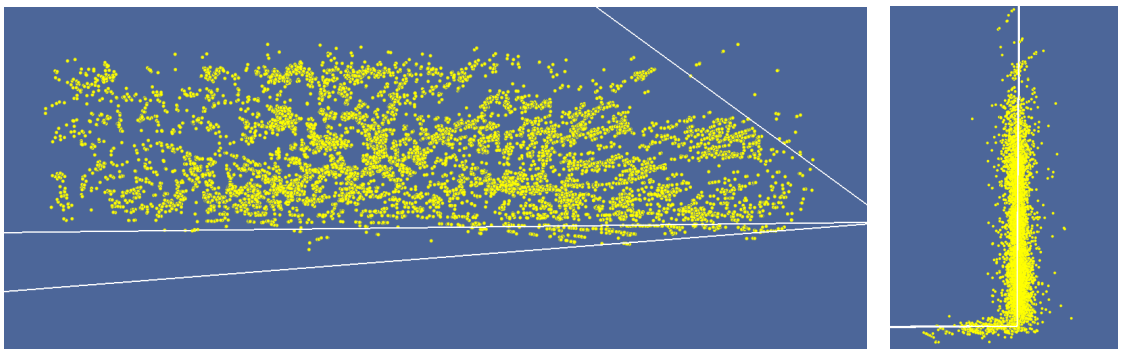
(a) Front view (b) Top view

Figure 6.11: The CAD model used for the point cloud evaluation of the *Occluded Cube* sequence. The yellow points show the point cloud alligned by an ICP- based algorithm.

The point cloud resulting from the structure and motion estimation is aligned to the 3D CAD model using an algorithm which is based on the ICP approach [17, 99]. It determines the relative position and orientation of the CAD model to the point cloud by minimizing a robust and scale invariant cost function $D_r(\mathbf{p}^\top, s)$ (similar to equation (8) of [99]),

$$D_r(\mathbf{p}^\top, s) = \sum_{j=1}^{J} (1 - exp(-\frac{d_j(\mathbf{p}^\top, s)^2}{\kappa s^3})) \quad , \tag{6.2}$$

with seven parameters $(\mathbf{p}^\top, s)$, $\mathbf{p} \in \mathbb{R}^6$, $s \in \mathbb{R}$. The vector $\mathbf{p}$ contains the 6 parameters for the translational and rotational mapping between the initial camera coordinate system and the object coordinate system of the CAD model (cf. equation (2.7)). The value s determines the scale factor between the reconstructed point cloud and the CAD model while κ determines the weighting of the costs of the change in scale. The values for $d_j(\mathbf{p}^\top, s)$ are Euclidean distances between each point $\mathbf{P}_j$ of the point cloud and the CAD model using the current parameter setting for $(\mathbf{p}^\top, s)$. The distances are calculated according to the presented method in [99]. The global optimization of the cost function is initialized with reasonable values for the search space boundaries and κ to achieve optimal convergence. After the minimization, the resulting parameters $(\mathbf{p}^\top, s)$ determine the alignment between the camera coordinate system and the object coordinate system. Then, points are removed from the point cloud by thresholding the Euclidean distance to the object.



(a) Camera view (b) Side view

Figure 6.12: The CAD model used for the point cloud evaluation of the sequence *Sprayed Wall* consists of two planes. After the alignment, the point cloud error ε_{OBJ} is measured by the sum of squared distances between the points and the CAD model.

The remaining number of object points is J_{OBJ}. The threshold is chosen to a value which removes the object points located on the foreground objects. In a second optimization step, the ICP algorithm recomputes the exact alignment of the coordinate systems using the squared Euclidean distance $d^2(\text{OBJ}, \mathbf{P}_j)$ between the CAD Object and the point $\mathbf{P}_j$ as error measure instead of the robust distance in the cost function (6.2). For the second optimization, the scale factor is set to 1. Examples for the point cloud alignment are shown in Figure 6.11 and in Figure 6.12. The resulting sum of squared distances is then used for the accuracy evaluation of the point cloud. The normalization with the number of object points J_{OBJ} gives the root mean squared error for each object point:

$$\varepsilon_{\text{OBJ}} = \sqrt{\frac{\sum_{j=1}^{J_{\text{OBJ}}} d^2(\text{OBJ}, \mathbf{P}_j)}{J_{\text{OBJ}}}} \quad . \tag{6.3}$$

For the cube model, the scale factor is determined by the ICP algorithm in the first robust optimization step. For the edge model, a large interval of scale factors are appropriate (cf. Figure 6.12). Hence, it is set manually to a reasonable value.

As the absolute sizes in the scene of the observed models are known, the error can be measured with absolute distances in mm. The ratio of a known length s_{SCENE} in the scene and the corresponding length in the object s_{OBJ} determines the scale factor $\frac{s_{\text{SCENE}}}{s_{\text{OBJ}}}$ for the computation of the resulting absolute error. The resulting distance $\varepsilon_{\text{OBJ}}[mm]$,

$$\varepsilon_{\text{OBJ}}[mm] = \frac{s_{\text{SCENE}}}{s_{\text{OBJ}}} \cdot \varepsilon_{\text{OBJ}} \quad , \tag{6.4}$$

shows the mean distance of each of the J_{OBJ} 3D object points to the surface of the observed scene and is used for the evaluation. The length of an edge of the real cube in the *Occluded*

Table 6.5: Results of the point cloud evaluation for *Occluded Cube* with 1000 tracked feature points and 191 frames for the full turn and 95 frames for the half turn.

Full turn	J_{OBJ}	T	ε_{OBJ}	$\varepsilon_{\text{OBJ}}[mm]$
SIFT KLT	9857	0	1.04	1.57
SIFT KLT FTR	9689	289	1.00	1.50
DoG SIFT KLT	10587	0	0.96	1.44
DoG SIFT KLT FTR	10477	312	0.95	1.43

Half turn	J_{OBJ}	T	ε_{OBJ}	$\varepsilon_{\text{OBJ}}[mm]$
SIFT KLT	5179	0	0.87	1.31
SIFT KLT FTR	5198	142	0.79	1.19
DoG SIFT KLT	5275	0	0.75	1.13
DoG SIFT KLT FTR	5247	161	0.73	1.10

Cube sequence is 150 mm. The height of the wall in the *Sprayed Wall* sequence is 2740 mm.

- *Occluded Cube* (Table 6.5): the results for *Occluded Cube* are shown for two cases: (1) for the complete 360° turn and (2) for a 180° turn. The distance between camera and cube is about 400 mm. All object points on the foreground objects are discarded using the threshold of 15 mm for the Euclidean distance between point and cube. The corresponding distance threshold in the CAD object coordinate system is 10.

  By comparing SIFT KLT and the developed DoG SIFT KLT FTR, the localization of the point cloud improved by 9.9 % for the full 360° turn and by 16.1 % for the 180° turn.

  The localization of the 10000 points in the full turn is improved by 0.14 mm per point, the localization of the 5000 points in the half turn is improved by 0.21 mm per point.

- *Sprayed Wall* (Table 6.6 and Figure 6.13): for the *Sprayed Wall* sequence, the distance between camera and the wall is about 6200 mm. Object points on the foreground objects are removed using the threshold of 700 mm for the Euclidean distance between each point and the wall. The corresponding distance threshold in the CAD object coordinate system is about 80. As for this model many scale factors are appropriate for the point cloud alignment, the point cloud scaling value s_{OBJ} is slightly different for each reconstruction. Hence, the value is adapted manually for each reconstruction using known scene lengths.

  In this evaluation, different numbers of tracked features are considered. In Table 6.6, the resulting trajectory mean lengths $|\bar{\mathbf{t}}|$ and the distance measure $\varepsilon_{\text{OBJ}}[mm]$

Table 6.6: Results of the point cloud evaluation for *Sprayed Wall* with different numbers of tracked feature points. Due to the simple CAD edge model, the scale factor between ε_{OBJ} and $\varepsilon_{\text{OBJ}}[mm]$ is slightly different for each evaluation.

1000 tracked features	J_{OBJ}	T	$\lvert\bar{\mathbf{t}}\rvert$	ε_{OBJ}	ε_{OBJ} [mm]	$\varepsilon_{\text{RMSE}}$ [px]
SIFT KLT	1765	0	7.03	5.59	66.37	0.2012
SIFT KLT FTR	985	38	8.08	6.19	68.40	0.1999
DoG SIFT KLT	2375	0	6.93	4.04	47.77	0.2093
DoG SIFT KLT FTR	2239	119	8.30	4.12	48.84	0.2137

3000 tracked features	J_{OBJ}	T	$\lvert\bar{\mathbf{t}}\rvert$	ε_{OBJ}	ε_{OBJ} [mm]	$\varepsilon_{\text{RMSE}}$ [px]
SIFT KLT	6096	0	7.10	4.76	58.68	0.2118
SIFT KLT FTR	5770	243	8.43	4.74	58.42	0.2160
DoG SIFT KLT	6916	0	6.92	4.42	54.42	0.2125
DoG SIFT KLT FTR	6587	343	8.39	4.36	53.50	0.2169

5000 tracked features	J_{OBJ}	T	$\lvert\bar{\mathbf{t}}\rvert$	ε_{OBJ}	ε_{OBJ} [mm]	$\varepsilon_{\text{RMSE}}$ [px]
SIFT KLT	10693	0	7.02	4.60	55.95	0.2128
SIFT KLT FTR	10386	467	8.27	4.59	55.82	0.2182
DoG SIFT KLT	11506	0	6.91	4.54	55.22	0.2127
DoG SIFT KLT FTR	11065	569	8.11	4.55	55.34	0.2182

7000 tracked features	J_{OBJ}	T	$\lvert\bar{\mathbf{t}}\rvert$	ε_{OBJ}	ε_{OBJ} [mm]	$\varepsilon_{\text{RMSE}}$ [px]
SIFT KLT	15163	0	6.93	4.75	53.95	0.2138
SIFT KLT FTR	14690	626	8.06	4.83	54.74	0.2182
DoG SIFT KLT	15848	0	6.85	4.77	54.93	0.2134
DoG SIFT KLT FTR	15170	742	8.04	4.77	54.83	0.2188

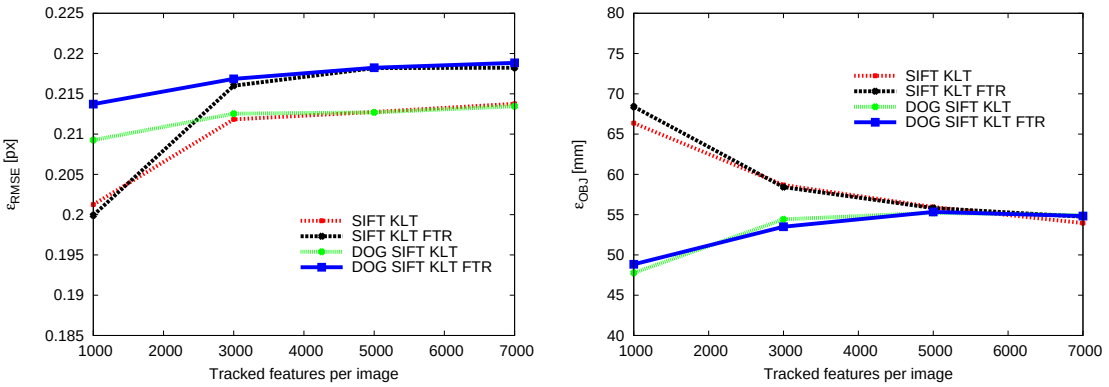
are demonstrated. Additionally, the resulting reprojection error $\varepsilon_{\text{RMSE}}$ is shown. As expected, the mean trajectory length for the FTR method is generally larger. The number of reconstructed object points is smaller for FTR. If FTR is used, the feature localization method DoG SIFT leads to more trajectories with non-consecutive correspondences compared to the SIFT localization method.

The accuracy of the reconstructed point cloud for the proposed localization method DoG SIFT performs superior to the other methods regarding 1000 and 3000 tracked features. For a higher amount of tracked features the benefit decreases. For 5000 and 7000 tracked features the results for $\varepsilon_{\text{OBJ}}[mm]$ are approximately the same. Note, that the proposed method DoG SIFT generally leads to a higher amount of object points.

In Figure 6.13, the reprojection errors and the distances to the CAD model are visualized. Although the same feature detectors are used, the reprojection error is larger for the FTR methods than for the methods without using FTR (cf. Figure 6.13(a)). The reprojection errors of the proposed combination of DoG SIFT and FTR are even the largest. On the contrary, the distance $\varepsilon_{\text{OBJ}}[mm]$ to the point cloud is small. This shows, that the reprojection error does not provide a fair distance measure for the comparison of differently constrained bundle adjustments. Like in Section 6.1, the optimal number of tracked features is 3000, which is a reasonable value for the approach of structure and motion recovery.

The improvement in $\varepsilon_{\text{OBJ}}[mm]$ is about 28 % for 1000 tracked feature points, which corresponds to the mean error of 48 mm instead of 66 mm for each of the reconstructed object points. In this case, the reference SIFT procedure for the feature localization provides a small amount of object points. These points appear to be insufficient for a suitable representation of the scene which leads to an unstable reconstruction result. Although the reprojection error $\varepsilon_{\text{RMSE}}$ is low for 1000 tracked features, the localization error of the point cloud $\varepsilon_{\text{OBJ}}[mm]$ is high.

The improvement of DoG SIFT KLT FTR for 3000 tracked feature points is 9 %, which corresponds to 5 mm improvement per point. For a high amount of features, the impact of DoG SIFT KLT FTR on the reconstruction accuracy vanishes.



(a) Reprojection error

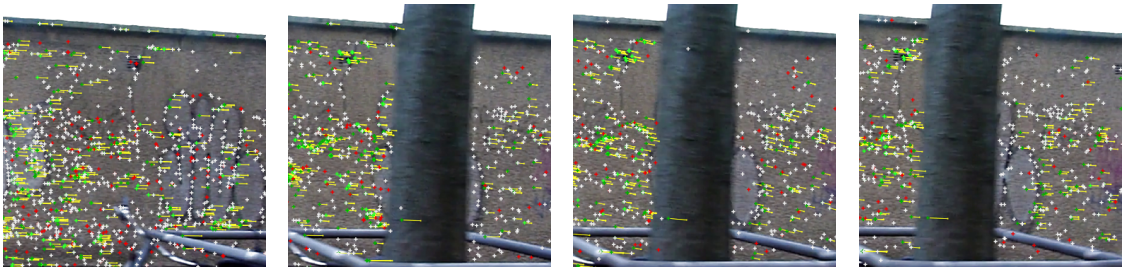
(b) Point cloud error

Figure 6.13: The resulting reprojection error $\varepsilon_{\text{RMSE}}$ and the localization error of the reconstructed point cloud $\varepsilon_{\text{OBJ}}[mm]$ for the *Sprayed Wall* sequence.

The Figure 6.14 shows image examples for background texture before the occlusion with the tree (Figure 6.14(a)) and after its reappearance (Figures 6.14(b) - Figures 6.14(d)). In this example, the number of features per frame is 5000. By manually counting the feature points detected on the reappearing texture, the maximally possible amount of retrieved trajectories can be estimated. For frame 15, 35 features reappear after being occluded by the tree. For frame 16, 43 features

reappear. The number of image frames, in which background reappears after being occluded by the tree is 23. The number of features which reappear is approximately the same for each of the 23 frames. It follows, that in total about 900 discontinued trajectories are induced by the tree. As the number of extracted trajectories with a non-consecutive correspondence is $T = 569$ (cf. Table 6.6, 5000 tracked features), the success rate of the retrieval is about 60 %.

It should be noted, that new features are detected in any region of the image. Only features nearby the tree can leads to a successful non-consecutive correspondence. Additionally, the frames where background textures reappear are limited (23 frames) due to the camera motion. The 3D object points are generated in the entire sequence (56 frames). Thus, the number of retrieved correspondences is relatively small compared to the number of reconstructed object points.



(a) Frame 5 (b) Frame 14 (c) Frame 15 (d) Frame 16

Figure 6.14: Part of the *Sprayed Wall* sequence with feature points and correspondences to the previous frame. Frame 5 shows the background texture while the frames 14, 15, and 16 show its reappearance.

For all the sequences included in this chapter, the proposed FTR significantly increases the trajectory length. In combination with the proposed feature localization method DoG SIFT, the resulting point cloud has the smallest errors. It is demonstrated that the reprojection error is not suited for a fair comparison between reconstructions resulting from differently constrained bundle adjustments.

6.3 Evaluation of Occlusion Information

In this section, the effectiveness of the presented method for the generation of occlusion information as shown in Chapter 5 is demonstrated using natural image sequences. The results of this experiment are image regions derived from trajectories with non-consecutive correspondences for each image of the input sequence. A visualization of these regions is shown in Figure 5.6. The regions are classified as foreground or background using the validation with a neighborhood histogram comparison as explained in

Section 5.3.1. The data extracted from these trajectories is called *occlusion information* (cf. Section 5.3.1).

In this experiment, the number of connected trajectories is compared with the number of extracted foreground regions.

6.3.1 Input Data and Experimental Setup

During sequential structure and motion recovery, newly appearing features are reconnected to previously discontinued feature trajectories as explained in Section 5.1. The new feature in the current image is assigned to the corresponding feature trajectory and its 3D object point. The new constraint is used in the bundle adjustment. The reprojections of each 3D object point into previous camera images provide image positions with possibly occluded scene content. After the validation procedure as explained in Section 5.3.1 the occluded image positions are extracted. These regions are deemed to belong to the foreground.

In this experiment, the number of connected trajectories and the number of occluded image positions are compared for four natural image sequences. The examples presented here are the *Playground* sequence, the *Column* sequence, the *Person* sequence, and the *Hand* sequence. The *Playground* sequence and the *Column* sequence are captured with a consumer photo camera using its video mode of 1280×720 pixels. The *Person* sequence and the *Hand* sequence are taken from a test sequence data base for video segmentation [79]. Their resolution is 720×480 pixel. Each of the sequences contains a static background scene and a moving foreground object which temporarily occludes the background. The *Playground* sequence additionally shows some static untextured foreground scene parts which temporarily occlude the background scene and the moving object. As the intrinsic camera parameters were unknown for these sequences, they are estimated during sequential structure and motion recovery [90].

- The *Playground* sequence (Figure 5.1) is already known from Chapters 5 and Section 6.2. The camera moves from the right to the left. The background is temporarily occluded by the swinging child and the swing rack.

- The *Column* sequence (Figure 6.19, top row) shows one large foreground object, a column, which passes the field of view twice. The camera moves from the right to the left and then back to the right.

- The *Person* sequence (Figure 6.20), top row) shows a person walking from right to left while the camera is slowly panning to the left.

- In the *Hand* sequence (Figure 6.21, top row), the camera movement is low and pans back and forth. In this sequence, the foreground object, a hand, is located in the center for all images, in particular from the first frame on.

6.3.2 Results

In the evaluation, the number of extracted foreground positions per frame is compared with the number of connected trajectories per frame. The resulting diagrams for *Playground* and *Column* are shown in Figure 6.15. The results for *Person* and *Hand* are shown in Figure 6.18.

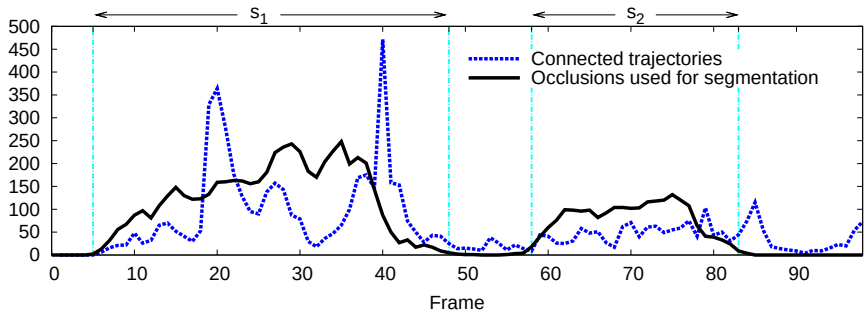
- For *Playground*, the number of extracted foreground regions and the number of connected trajectories are shown for each frame in Figure 6.15(a). The frame interval in which the child and the swing rack occlude the scene for the first time is denoted with s_1. The second occlusion interval is denoted with s_2. Within these intervals, many trajectories are connected (blue line). The numbers of extracted occluded image positions are plotted with the black line. Note, that one connected trajectory may provide numerous useful occluded image positions in the previous frames. On the other hand, no foreground region is induced if the trajectories discontinue without occluding scene content, e.g. for the frames 48-57 and 83-98. Most of the foreground regions are extracted inside the intervals s_1 and s_2.

  The peaks for connected trajectories for frame 20 and frame 40 are due to strong motion blur in the previous frames. This is illustrated for the example of frame 40 in Figure 6.16. Many trajectories discontinue after frame 38 due to motion blur in frame 39. This leads to many continuations of trajectories in the frame 40 and possibly in the following frames. As shown in Figure 6.15(a), the number of extracted occlusions in the frames before these peaks show no significant change, because many of the trajectories do not contain non-consecutive correspondences induced by occlusion.

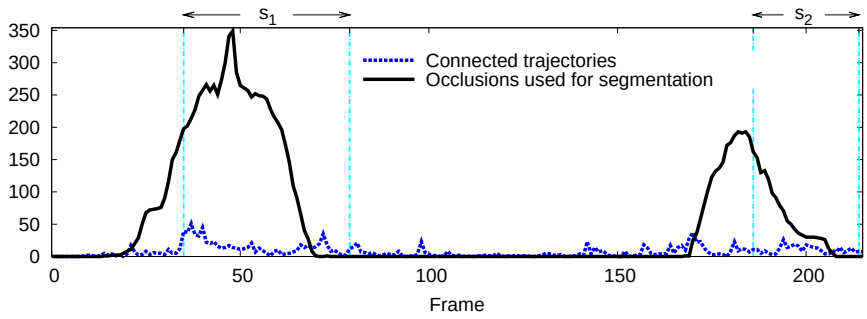
- In the *Column* sequence (Figure 6.15(b)), trajectories with non-consecutive correspondences due to occlusion are expected in the interval s_1 (between the frames 35 and 79) and in interval s_2 (between the frames 168 and 215). Because of the large foreground object, connected trajectories induced by occlusion lead to many foreground regions in the frames before. The column is visible for the frames 15 - 78 and after frame 167 until the end of the sequence. As shown in Figure 6.15(b), foreground regions are extracted in these frames only. To the end of the sequence the number of foreground regions decrease because the background features do not reappear.

  Non-consecutive correspondences are extracted in the other frames as well. This is due to motion blur as illustrated for an example in Figure 6.17. The frames 168 and 169 show significant motion blur, which leads to a peak in the diagram of connected trajectories after these frames.

- The results of the *Person* sequence (Figure 6.18(a)) show a monotonically increasing number of extracted foreground regions until frame 45 while the person comes



(a) *Playground* sequence [26], image examples in Figure 5.1



(b) *Column* sequence, image examples in Figure 6.19

Figure 6.15: Results for the sequences taken with a hand-held camera: the number of connected trajectories in each frame (dotted blue line) and the number of occlusions used for the segmentation for each frame (black line). The intervals s_1, s_2 depict the parts where background reappears after being occluded. If a connected trajectory results from occlusion, numerous reprojections of the corresponding 3D object point are usable for the segmentation.

nearer to the camera. After the person has left the field of view in frame 60, no more foreground regions are extracted.

In this sequence, motion blur is avoided by using a slowly moving camera with small translational movement. Not regarding the connected trajectories induced by motion blur, the results of *Playground* and *Column* are similar to the results of this sequence.

- In the *Hand* sequence (Figure 6.18(b)), the translational camera movement is small and pans back and forth. In contrast to the examples presented before, a large proportion of the connected trajectories is induced by object points which leave and re-enter the field of view several times. Additionally, the translational movement of the camera is small and the foreground object is already present in the first frame of the sequence. Thus, the number of extracted foreground regions is small.

(a) Frame 38 (b) Frame 39 (c) Frame 40 (d) Frame 41

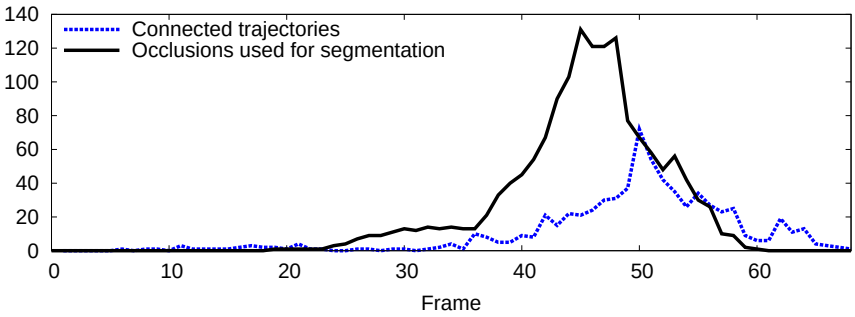
Figure 6.16: Background part of the *Playground* sequence. The frame 39 shows significant motion blur due to the shaky hand-held camera. The blur leads to an increase in the number of discontinued feature trajectories after frame 39 (cf. Figure 6.15(a)). The figure shows parts of the images $38, 39, 40, 41$.

(a) Frame 167 (b) Frame 168 (c) Frame 169 (d) Frame 170

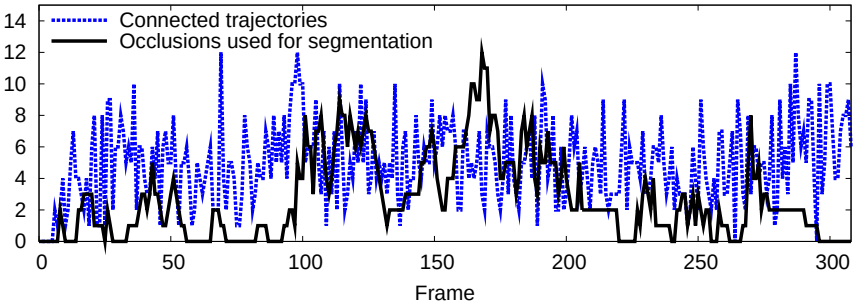
Figure 6.17: Part of the *Column* sequence. The frames 168 and 169 shows significant motion blur due to the shaky hand-held camera. The blur leads to an increase in the number of discontinued feature trajectories after these frames as shown in Figure 6.15(b)). The figure shows parts of the images $167, 168, 169, 170$.

The number of extracted foreground regions varies from sequence to sequence. It depends on the background texture, the camera movement, and the object motion. In the *Playground* sequence, up to 250 foreground positions are extracted in a frame. Due to the large foreground object, the *Column* sequence provides up to 350 extracted foreground positions. In these two sequences, many trajectories are connected due to motion blur, which supports the scene reconstruction. But, these trajectories do not result in foreground regions.

In the sequences from the data set [79], no motion blur is present. In the *Person* sequence, up to 137 foreground positions are extracted for a frame. In case of very limited camera and object movement throughout the sequence like in the *Hand* sequence the number of connected trajectories is smaller. Many of the trajectories are connected because their object points leave and re-enter the field of view. Thus, only a few foreground positions (between 0 and 12) are extracted. Although the scene reconstruction improves



(a) *Person* sequence [24], image examples in Figure 6.20



(b) *Hand* sequence [24], image examples in Figure 6.21

Figure 6.18: Selected sequences from the data set [79]: the number of connected trajectories in each frame (dotted blue line) and the number of extracted occlusions (black line). If a connected trajectory results from occlusion, numerous reprojections of the corresponding 3D object point are usable for the segmentation.

by connecting the discontinued trajectories, the *Hand* sequence does not provide optimal conditions for the automatic segmentation approach as presented in this work.

The automatic video segmentation and the following visual effect creation using these four sequences is presented in the next section.

6.4 Demonstration: Integration of Virtual Objects between Scene Elements

An important visual effect (VFX) used in movie production is the integration of virtual objects into natural image sequences. With this technique, it is possible to add synthetic objects to a previously captured scene by the movie editor. To guarantee the perspectively correct relative positions of the integrated objects in each camera view, accurate camera parameters for the input image sequence are required. The technique of structure and motion recovery as presented here provides an automatic solution for this. Another important

tool for VXF creation is video segmentation, often called *matte creation* [47], which separates the foreground from the background. The segmentation is used to apply different image processing tools to the layers of the image sequence. One example is the occlusion of the integrated objects with foreground scene content, such as an actor moving in front of a partial virtual scene. In movie production industry, the matte creation is still mainly done manually [47].

An automatic approach for matte creation using occlusion is presented in Chapter 5. The segmentation technique incorporates 3D information from the structure and motion estimation to distinguish between foreground and background. Image regions are deemed foreground if they occlude the background scene temporarily. The foreground regions following from the automatic video segmentation are used to occlude the integrated virtual objects in a compositing step as shown in Section 5.4.

The video segmentation can be used for various VFX. As a second example, the background blur effect is presented. This technique focuses the observers attention on the foreground by applying a strong Gaussian blur to the background.

In this Section, the applications are shown using natural image sequences and synthetic objects which are integrated into the sequence using the 3D modeling tool Blender[2] and the estimated camera path resulting from the presented approach.

6.4.1 Input Data and Experimental Setup

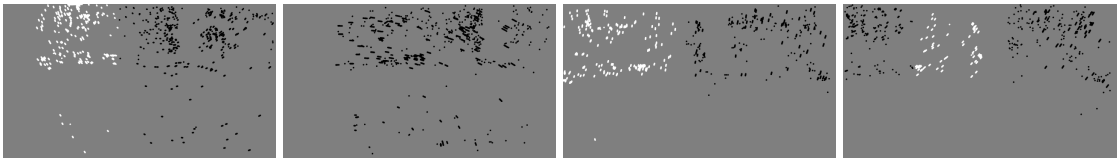
The input data for the application demonstration is the same as in the previous Section 6.3: image sequences with static scene background and a foreground object. The four sequences *Playground*, *Column*, *Person*, and *Hand* are introduced in Section 6.3. For all the sequences, the camera path is estimated using the presented structure and motion recovery workflow (cf. Figure 5.3). The scene reconstruction is used for the integration of the virtual objects into the scene. Here, the accurate estimation of the camera parameters is crucial to guarantee the perspectively correct relative positions of the objects in each view. Occlusion information is extracted from the established trajectories. It initializes the automatic video segmentation approach. The initialization using only the occlusion information is compared with the presented ALO (*appearance learning from occlusions*) method which collects foreground color values from the foreground regions of the sequence. The compared initialization methods are

- 2D grid segmentation without ALO
- 2D grid segmentation with ALO
- 3D grid segmentation with ALO

For the sequences *Playground*, *Column*, and *Person* virtual objects are integrated into the sequence. The video segmentation is used for the automatic occlusion of the integrated

[2] `http://www.blender.org`

(a) Input sequence



(b) Occlusion information

(c) 2D grid segmentation

(d) 2D grid segmentation with ALO

(e) 3D grid segmentation with ALO

(f) Integrated objects

Figure 6.19: For the *Column* sequence, the ALO scheme shows a significant improvement compared to the standard 2D grid segmentation, especially in the second image example. Together with using a 3D grid segmentation, the best results are achieved. The bottom row shows integrated objects which are occluded using the 3D grid segmentation with ALO [24]. The images show the frames 57, 72, 185, and 196.

objects in a compositing step as shown in Section 5.4. The *Hand* sequence is used to demonstrate the background blur effect.

6.4.2 Results

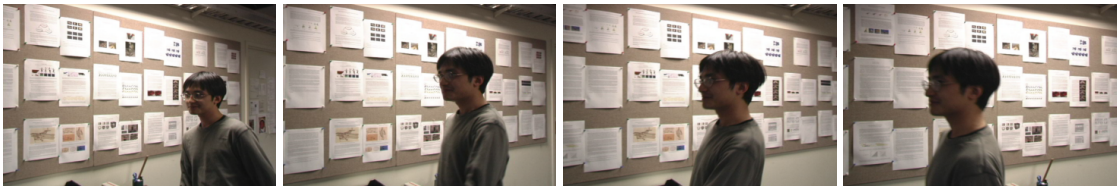
The sequences, the extracted occlusion information, the resulting segmentation, and the application are shown in Figure 6.19 (*Column*), in Figure 6.20 (*Person*), and in Figure 6.21 (*Hand*). The results of the *Playground* sequence have already been presented for demonstrating the algorithms in Chapter 5. The input sequence and the occlusion information are shown in Figure 5.6, the result is shown in Figure 5.7.

For each of the Figures 6.19, 6.20, and 6.21, the first row gives the input sequence, the second row shows the occlusion information with white regions illustrating the foreground and black regions illustrating the background. Then, the segmentation results are shown in the following rows. The last row demonstrates the application which is the integration of virtual objects between scene elements for *Column* and *Person*. The background blur effect is demonstrated for the *Hand* sequence in the last row of Figure 6.21. Here, a strong Gaussian blur is applied to the background to focus the observers attention on the foreground. The resulting videos are available for download at: `http://www.tnt.uni-hannover.de/staff/cordes/`

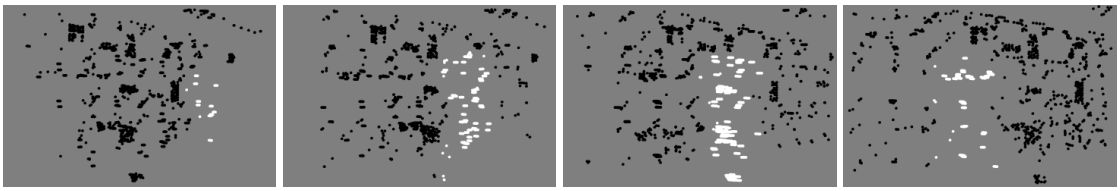
- In the *Column* sequence (Figure 6.19), a column is passing the field of view twice. Example frames of the input sequence are presented in Figure 6.19(a). Figure 6.19(b) shows the extracted occlusion information. Initialized with these images, the standard two-dimensional grid (2D grid) segmentation results in Figure 6.19(c). The usage of the presented approach for ALO improves the segmentation, visible in Figure 6.19(d) while the three-dimensional grid (3D grid) segmentation together with ALO results in Figure 6.19(e). This method provides the best results.

  Application: Two virtual objects are integrated into the sequence (cf. Figure 6.19(f)). They are placed on the ground plane *behind* the column. Due to an accurate estimation of the camera parameters, the objects are perspectively correct in each camera view. They show no drift. The occlusion of the objects using the segmentation results is convincing.

- In the *Person* sequence (Figure 6.20), a person is walking from right to left while the camera is panning slowly to the left (cf. Figure 6.20(a)). The extracted foreground and background regions are shown in Figure 6.20(b). These images and the extracted foreground and background models are used to initialize the 2D grid segmentation (cf. Figure 6.20(c)). The 2D grid segmentation including ALO is shown in Figure 6.20(d). The 3D grid segmentation including ALO (cf. Figure 6.20(e)) provides the best results for the video segmentation.



(a) Input sequence

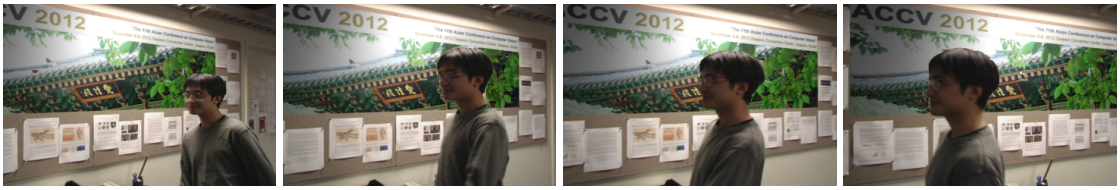


(b) Occlusion information

(c) 2D grid segmentation

(d) 2D grid segmentation with ALO

(e) 3D grid segmentation with ALO



(f) Integrated objects

Figure 6.20: For the *Person* sequence [79], the ALO approach adds too much foreground content which is improved by the combination of 3D grid segmentation and ALO. Again, this combination provides the best results. The bottom row shows the integration and occlusion of the *ACCV* logo on the board [24]. The images show the frames 35, 43, 47, and 53.

Application: This sequence was used for a publication in the *ACCV 2012* proceedings [24]. Thus, the integrated virtual object is the *ACCV* logo which is attached to the board in the background. The virtual object is occluded automatically using the video segmentation in a compositing step.

- In the *Hand* sequence (Figure 6.21), scene content leaves and re-enters the field of view. Due to the limited camera movement, only few occlusion information is available (cf. Figure 6.18(b)). The visualization of the occlusion information is given in Figure 6.21(b). Only very few foreground positions are extracted. Some of the frames contain no extracted foreground region. This initialization is not sufficient for a frame by frame segmentation without using ALO as shown in Figure 6.21(c). For the right image, no foreground regions is segmented. Nevertheless, a reliable segmentation is obtained using the presented ALO scheme (cf. Figure 6.21(d)). This result is achieved without using a 3D graph. The improvement results from ALO only. Although, no foreground information is available in the fourth image example, the resulting segmentation is correct. The 3D grid segmentation leads to nearly the same results as the 2D grid segmentation.

  Application: In Figure 6.21(e), the application of the *background blur* effect is demonstrated for this sequence.

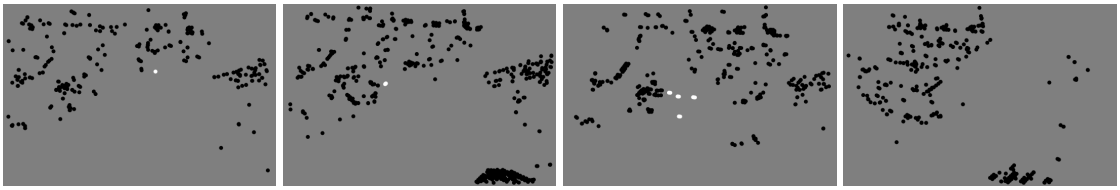
- The results of the *Playground* sequence have already been presented in Figure 5.6 and in Figure 5.7.

  Application: Like in the *Column* sequence, two objects are integrated. They are placed on the ground plane behind the swinging child. The automatic occlusion with the foreground objects is convincing.

The presented *appearance learning from occlusion* (ALO) provides useful information which leads to solutions even when no foreground region is available for several frames. Generally, ALO leads to more pixels assigned to the foreground because more color values are included for the representation as GMM models for the foreground. This enables the nearly perfect segmentation results for the *Hand* sequence. In the examples *Column* and *Person*, ALO leads to a slight over segmentation for some frames when the 2D grid segmentation is used. This problem is solved by using the 3D grid segmentation, which leads a temporarily consistent segmentation. This combination provides the best results. For the *Hand* sequence, the combination of 2D grid segmentation and ALO already provides optimal results. Using the 3D grid segmentation is not necessary.

Due to the accurate estimation of the camera parameters, the integrated objects are perspectively correct in each camera view. They show no drift. The video segmentation leads to a convincing occlusion of the objects with scene foreground.

(a) Input sequence



(b) Occlusion information

(c) 2D grid segmentation

(d) 2D grid segmentation with ALO

(e) Background blur effect

Figure 6.21: For the *Hand* sequence [79], the segmentations are obtained by using the 2D grid graph. The improvement in the fourth row is due to ALO. The 3D grid graph does not lead to further improvements. The bottom row demonstrates the blurred background effect [24]. The images show the frames 102, 229, 271, and 349.

(a) Color similarity of foreground and background (b) Smeared object boundaries due to motion blur

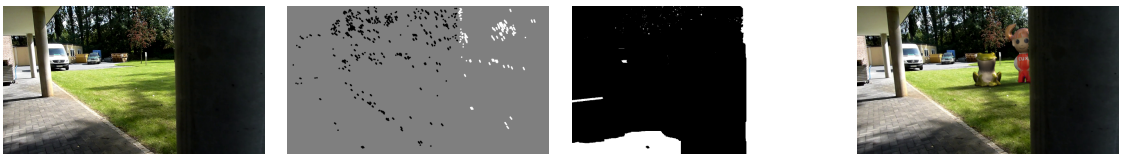
Figure 6.22: Errors resulting from a misleading segmentation [25]. (a): although there is no occlusion information in the fence, it is segmented as foreground because of its visual similarity to the swing rack; (b): although the point is correctly classified as foreground, it is isolated by the segmentation algorithm because of the strong motion blur of the foreground object.

6.4.3 Limitations

Although for most regions of the shown sequences the foreground is segmented reliably, it can happen that background regions are labeled as foreground because of visual similarities. The Figures 6.22 and 6.23 show examples in detail.

- In Figure 6.22(a), a small part of the fence which belongs to the background occlude the virtual objects because of a misleading segmentation. Here, the fence is visually very similar to the part of the swing rack which is a foreground region. Additionally, there is no boundary visible between fence and rack (cf. right image of Figure 6.22(a)).
- In Figure 6.22(b), the segmentation algorithm assigns a small part of the child to the background, although it has attached a correctly classified foreground region (small white region in the left image of Figure 6.22(b)). This is due to the strong motion blur which smears the boundary edge of the foreground object. Here, the energy minimization results in a segmentation boundary inside the foreground object (cf. right image of Figure 6.22(b)).
- In Figure 6.23, the region in the bottom left is labeled as foreground (cf. Figure 6.23(c)) although there is a correctly located background position (cf. Figure 6.23(b)). The region contains color values which are very similar to the foreground object, the column. Thus, the region is assigned to the foreground. Here, the application of the automatic occlusion of the virtual objects is not affected as shown in Figure 6.23(d).

In these examples, the segmentation algorithm leads to suboptimal solutions. In each of the cases, user interaction would solve the problem. User interaction is common in the field of image and video segmentation. The user marks some additional strokes which helps the algorithm to determine the Gaussian mixture models of foreground and background. This user interaction can easily be integrated in the presented framework [26].



(a) Input (b) Classified regions (c) Segmentation (d) Integrated objects

Figure 6.23: Background labeled as foreground: due to the visual similarity of the column to the shadowed background, the background region on the bottom left is labeled as foreground. The correctly extracted background point in this region is isolated. In this case, it does not affect the final result with the integrated objects shown on the right.

The visualization of the occlusion information additionally helps the user as orientation which eases the editing and reduces the required user strokes. The results presented in this chapter are obtained fully automatic.

7 Conclusions

The presented techniques extend a sequential structure and motion recovery technique with a feature tracking which handles foreground occlusions. In common approaches, foreground occlusions disturb the tracking and therefore decrease the reconstruction accuracy. In the presented application of integrating virtual objects into video, a highly accurate estimation of the camera path is required for perspectively correct views on the virtual objects in each camera. Furthermore, if foreground objects occlude the scene, the integrated objects should be occluded by the foreground objects, too.

In the reference method, occlusions with foreground objects lead to discontinued trajectories and erroneous reconstructed 3D object point positions. The discontinued trajectories are connected by using a robust feature matching between non-consecutive frames. For this matching, the *scale invariant feature transform* (SIFT) is required. SIFT is known for its ability of establishing stable correspondences under varying viewpoint conditions such as lighting or perspective change between the camera views.

Due to the importance of feature localization accuracy in structure and motion recovery techniques, an accuracy analysis of the SIFT feature detector is performed. The analysis predicts the localization error of SIFT by applying the image signal transfer model of a camera and the *Difference of Gaussians* (DoG) pyramid which is used for the scale invariant feature detection. The localization error is dependent on the subpixel and, in particular, on the subscale position of the feature. It shows a spatial error of up to 0.14 px, only considering a subpixel shift in x direction and subscale shift. This error is derived using Gaussian blobs as input features. A new localization technique, called DoG SIFT, is presented. It is based on the image signal transfer model and shows an error of only 0.009 px on the Gaussian blobs with no dependency on the subpixel and subscale position.

To evaluate the impact of the new feature localization technique on natural images, a benchmark data set with ground truth intrinsic camera parameters and rectified images is used in a structure and motion recovery approach. Additionally, a turntable sequence is captured with calibrated intrinsic camera parameters. As only the feature localization technique is exchanged in the scene reconstruction pipeline, the reprojection error is used as a reasonable measure for the evaluation. The comparison between DoG SIFT and the reference SIFT localization technique shows an improvement of up to 14 % while increasing the number of reconstructed object point significantly by about 50 %.

The new feature localization technique is applied to the improved structure and motion recovery which applies *feature trajectory retrieval* (FTR). FTR combines the advantages of feature tracking for consecutive frames and feature matching for non-consecutive frames. The discontinued feature tracks are stored in memory and compared to newly

detected features for the retrieval. The successfully retrieved features are assigned to the previously discontinued trajectories and their 3D object points in sequential scene estimation.

The proposed FTR technique increases the mean trajectory length significantly compared to standard SIFT tracking and to KLT tracking. It enables the handling of occlusions and improves the scene reconstruction. It is demonstrated that the accuracy of the reconstruction increases although the reprojection error may lead to a larger error. This is shown by evaluating reconstructed point clouds of scenes with known geometry. The reconstruction accuracy is determined by the distance between the point cloud and a CAD model of the scene. The distance is decreased by up to 28 % using the combination of the DoG SIFT feature localization and FTR.

Besides the accurate estimation of the perspectively correct camera view, a new method for the automatic occlusion of integrated virtual objects is presented. The extracted trajectories with non-consecutive correspondences provide information which enables the application of automatic video segmentation. The video segmentation eases the integration of virtual objects between scene elements significantly. The demonstrated application is the correct occlusion of the integrated objects with the foreground of the captured scene.

The feature tracks with non-consecutive correspondences provide the cues for the classification of small image regions as foreground or background. Hereby, a new and intuitive definition of foreground is derived. The foreground is defined as image regions which occlude the background scene temporarily. For the representation of foreground and background, *Gaussian mixture models* (GMM) are estimated from the classified regions. The foreground regions result from the projection of 3D objects points into the camera views in which the object point is occluded. They are located inside the foreground objects and not on the object boundaries which is a very advantageous property for their usage in object segmentation. Due to the reprojection of 3D points *behind* the objects, the foreground object appearance represented by color values improves with each new camera view. The extraction of this information about the foreground is called *appearance learning from occlusions* (ALO). The representations of foreground and background are used to initialize a state of the art segmentation method which minimizes an energy term consisting of regional and boundary costs. For the efficient optimization, the image is represented as a two-dimensional grid graph. For the segmentation of video, the time domain is incorporated by using a three-dimensional grid to build the graph. The ALO provides information which increases the quality of the resulting video segmentation significantly.

Finally, demonstrations show two applications in the field of *virtual effect* (VFX) creation. While common approaches require user interaction, our results are obtained automatically and take advantage of the reconstructed scene and the video segmentation. The demonstrated applications are:

1. The integration of virtual objects between scene elements: it is shown that the integrated objects behave perspectively correct. The integrated objects are occluded correctly with foreground scene content.

2. The background blur effect which focuses the observer on the foreground object.

The resulting sequences are available for download at:
`http://www.tnt.uni-hannover.de/staff/cordes/`

Future research should combine the areas of video segmentation and motion segmentation with the presented approach. While state of the art methods provide impressive results for well textured foreground objects, their performance decreases when a foreground object is textureless and provides only a few trajectories or possibly many motion models. The presented method does not require any feature on the foreground objects. Thus, it is unique for the extraction of object appearance and demonstrates its strength for textureless and arbitrarily moving foreground objects. But, the performance of the presented approach depends on reliable feature trajectories extracted from the background.

The background is represented by SIFT features tracked with KLT. If the background scene is not sufficiently covered with features, the performance of the approach decreases. In order to increase the completeness of the background scene, complementary feature types should be incorporated.

8 Appendix

8.1 Hessian Matrix used for the SIFT detector

The approximated derivatives D_x, D_y, and D_i as used in Section 3.1.2 are computed as follows:

$$D_x := D_x(x,y,i) = \frac{1}{2}(D(x+1,y,i) - D(x-1,y,i)) \tag{8.1}$$

$$D_y := D_y(x,y,i) = \frac{1}{2}(D(x,y+1,i) - D(x,y-1,i)) \tag{8.2}$$

$$D_i := D_i(x,y,i) = \frac{1}{2}(D(x,y,i+1) - D(x,y,i-1)) \quad . \tag{8.3}$$

The approximated matrix entries of the hessian matrix as used in Section 3.1.2 are computed as follows:

$$D_{xx} := D_{xx}(x,y,i) = D(x+1,y,i) + D(x-1,y,i) - 2 \cdot D(x,y,i) \tag{8.4}$$

$$D_{yy} := D_{yy}(x,y,i) = D(x,y+1,i) + D(x,y-1,i) - 2 \cdot D(x,y,i) \tag{8.5}$$

$$D_{ii} := D_{ii}(x,y,i) = D(x,y,i+1) + D(x,y)(x,y,i-1) - 2 \cdot D(x,y)(x,y,i) \tag{8.6}$$

$$D_{xy} := D_{xy}(x,y,i) = \frac{1}{4}(D(x+1,y+1,i) + D(x-1,y-1,i)) \tag{8.7}$$

$$-\frac{1}{4}((D(x+1,y-1,i) - D(x-1,y+1,i)) \tag{8.8}$$

$$D_{xi} := D_{xi}(x,y,i) = \frac{1}{4}(D(x+1,y,i+1) + D(x-1,y,i-1)) \tag{8.9}$$

$$-\frac{1}{4}(D(x-1,y,i+1) - D(x+1,y,i-1)) \tag{8.10}$$

$$D_{yi} := D_{yi}(x,y,i) = \frac{1}{4}(D(x,y+1,i+1) + D(x,y-1,i-1)) \tag{8.11}$$

$$-\frac{1}{4}(D(x,y-1,i+1) - D(x,y+1,i-1)) \quad . \tag{8.12}$$

8.2 Repeatability Comparison of DoG SIFT and SIFT

A well-established criterion for the evaluation of feature detectors is the repeatability criterion. It is introduced in [70] together with a benchmark data set consisting of images

showing planar scenes and homographies. The repeatability is based on the *overlap error* which measures the percentage of the overlap of two ellipses surrounding the features in an image pair. The overlap is found by mapping the feature from one image to the other using the given homography. A corresponding feature pair is found if its overlap error is smaller than 40 % [70]. Then, the number of corresponding features is divided by the number of maximally possible number of correspondences.

8.2.1 Input Data and Experimental Setup

For the use of highly-accurate and high resolution image data and corresponding homographies, a benchmark is created in [21, 22]. It provides image resolutions of up to 8 megapixel. Like the standard benchmark [70], each sequence of the high resolution benchmark consists of six images and corresponding homographies between the reference image and each of the other five images. The images undergo an increasing level of perspective change. In this section, three of the sequences are shown: *Grace*, *There*, and *Underground*. The images are shown in Figure 8.1. The resolution of the images is 3456×2304. More repeatability evaluations can be found in [23].

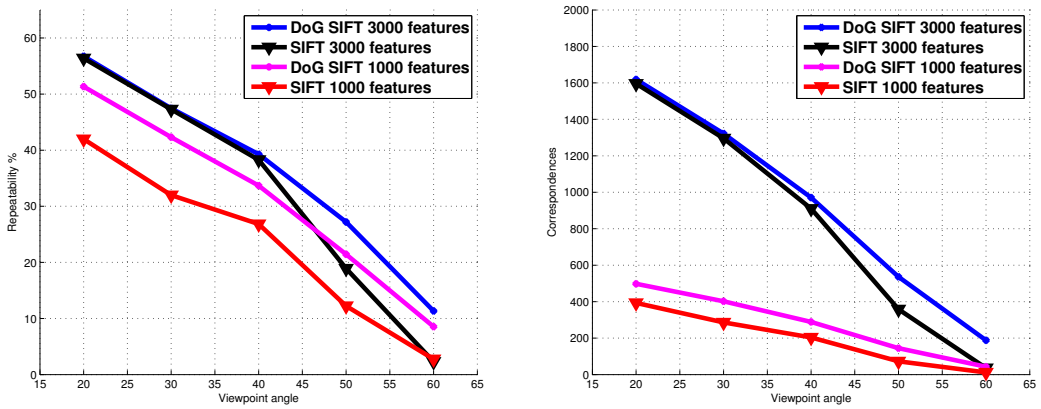
(a) *Grace*

(b) *There*

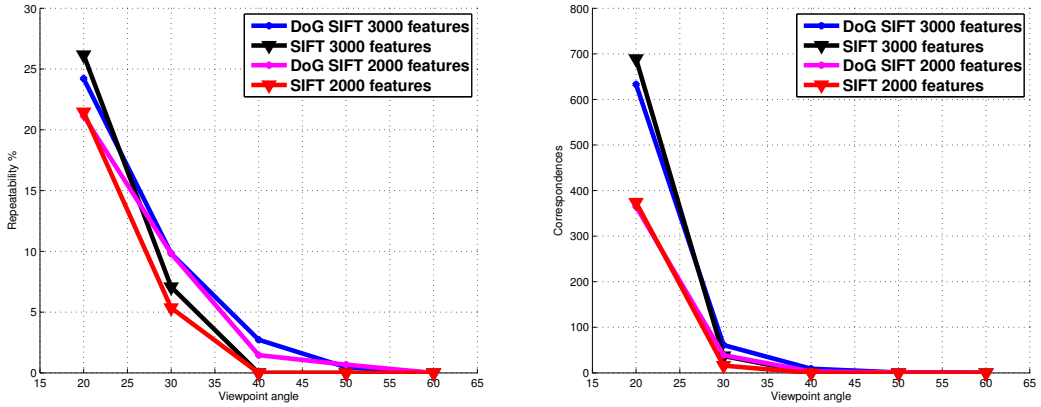
(c) *Underground*

Figure 8.1: Input data for repeatability evaluation [22]. The benchmark data set consists of planar scenes captured from different viewpoints and corresponding homographies which determine the mapping between image pairs.

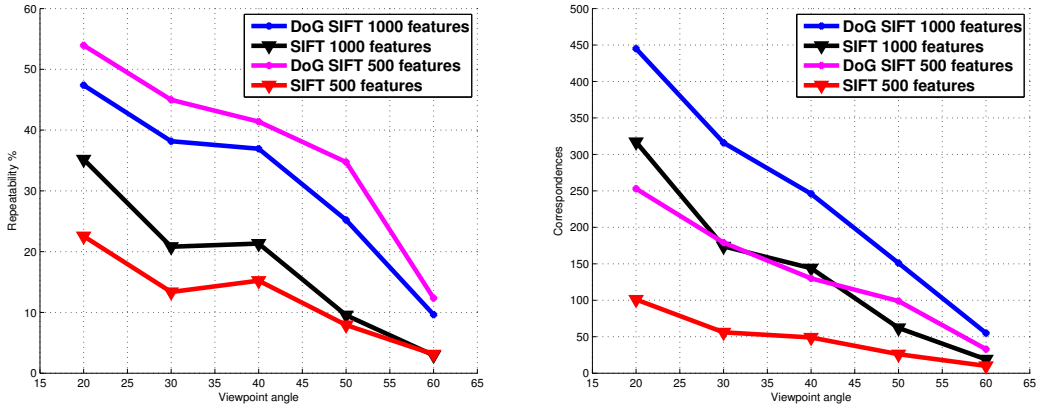
The features are selected by using the minimal residuum selection criterion for SIFT and DoG SIFT (cf. equation (3.8) for SIFT and equation (4.23) for DoG SIFT). The number of selected feature is chosen according to the total number of extracted features by SIFT in the image sequence. For the *Grace* sequence, between 3500 and 4500 are



(a) *Grace*



(b) *There*



(c) *Underground*

Figure 8.2: Repeatability comparison of SIFT and DoG SIFT feature localization. The presented DoG SIFT feature localization provides better repeatability compared to SIFT, especially for the strong viewpoint changes.

extracted for the six images using the standard SIFT parameters [61]. For *There*, between 5000 and 7000 are extracted. Here, the strongest perspective distortions are present in the images. For the *Underground* sequence with few texture, only between 1000 and 3000 features are detected. For *There* and *Underground*, the image doubling option is enabled [61] to provide enough features for the evaluation.

In contrast to the evaluations resulting from the SAM scenario, the estimated covariance matrix Σ (cf. equation 4.21) of the extracted features is incorporated in the error measure, the *overlap error* which is the basis for the repeatability rate.

8.2.2 Results

The repeatability results are presented in Figure 8.2. The repeatability rates are shown on the left while on the right the resulting number of corresponding feature pairs are shown. The localization method DoG SIFT is superior to SIFT in any of the cases except for the first image pair of *There*. For *There* and *Grace*, the main improvement is achieved for the strong viewpoint changes. Here, the repeatability rate improves by up to 9.2 %. For *Underground*, the improvement is present throughout the sequence. The gain in repeatability is up to 17.4 % for 1000 features and up to 31.6 % for the smaller set of 500 features.

Bibliography

[1] Nicholas Apostoloff and Andrew Fitzgibbon. Automatic video segmentation using spatiotemporal t-junctions. In *British Machine Vision Conference (BMVC)*, pages 111.1–111.10. BMVA Press, 2006.

[2] Kalle Aström and Anders Heyden. Stochastic modelling and analysis of sub-pixel edge detection. In *International Conference on Pattern Recognition (ICPR)*, pages 86–90, August 1996.

[3] Autodesk. 3D animation software maya.

[4] Autodesk. 3D studio max software.

[5] Sebastiano Battiato, Giovanni Maria Farinella, Enrico Messina, and Giovanni Puglisi. Understanding geometric manipulations of images through bovw-based hashing. In *International Conference on Multimedia and Expo (ICME)*, pages 1 –6, 2011.

[6] Adam Baumberg and Alper Yilmaz. Reliable feature matching across widely separated views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1774–1781, 2000.

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417, Berlin / Heidelberg, Germany, 2006. Springer.

[8] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez, and Philip H. S. Torr. Interactive image segmentation using an adaptive gmmrf model. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science, pages 428–441. Springer, 2004.

[9] Alan C. Bovik. Basic gray-level image processing. In Alan C. Bovik, editor, *Handbook of Image and Video Processing*. Academic Press, 2000.

[10] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 105 –112, 2001.

[11] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 26(9):1124 –1137, 2004.

[12] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 23(11):1222–1239, 2001.

[13] Matthew Brown and David G. Lowe. Invariant features from interest point groups. In *British Machine Vision Conference (BMVC)*, pages 656–665, 2002.

[14] Gertjan J. Burghouts and Jan-Mark Geusebroek. Performance evaluation of local colour invariants. *Comput. Vis. Image Underst.*, 113:48–62, 2009.

[15] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

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

[17] Kai Cordes, Patrick Mikulastik, Alexander Vais, and Jörn Ostermann. Extrinsic calibration of a stereo camera system using a 3D CAD model considering the uncertainties of estimated feature points. In *The 6th European Conference on Visual Media Production (CVMP)*, pages 135–143. IEEE Computer Society, 2009.

[18] Kai Cordes, Oliver Müller, Bodo Rosenhahn, and Jörn Ostermann. HALF-SIFT: High-accurate localized features for SIFT. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Feature Detectors and Descriptors: The State Of The Art and Beyond*, pages 31–38, Miami Beach, USA, 2009. IEEE Computer Society.

[19] Kai Cordes, Oliver Müller, Bodo Rosenhahn, and Jörn Ostermann. Bivariate feature localization for SIFT assuming a gaussian feature shape. In George Bebis, editor, *Advances in Visual Computing, 6th International Symposium (ISVC)*, volume 6453 of *Lecture Notes in Computer Science (LNCS)*, pages 264–275. Springer Berlin/Heidelberg, 2010.

[20] Kai Cordes, Oliver Müller, Bodo Rosenhahn, and Jörn Ostermann. Feature trajectory retrieval with application to accurate structure and motion recovery. In George Bebis, editor, *Advances in Visual Computing, 7th International Symposium (ISVC)*, volume 6938 of *Lecture Notes in Computer Science (LNCS)*. Springer Berlin/Heidelberg, 2011.

[21] Kai Cordes, Bodo Rosenhahn, and Jörn Ostermann. Increasing the accuracy of feature evaluation benchmarks using differential evolution. In *IEEE Symposium*

Series on Computational Intelligence (SSCI) - IEEE Symposium on Differential Evolution (SDE). IEEE Computer Society, 2011.

[22] Kai Cordes, Bodo Rosenhahn, and Jörn Ostermann. High-resolution feature evaluation benchmark. In *15th International Conference on Computer Analysis of Images and Patterns (CAIP)*, volume 8047 of *Lecture Notes in Computer Science (LNCS)*, pages 327–334. Springer Berlin/Heidelberg, 2013.

[23] Kai Cordes, Bodo Rosenhahn, and Jörn Ostermann. Localization accuracy of interest point detectors with different scale space representations. In *11th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*. IEEE Computer Society, 2014. to be published.

[24] Kai Cordes, Björn Scheuermann, Bodo Rosenhahn, and Jörn Ostermann. Learning object appearance from occlusions using structure and motion recovery. In Kyoung Mu Lee, Jim Rehg, Yasuyuki Matsushita, and Zhanyi Hu, editors, *The 11th Asian Conference on Computer Vision (ACCV)*, volume 7726 of *Lecture Notes in Computer Science (LNCS)*, pages 611–623. Springer Berlin/Heidelberg, 2012.

[25] Kai Cordes, Björn Scheuermann, Bodo Rosenhahn, and Jörn Ostermann. Occlusion handling for the integration of virtual objects into video. In Gabriela Csurka and José Braz, editors, *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 173–180, 2012.

[26] Kai Cordes, Björn Scheuermann, Bodo Rosenhahn, and Jörn Ostermann. Foreground segmentation from occlusions using structure and motion recovery. In José Braz, Gabriela Csurka, Paul Richard, Martin Kraus, and Robert S. Laramee, editors, *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, volume 359 of *Communications in Computer and Information Science (CCIS)*, pages 340–353. Springer Berlin/Heidelberg, 2013.

[27] Kai Cordes, Oliver Topic, Manuel Scherer, Carsten Klempt, Bodo Rosenhahn, and Jörn Ostermann. Classification of atomic density distributions using scale invariant blob localization. In Mohamed Kamel and Aurélio Campilho, editors, *Image Analysis and Recognition, 8th International Conference (ICIAR)*, volume 6753 of *Lecture Notes in Computer Science (LNCS)*, pages 161–172. Springer Berlin/Heidelberg, 2011.

[28] Kurt Cornelis, Frank Verbiest, and Luc Van Gool. Drift detection and removal for sequential structure from motion algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26:1249–1259, 2004.

[29] Timo Dickscheid, Falko Schindler, and Wolfgang Förstner. Coding images with local features. *International Journal of Computer Vision (IJCV)*, 94(2):154–174, 2011.

[30] Ralf Dragon, Bodo Rosenhahn, and Jörn Ostermann. Multi-scale clustering of frame-to-frame correspondences for motion segmentation. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *European Conference on Computer Vision (ECCV)*, volume 7573 of *Lecture Notes in Computer Science*, pages 445–458. Springer Berlin/Heidelberg, 2012.

[31] Chris Engels, Friedrich Fraundorfer, and David Nistér. Integration of tracked and recognized features for locally and globally robust structure from motion. In *VISAPP (Workshop on Robot Perception)*, pages 13–22, 2008.

[32] Oliver Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.

[33] Oliver Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images : The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, 2001.

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

[35] Andrew W. Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In Hans Burkhardt and Bernd Neumann, editors, *European Conference on Computer Vision (ECCV)*, volume 1406 of *Lecture Notes in Computer Science (LNCS)*, pages 311–326. Springer Berlin / Heidelberg, 1998.

[36] Wolfgang Förstner, Timo Dickscheid, and Falko Schindler. Detecting interpretable and accurate scale-invariant keypoints. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2256–2263, Kyoto, Japan, 2009.

[37] Blender Foundation. 3d animation software blender.

[38] Jan-Michael Frahm, Pierre Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, and Marc Pollefeys. Building rome on a cloudless day. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *European Conference on Computer Vision (ECCV)*, volume 6314 of *Lecture Notes in Computer Science (LNCS)*, pages 368–381. Springer, 2010.

[39] Jan-Michael Frahm, Marc Pollefeys, Svetlana Lazebnik, David Gallup, Brian Clipp, Rahul Raguram, Changchang Wu, Christopher Zach, and Tim Johnson. Fast robust large-scale mapping from video and internet photo collections. *Journal of Photogrammetry and Remote Sensing (ISPRS)*, 65(6):538 – 549, 2010. ISPRS Centenary Celebration Issue.

[40] Andrea Fusiello, Emanuele Trucco, Tiziano Tommasini, and Vito Roberto. Improving feature tracking with robust statistics. *Pattern Analysis and Applications*, 2:312–320, 1999.

[41] Joseph W. Goodman. *Introduction to Fourier Optics*. McGraw-Hill, 1968.

[42] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[43] Richard Hartley and Andrew Zisserman. *Multiple View Geometry*. Cambridge University Press, second edition, 2003.

[44] Nils Hasler, Bodo Rosenhahn, Thorsten Thormählen, Michael Wand, and Hans-Peter Seidel. Markerless motion capture with unsynchronized moving cameras. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 224–231, 2009.

[45] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *European Conference on Computer Vision (ECCV)*, volume 6311 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin/Heidelberg, 2010.

[46] Rob Hess. An open-source SIFT library. In *Proceedings of the International Conference on Multimedia*, MM '10, pages 1493–1496, New York, NY, USA, 2010. ACM.

[47] Peter Hillman, J.P. Lewis, Sebastian Sylwan, and Erik Winquist. Issues in adapting research algorithms to stereoscopic visual effects. In *IEEE International Conference on Image Processing (ICIP)*, pages 17 –20, 2010.

[48] Raphael Höver. Echtzeitfähige Schätzung von Merkmalspositionen mit Subpel-Genauigkeit. Master's thesis, Universität Hannover, Germany, 2006.

[49] Thomas S. Huang and Oliver Faugeras. Some properties of the e-matrix in two-view motion estimation. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 11(12):1310–1312, 1989.

[50] Ramakrishna Kakarala and Alfred O. Hero. On achievable accuracy in edge localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(7):777 –781, 1992.

[51] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *International Conference on Computer Vision and Pattern Recognition (ICCV)*, pages 506–513, 2004.

[52] Pushmeet Kohli, Victor Lempitsky, and Carsten Rother. Uncertainty driven multi-scale optimization. In *Proceedings of the 32nd DAGM conference on Pattern recognition*, pages 242–251, Berlin, Heidelberg, 2010. Springer-Verlag.

[53] Christian Kurz, Thorsten Thormählen, Bodo Rosenhahn, and Hans-Peter Seidel. Exploiting mutual camera visibility in multi-camera motion estimation. In George Bebis, editor, *Advances in Visual Computing, 5th International Symposium (ISVC)*, volume 5875 of *Lecture Notes in Computer Science (LNCS)*, pages 391–402, 2009.

[54] MIT Media LAB. *Vision texture library*. http://www-white.media.mit.edu/vismod/imagery/VisionTexture., 1995.

[55] Tony Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision (IJCV)*, 11:283–318, 1993.

[56] Tony Lindeberg. Feature detection with automatic scale selection. *Technical report*, 1994. ISRN KTH NA/P–94/05–SE.

[57] Tony Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):224–270, 1994.

[58] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision (IJCV)*, 30:79–116, 1998.

[59] Tony Lindeberg and Jonas 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.

[60] Jun Liu and Roger Hubbold. Automatic camera calibration and scene reconstruction with scale-invariant features. In George Bebis, editor, *Advances in Visual Computing, 2nd International Symposium (ISVC)*, volume 4291 of *Lecture Notes in Computer Science (LNCS)*, pages 558–568. Springer, 2006.

[61] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[62] Wenjun Lu and Min Wu. Multimedia forensic hash based on visual words. In *International Conference on Image Processing (ICIP)*, pages 989 –992, 2010.

[63] Bruce Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[64] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference (BMVC)*, volume 1, pages 384–393, 2002.

[65] Roland Mech. *Schätzung der 2D-Form bewegter Objekte in Bildfolgen unter Nutzung von Vorwissen und einer Aperturkompensation*. Thesis/dissertation, Leibniz Universität Hannover, Germany, 2003.

[66] Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 525 –531, 2001.

[67] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision (ECCV)*, pages 128–142. Springer-Verlag, 2002.

[68] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision (IJCV)*, 60(1):63–86, 2004.

[69] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–1630, 2005.

[70] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(1-2):43–72, 2005.

[71] Patrick Mikulastik. *Verbesserung der Genauigkeit der Selbstkalibrierung einer Stereokamera mit 3D-CAD-Modellen*. Thesis/dissertation, Leibniz Universität Hannover, Germany, 2009.

[72] Patrick Mikulastik, Hellward Broszio, Thorsten Thormählen, and Onay Urfalıoğlu. Error analysis of feature based disparity estimation. In *Advances in Image and Video Technology (PSIVT)*, volume 4319, pages 1–12. Springer, 2006.

[73] Patrick Mikulastik, Raphael Höver, and Onay Urfalıoğlu. Error analysis of subpixel edge localization. In *The International Conference on Signal-Image Technology & Internet Based Systems*, 2006.

[74] Federico Pedersini, Augusto Sarti, and Stefano Tubaro. Estimation and compensation of subpixel edge localization error. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 19(11):1278 – 1284, 1997.

[75] Haim Permuter, Joseph Francos, and Ian Jermyn. Gaussian mixture models of texture and colour for image database retrieval. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 3, pages III – 569–572, 2003.

[76] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision (IJCV)*, 59(3):207–232, 2004.

[77] Peter Rockett. The accuracy of sub-pixel localisation in the canny edge detector. In *British Machine Vision Conference (BMVC)*, pages 392–401, 1999.

[78] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: interactive foreground extraction using iterated graph cuts. *ACM SIGGRAPH Papers*, 23(3):309–314, 2004.

[79] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2195 – 2202, 2006.

[80] Björn Scheuermann and Bodo Rosenhahn. Slimcuts: Graphcuts for high resolution images using graph reduction. In Yuri Boykov, Fredrik Kahl, Victor S. Lempitsky, and Frank R. Schmidt, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, volume 6819 of *Lecture Notes in Computer Science (LNCS)*. Springer, 2011.

[81] Börn Scheuermann, Markus Schlosser, and Bodo Rosenhahn. Efficient pixel-grouping based on dempster's theory of evidence for image segmentation. In KyoungMu Lee, Yasuyuki Matsushita, JamesM. Rehg, and Zhanyi Hu, editors, *Asian Conference on Computer Vision (ACCV)*, volume 7724 of *Lecture Notes in Computer Science*, pages 745–759. Springer Berlin Heidelberg, 2013.

[82] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Comparing and evaluating interest points. *IEEE International Conference on Computer Vision (ICCV)*, pages 230–235, 1998.

[83] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision (IJCV)*, 37(2):151–172, 2000.

[84] Yaser Sheikh, Omar Javed, and Takeo Kanade. Background subtraction for freely moving cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1219–1225, 2009.

[85] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.

[86] Mikhail Sindeeva, Anton Konushin, and Carsten Rother. Alpha flow for video matting. In Kyoung Mu Lee, Jim Rehg, Yasuyuki Matsushita, and Zhanyi Hu, editors, *The 11th Asian Conference on Computer Vision (ACCV)*, volume 7726 of *Lecture Notes in Computer Science (LNCS)*. Springer Berlin/Heidelberg, 2012.

[87] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH Conference Proceedings*, pages 835–846, New York, NY, USA, 2006. ACM Press.

[88] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision (IJCV)*, 80:189–210, 2008.

[89] Christoph Strecha, Wolfgang von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1 –8, 2008.

[90] Thorsten Thormählen. *Zuverlässige Schätzung der Kamerabewegung aus einer Bildfolge*. Thesis/dissertation, Leibniz Universität Hannover, Germany, 2006.

[91] Thorsten Thormählen and Hellward Broszio. Automatic line-based estimation of radial lens distortion. *Integrated Computer-Aided Engineering*, 12(2):177–190, 2005.

[92] Thorsten Thormählen, Hellward Broszio, and Ingolf Wassermann. Robust line-based calibration of lens distortion from a single view. In *Proceedings of Mirage 2003 (Computer Vision / Computer Graphics Collaboration for Model-based Imaging, Rendering, Image Analysis and Graphical Special Effects)*, pages 105–112, 2003.

[93] Thorsten Thormählen, Nils Hasler, Michael Wand, and Hans-Peter Seidel. Registration of sub-sequence and multi-camera reconstructions for camera motion estimation. *Journal of Virtual Reality and Broadcasting*, 7(2), 2010.

[94] Philip H. S. Torr. An assessment of information criteria for motion model selection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 47–52, 1997.

[95] Philip H. S. Torr, Andrew W. Fitzgibbon, and Andrew Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *International Journal of Computer Vision (IJCV)*, 32:27–44, 1999.

[96] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, London, UK, 2000. Springer-Verlag.

[97] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3-D machine vision metrology using off-the-shelf cameras and lenses. *IEEE Transaction on Robotics and Automation*, 3(4):323–344, 1987.

[98] Tinne Tuytelaars and Krystian Mikolajczyk. *Local invariant feature detectors: a survey*, volume 3. Foundations and Trends in Computer Graphics and Vision, 2008.

[99] Onay Urfalioglu, Patrick Mikulastik, and Ivo Stegmann. Scale invariant robust registration of 3D-point data and a triangle mesh by global optimization. In *Proceedings of the 8th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, volume 4179 of *Lecture Notes in Computer Science (LNCS)*, pages 1059–1070, 2006.

[100] Anton van den Hengel, Anthony Dick, Thorsten Thormählen, Ben Ward, and Philip H. S. Torr. Videotrace: rapid interactive scene modelling from video. In *ACM SIGGRAPH 2007 papers*, number 86 in SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[101] Guoshen Yu and Jean-Michel Morel. A fully affine invariant image comparison method. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1597–1600, Washington, DC, USA, 2009. IEEE Computer Society.

[102] Guofeng Zhang, Zilong Dong, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Efficient non-consecutive feature tracking for structure-from-motion. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *European Conference on Computer Vision (ECCV)*, volume 6315 of *Lecture Notes in Computer Science (LNCS)*, pages 422–435. Springer, 2010.

L E B E N S L A U F

PERSÖNLICHE ANGABEN

Name: Kai Cordes

Familienstand: ledig, 2 Kinder

Geburtsdatum: 17.12.1973

Geburtsort: Nienburg

SCHULE & STUDIUM

1993	Abitur in Nienburg
1993 – 1994	Zivildienst Paritätischer Wohlfahrtsverband, Nienburg
1994 – 2003	Studium Mathematik mit Studienrichtung Informatik, Leibniz Universität Hannover
Juni 2002	Studienarbeit: *Untersuchung von Verfahren zur blockbasierten Textursynthese*
August 2003	Diplomarbeit: *Übertragung eines Textursyntheseverfahrens auf die Textursegmentierung von Fernerkundungsdaten*, ausgezeichnet als eine der besten Diplomarbeiten 2003

BERUF

2003 – 2004	Hilfswissenschaftler am Institut für Informationsverarbeitung (TNT)
2004 –	Wissenschaftlicher Mitarbeiter am Institut für Informationsverarbeitung (TNT)