

Random Sampling Methods for Two-View Geometry Estimation

PROEFSCHRIFT

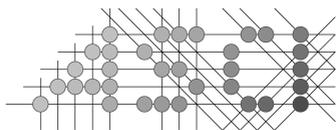
ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 10 mei 2007 om 10:00 uur
door Richard Jacobus Maria DEN HOLLANDER
elektrotechnisch ingenieur
geboren te Leiderdorp.

Dit proefschrift is goedgekeurd door de promotor:
Prof.dr.ir. R.L. Lagendijk

Toegevoegd promotor:
Dr. A. Hanjalić

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. R.L. Lagendijk,	Technische Universiteit Delft, promotor
Dr. A. Hanjalić,	Technische Universiteit Delft, toegevoegd promotor
Prof.dr. F.M. Dekking,	Technische Universiteit Delft
Prof.dr.ir. F.C.A. Groen,	Universiteit van Amsterdam
Prof.dr. M. Schaepman,	Universiteit Wageningen
Prof.dr. R. Leonardi,	University of Brescia, Italië
Dr. R.C. Veltkamp,	Universiteit Utrecht
Prof.dr.ir. J. Biemond,	Technische Universiteit Delft, reservelid



Advanced School for Computing and Imaging

This work was carried out in the ASCI graduate school.
ASCI dissertation series number 142.

The printing of this thesis was financially supported by TNO Defence, Security and Safety.

ISBN 978-90-9021839-7

Copyright © 2007 by R.J.M. den Hollander

All rights reserved. No part of this thesis may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, any information storage or retrieval system, or otherwise, without written permission from the copyright owner.

Random Sampling Methods for Two-View Geometry Estimation

Contents

1	Introduction	1
1.1	From image features to imaging geometry	4
1.2	Scope	5
1.3	Thesis outline	7
2	Local Features and the Epipolar Geometry	9
2.1	Interest points	9
2.2	Local descriptors	13
2.3	Feature comparison and local geometric constraints	15
2.4	The epipolar geometry	17
2.4.1	Camera projection	17
2.4.2	The fundamental matrix	19
2.4.3	Robust fundamental matrix estimation	21
2.5	Complexity of epipolar geometry estimation	27
2.6	Discussion	28
3	A Priori Inlier Probabilities for Correspondences based on Quadrics	31
3.1	Introduction	31
3.2	Quadric comparisons	32
3.3	Complexity	36
3.4	Experimental results	36
3.4.1	Synthetic data	36
3.4.2	Real data	39
3.5	Discussion	40
4	Weighted Data Sampling in the RANSAC Algorithm	41
4.1	Introduction	41
4.2	Weighted sampling	42
4.2.1	Method of point weighting	42
4.2.2	Analysis of the effective inlier ratio	43

4.2.3	Hyperplane estimation	46
4.3	Experimental results	49
4.3.1	Line fitting	50
4.3.2	Plane fitting	51
4.3.3	Fundamental matrix estimation	57
4.4	Discussion	62
5	A Six-Point RANSAC Algorithm for Fundamental Matrix Estimation	65
5.1	Introduction	65
5.2	Hyperplane estimation	66
5.3	Fundamental matrix estimation	69
5.4	Implementation details	75
5.5	Experimental results	77
5.5.1	Plane fitting	77
5.5.2	Fundamental matrix estimation	79
5.6	Discussion	82
6	Discussion	85
6.1	Correspondence search	85
6.2	Non-uniform sampling	86
6.3	Zero or multiple epipolar geometries	86
6.4	Algorithm evaluation	87
A	Homogeneous Coordinates	89
B	Two-View Data Sets	91
C	List of Symbols	97
	Bibliography	99
	Summary	107
	Samenvatting	111
	Acknowledgements	115
	Curriculum Vitae	117

Chapter 1

Introduction

The automatic interpretation of a scene in an image is the ultimate goal in computer vision. It is an ambitious goal, since the aim is essentially to create a human-like system. Humans, as we all know, have a remarkable ability in processing and interpreting visual information. We can make the transition from the pictorial information to the meaning of the image content in milliseconds. The scene is decomposed into separate objects and each is compared to collected information from past experiences. Without a serious effort we are able to identify the objects and reason about their structures.

The structure of objects is essential in the recognition process. Our experience tells us in what way structures may change when viewed under different conditions. For example, an object may be imaged from another viewpoint or the amount of illumination may change. Furthermore, other objects may partially occlude the object of interest. All these factors should be taken into account when an automated system, i.e. a camera connected to a computer, is designed to interpret or recognize a scene.

Modeling of the imaging process inside the camera seems to be the first step in designing such a system. A camera model describes the projection of 3D world points onto 2D image coordinates. A combined model of two differently positioned cameras will reveal the relation between their projections. This model of the two views is called the epipolar geometry and is described by the fundamental matrix [15, 24]. It gives the relation between different projections of a single world point, see Fig. 1.1. The epipolar geometry relates all possible image pairs that can be made from a particular object. It therefore provides a valuable ingredient for computer algorithms interpreting the camera images.

One group of algorithms making use of the epipolar geometry is directed towards analysis of the scene. Object recognition approaches [75, 8] can compare in this way the configurations of image points in different images. When the point projections in two images allow a common camera configuration, we can affirm the presence

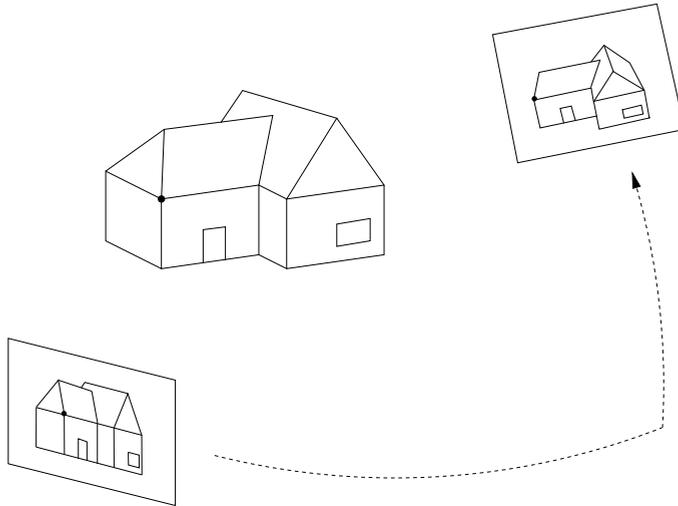


Figure 1.1: An object viewed from different directions. The relation between the projections of a 3D point (e.g. the black dot) is described by the epipolar geometry. The pair of point projections forms a correspondence.

of the same 3D object in the images. This automated method is useful for visual retrieval [21, 7, 76, 22], where the goal is to search large image or video archives for particular content. Especially in the “query by example” scenario, where an image example of the desired target image(s) is provided, object recognition techniques are applicable (see Fig. 1.2). In [72] the epipolar geometry is used to find similar scenes throughout a movie. This has its application in the grouping of movie frames, such that the logical structure of a movie can be deduced. Another example is the use of the epipolar geometry for location matching during visual navigation [20].

A second group of algorithms uses the epipolar geometry to find the structure of an imaged scene (see Fig. 1.3). Two images of a scene are sufficient to perform reconstruction of that scene up to some projective transformation [15, 27]. In case a whole sequence of images from a scene is available, this will allow an even better estimate of the structure. In so called “structure from motion” algorithms, epipolar geometries in an image sequence are used to establish a scene structure for 3D modeling [78, 67, 85] or robot navigation [6, 54]. Also three-view relationships between images are used for scene reconstruction [5, 17], though their computation usually relies on finding epipolar geometries first.

In general, it is important that the epipolar geometry can be estimated efficiently, both in case of visual retrieval [75] and scene reconstruction [63]. When scene reconstruction algorithms are required to run in real-time, e.g. for robot navigation,

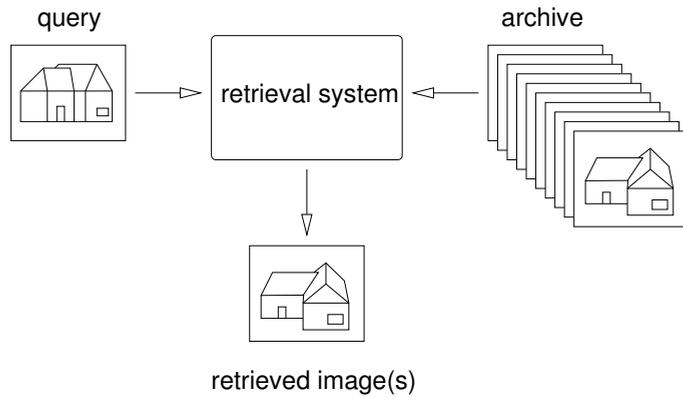


Figure 1.2: A retrieval system compares the query image with the images in the archive, and outputs any similar archive images.

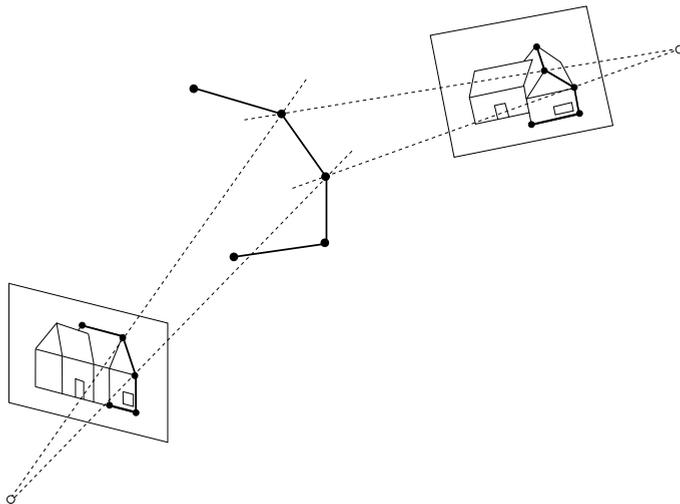


Figure 1.3: The epipolar geometry can be used to reconstruct a scene, using the rays crossing the camera's optical center (white dot) and the image points. At the intersection point of two rays lies the 3D scene point.

they impose strict time constraints on the computational routines. In a sequence of images, the epipolar geometry is estimated between all pairs of consecutive frames and is therefore a constantly recurring routine.

In case of visual retrieval, the amount of data being considered is often substantial. A retrieval system should be able to process the user's requests in a small amount of time. Efficient estimation of the epipolar geometry can speed up the finding of two similar scenes. Detecting the absence of an epipolar geometry, in case of images from different scenes, will also be very useful; it is a situation which occurs frequently in image retrieval. It is desirable that this can be detected quickly as well.

1.1 From image features to imaging geometry

When there is no information available on the way an image pair has been captured, the epipolar geometry for the particular pair is unknown. The sole source of information available is the image pair itself, and features in the images should be found which can relate their contents. The fundamental matrix can be determined from a set of *correspondences*, which are pairs of 2D projections from 3D points (like the dots in Fig. 1.1). To find such correspondences, we can compare local image neighborhoods and search for regional similarities. There has been done much work on the development of local features to obtain reliable correspondences across two different views of an object.

For local features it is important to accurately determine their positions of extraction in the image; the features should be computed at exactly the same positions on the object. This can be achieved by using interest points, which are positions in the images where the image structure is somewhat special. A part of an object which causes an interest point in one image, is likely to cause it in another image too. In this way, a subset of interest points in one image will have corresponding points in the other image, covering identical parts of an object.

After extraction of local features at the interest points (see Fig. 1.5), the features can be matched to obtain a set of initial correspondences. Each feature in the first image is compared to all the features in the second image, and the feature giving highest similarity is used to form a correspondence. Some of these will be true correspondences and connect identical object parts, but others may be incorrect due to accidental similarities of non-corresponding features. The true correspondences are called *inliers* with respect to the true fundamental matrix relation (the model describing the epipolar geometry). False correspondences are called *outliers*; they may be arbitrarily far off from this model, see Fig. 1.4.

To derive the fundamental matrix from the set of initial correspondences, we need to identify a sufficient number of inliers in the data. This can be accomplished through the use of robust estimators, which can find the inlier model even under the presence of many outliers. Unfortunately, they make fundamental matrix estimation much more complex. The goal of this thesis is to develop methods which reduce the

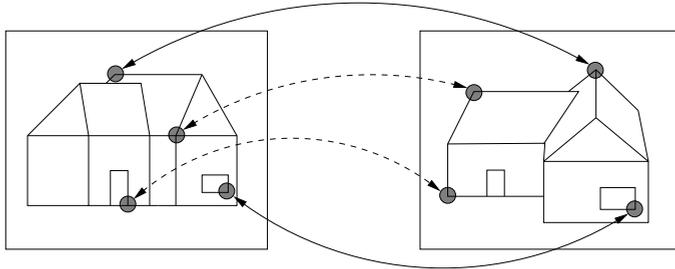


Figure 1.4: *Inlying (solid) and outlying (dotted) correspondences between local features.*

complexity of this estimation process.

1.2 Scope

There exist several different robust estimators which all have their own mechanism of dealing with outliers. The methods in this thesis are designed for one particular robust estimator: the RANSAC (RANDOM SAMPLE CONSENSUS) algorithm [16]. This algorithm has become the standard for fundamental matrix estimation under the presence of outliers. It randomly samples data points for the generation of models, and subsequently finds the model with most support points among the data.

The strength of the RANSAC algorithm is the sampling of minimal point sets for generating models; some of the computed models may then be unaffected by outliers. At the same time, however, the random sampling stage is very time-consuming. It may take many random samples before a set of inliers is found. For this reason, two methods (Chapter 3 and 4) have been developed to guide the sampling process. Instead of a purely random search, the search will be directed by using non-uniform sampling probabilities for the correspondences.

Another reason for the lengthy sampling stage, is the size of the sampled point set. The larger the point set, the longer it takes before an inlier is sampled for each of the points simultaneously. Therefore, a method (Chapter 5) has been developed that samples smaller-than-minimal point sets for estimating the fundamental matrix.

Although the proposed methods are directed towards fundamental matrix estimation, some of them have a larger extent. In particular, the methods in Chapter 4 and 5 are also applicable to hyperplane estimation. Line and plane fitting in 2D and 3D, respectively, will serve as examples in these chapters.

The methods described in Chapter 3 and 4 assume the presence of a single (dominant) epipolar relation. This holds when there is no independent movement of objects between two recordings. If objects do move independently, then multiple

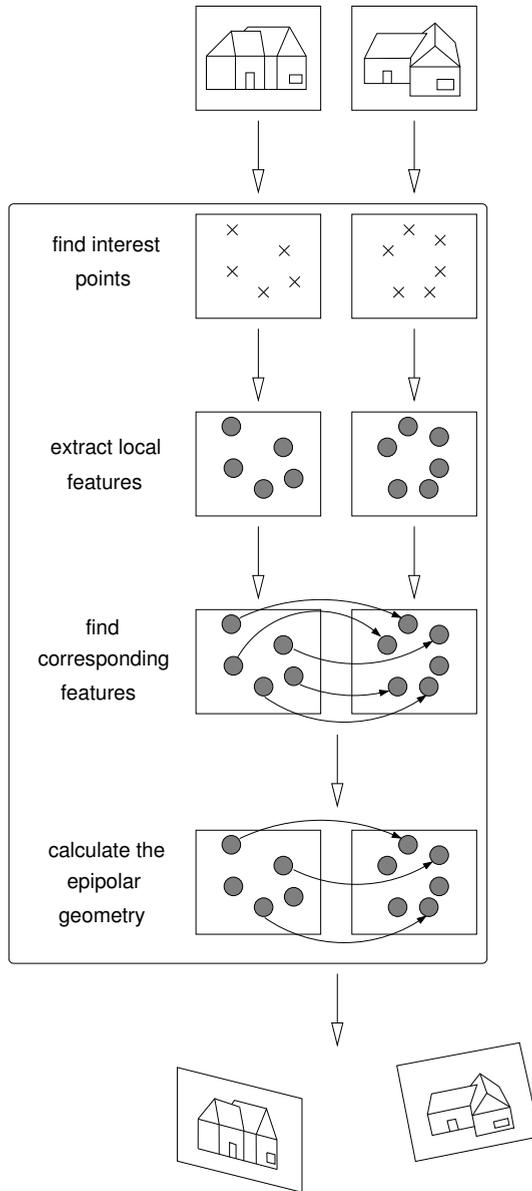


Figure 1.5: The sequence of steps necessary to find the epipolar geometry. The input is a pair of images showing a common scene, and the output is information about the camera configuration used for capturing the scene (see also Fig. 1.1).

epipolar relations are present; one for each of the moving objects. The method proposed in Chapter 5 may also be used for sequential extraction of multiple epipolar geometries.

1.3 Thesis outline

This thesis starts with an introduction to the field of epipolar geometry estimation. Then three different methods are presented that contribute to a more efficient estimation process.

In Chapter 2, a detailed description of the steps in Fig. 1.5 will be given. Different types of interest points and local features are discussed, along with their specific invariance properties. The features are matched to find a set of initial correspondences, which is likely to contain a certain percentage of outliers. The imaging process will be described and the concept of the epipolar geometry. It is shown how to calculate the epipolar geometry from the correspondences, and in what way robust estimators can deal with the outliers. Several robust estimators will be discussed, among which the RANSAC algorithm.

Chapter 3 will discuss methods for producing *a priori* inlier confidence values for the correspondences. These can help in determining which subset of correspondences is most likely to produce the true fundamental matrix. An algorithm is proposed which is capable of collecting such confidence values very quickly. It considers the relative position of correspondences in the images, and derives through a series of comparisons the probable set of inliers. Afterwards, the RANSAC algorithm can use the confidence values in the sampling stage.

The confidence values can also be acquired during fundamental matrix estimation, as is shown in Chapter 4. Here an adjustment of the RANSAC algorithm itself is proposed, which iteratively modifies the sampling probabilities of correspondences based on intermediate results from the estimation process. Experiments will show that this stimulates the sampling of inliers, and that the model with most support points is found more quickly.

A more fundamental adjustment of RANSAC is made in Chapter 5, where a combination of different model estimation methods is used to find the fundamental matrix. In particular, the generation of model hypotheses is applied together with a parameter voting step. It is shown that by a suitable definition of the parameter space, the fundamental matrix can be estimated using fewer correspondences. This is beneficial since it decreases the chance of having outliers in the sample.

The final chapter will present possible extensions of the proposed methods and topics for future research.

Chapter 2

Local Features and the Epipolar Geometry

In this chapter we will describe the process of epipolar geometry estimation from a pair of images. We first illustrate the use of interest points and the extraction of regional descriptors at these points. The extracted descriptors or features are then compared across the images to obtain a set of preliminary correspondences.

We then give an illustration of the process of image capture by a camera. This leads us to the concept of the epipolar geometry, which is the relation between images of a scene captured by two different cameras. It is indicated how we can find this relation from a set of correspondences, especially under the presence of outliers. Finally, we discuss complexity issues of the different stages of epipolar geometry estimation.

2.1 Interest points

The complexity of extracting and comparing features from all image positions is too high, especially when pixel increments are used. For this reason, a subset of positions in the images is selected, where the images shows some specific recognizable structure. This forms the collection of interest points. When the imaging conditions do not change too much, the same object part is likely to cause interest points in several differently captured images.

One of the first interest point detectors proposed was the Harris corner detector [23]. Corners are points in the image where the intensity changes substantially in two mutually orthogonal directions. When the original intensity image is denoted by $I(x, y)$, the Harris detector finds corners by using the second moment matrices

$$C(x, y) = g(x, y, \sigma) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.1)$$

where the elements contain the partial derivatives $I_x = \frac{\partial}{\partial x} I(x, y)$ and $I_y = \frac{\partial}{\partial y} I(x, y)$. The derivatives are approximated in the detector by convolution with $[-1 \ 0 \ 1]$ in horizontal and $[-1 \ 0 \ 1]^T$ in vertical direction, respectively. The values in $C(x, y)$ are obtained by averaging the I_x^2 , I_y^2 and $I_x I_y$ in a local region with the Gaussian

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.2)$$

for some reasonable choice of σ . A corner is detected when the measure

$$\det(C(x, y)) - \kappa \text{trace}^2(C(x, y)) \quad (2.3)$$

exceeds a threshold, where κ is some small constant. The measure roughly equals the determinant, and thus the product of the eigenvalues. Two large eigenvalues will thus indicate the presence of a corner point. An example of the function (2.3) is shown in Fig. 2.1 for a 800×640 sized image. Most corners are found at the window corners of the building, which have abrupt intensity changes.



Figure 2.1: The Harris corner measure for a test image. White areas indicate high “cornerness” of the region.

An important issue that remains in corner detection is the selection of a scale for the derivative operators I_x and I_y . The Harris detector just takes the derivatives of $I(x, y)$ at the highest resolution, yielding corners at detailed structures. A multi-scale approach may also result in corners that are caused by larger structures in the image. If the imaged objects have different scales, it is necessary to find corners of varying size to be able to establish correspondences. Different scales of the image are acquired by convolution with a Gaussian through $I(x, y, \sigma) = g(x, y, \sigma) * I(x, y)$, giving an

increasingly blurred image (see [46] for an overview). The derivative operation then follows as

$$\frac{\partial}{\partial x} I(x, y, \sigma) = \frac{\partial}{\partial x} g(x, y, \sigma) * I(x, y) \quad (2.4)$$

where the convolution is taken with the Gaussian derivative. Application of the Harris detector for different such scales may result in other interest point locations. Fig. 2.2 shows the corner measure for a Gaussian derivative of $\sigma = 5$; the corners now belong to the blurred test image. Note that the corners are mostly found at larger structures of the building; the corners at the windows are not detected anymore.

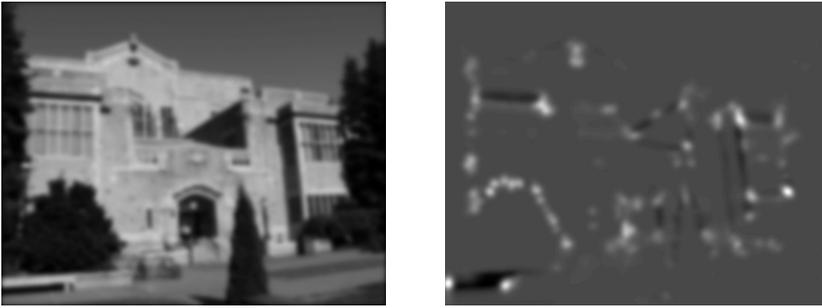


Figure 2.2: The image in Fig. 2.1 blurred with a Gaussian ($\sigma = 5$) and its Harris corner measure.

When all scales have to be considered, the number of interest points may become very large. As alternative, a characteristic scale can be found at which an interest point is extracted. A characteristic scale is the intrinsic scale of the local image structure [47]. This intrinsic scale can be retrieved for images of different resolutions, and should conform with the scale difference between them. As shown in [47], the characteristic scale can be found by looking for a local extremum of the normalized Laplacian operator. In [56] this concept is used to find scale-invariant interest points. The Harris detector is used here to find interest points at all scales, but only the interest points for which the Laplacian achieves a local maximum are retained.

The corner detector can be further enhanced by taking affine transformations of the image region into account. To acquire an accurate position, the area used for averaging the derivatives should be similar for all images. However, the object's position and pose may change in between images, and a circular neighborhood does not necessarily cover the same part of the object. If we view the object locally, the transformation of the object's surface can also be viewed locally. Since the object can be assumed to be locally planar, the transformation of the image neighborhood is a projective transformation. In case these small planes are relatively far away from

the camera when compared to their depth variation, the projective transformation can be approximated by an affine transformation [82]. It is therefore sufficient to find an image neighborhood that is invariant under an affine transformation. An iterative algorithm accomplishing this for the Harris detector is given in [57].

Next to corner point detection, several other interest point detectors have been proposed. A similar approach to detecting corner points and filtering them on the characteristic scale, is evaluating a suitable function directly over scale space. Extrema of such a function can be used to find potential interest points. In [47] the normalized Laplacian was used for finding scale-space extrema, which results in intensity blobs at different scales for interest points. A function that is somewhat similar to the normalized Laplacian, is the direct evaluation of differences over image scales [50]

$$D(x, y, \sigma) = g(x, y, \beta\sigma) * I(x, y) - g(x, y, \sigma) * I(x, y) \quad (2.5)$$

where the scale difference is chosen $\beta = \sqrt[3]{2}$. Local extrema of this function in position and scale are marked as interest points. We have shown in Fig. 2.3 the absolute value of this function for $\sigma = 5$. It can be seen that the function is triggered by extrema in image intensity, which match this particular choice of scale σ .



Figure 2.3: The absolute value of the difference of Gaussian (2.5) for $\sigma = 5$.

There are several other approaches which search for intensity extrema in the images, like [90] and [53]. In the latter, regions with extreme intensity values are sought that have pixel values all higher/lower than the surrounding boundary pixels. Using a thresholding operation, the intensity threshold where the smallest change of the region's size is observed, is selected for defining the final region.

A few comparative evaluations of interest point detectors have been performed so far. In [74] two evaluation criteria, repeatability and information content, were introduced and applied to a small set of detectors. Repeatability is the ratio of the number of detected common 3D-points in both images and the total number of

detected points. It measures how well the detector performs after a change of the imaging conditions. Information content is the entropy of local image descriptors at the positions found by the detector. The higher the entropy of the descriptor, the more interesting the positions are. It was shown that the Harris detector using Gaussian derivatives gave the best results for the two criteria.

In a more recent evaluation [59] a larger set of detectors was compared. The results showed that none of the detectors performed best in all experiments. Some detectors work better for specific types of scenes or objects. Multiple detectors may therefore be used to cope with a large variety of images.

2.2 Local descriptors

When the interest point locations are established, the surrounding regions can be used for comparison. To this end we need to extract a descriptor or feature from each region, which is unaffected by illumination or viewpoint changes. The local effect of changing illumination can be modeled by $cI(x, y) + b$, where c is a change in contrast and b in brightness. Subtracting the mean intensity and dividing by the standard deviation normalizes the intensity values, so that they can be used for comparison. Concerning the change in viewpoint, it was already noted that affine invariance is generally sufficient since the imaged objects are locally planar. Ideally, the descriptor should then be invariant to rotation, scaling and shear change.

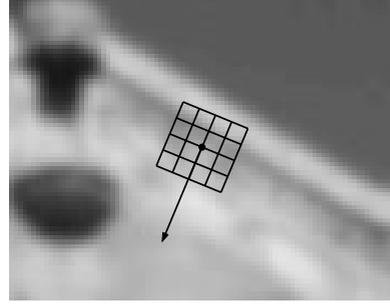
Many descriptors that have been introduced are based on gradient information in the region. The image retrieval method in [73] uses a set of rotationally invariant Gaussian derivatives [18]. The descriptor is computed at several scales to compensate for possible scale change.

A descriptor based on local gradient orientation histograms is used in [50, 51]. This SIFT (Scale Invariant Feature Transform) descriptor collects the gradients around the interest point in a 4×4 array of histograms. Each histogram quantizes the gradients from a 4×4 pixel region in 8 different directions. They are weighted by their gradient magnitude and the value of a Gaussian function at the center, which emphasizes gradient directions in close proximity of the interest point. The characteristic scale and dominant orientation of the region determine the size and orientation of the histogram array. The descriptor is in this way invariant to scaling and rotation. Due to the fact that a histogram is used, inaccuracies in the position of the descriptor can be compensated. An example of the histogram array for an interest point and the resulting descriptor are shown in Fig. 2.4. The arrow in Fig. 2.4(b) indicates the orientation of the interest region. The image is shown at the scale of the detected interest point, which is also used to compute the gradients. Fig. 2.4(c) shows the values of the resulting SIFT descriptor. It contains 8 direction bins for each histogram, which makes 128 histogram entries in total.

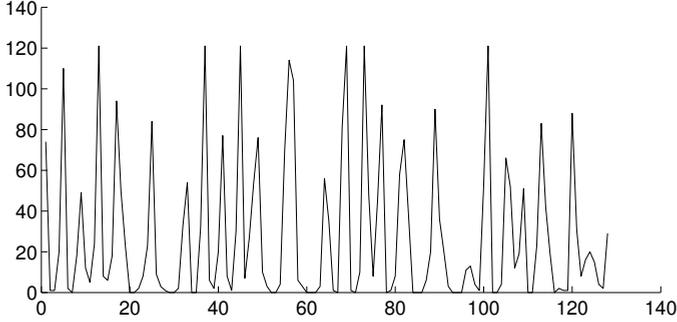
One way of obtaining shear invariant regions is using the second moment matrix in (2.1). Building on the work in [48], it is shown in [3] that if the local image



(a) Detail of the test image.



(b) Orientation and scale of the array of histograms for an interest point.



(c) Values of the SIFT descriptor.

Figure 2.4: The extraction of the SIFT descriptor.

coordinates belonging to a correspondence are first transformed by

$$(x_t \ y_t)^\top = C_a^{\frac{1}{2}}(x \ y)^\top \quad (x'_t \ y'_t)^\top = C'_a{}^{\frac{1}{2}}(x' \ y')^\top \quad (2.6)$$

then they are related by a 2×2 rotation matrix R through

$$(x'_t \ y'_t)^\top = R(x_t \ y_t)^\top \quad (2.7)$$

The matrices in (2.6) are the square roots of $C_a = (C_a^{\frac{1}{2}})^\top C_a^{\frac{1}{2}}$ and $C'_a = (C'_a{}^{\frac{1}{2}})^\top C'_a{}^{\frac{1}{2}}$, respectively. C_a and C'_a are affinely adapted versions of the second moment matrix C in (2.1). In particular, both the derivative operations and neighborhood averaging are performed with non-circular Gaussian functions. An algorithm is given in [3], which finds each of the matrices C_a and C'_a in an iterative process by computing C and transforming the neighborhood with $C^{\frac{1}{2}}$. This process continues until C has converged to the identity matrix. It is assumed here that no significant scale

differences are present, and that the descriptor should only be rotation invariant. Later an iterative algorithm was proposed that achieved full affine invariance [57].

Affine invariance of the region has also been achieved in [89], where regions are defined by means of edges near corner points. The edges define an affine invariant frame for the regions, and the size of the region is determined by some specific criterion. The descriptor is a set of color moment invariants. Cross-correlation is applied as additional check for the found matches, and is made possible by the affine normalization of the region. In [90] regions are initially established based on local intensity extrema. The boundary of the region is refined by extrema of a function of intensity, which is measured along rays emanating from the region's center. An ellipse is then fitted to the region, and its interior serves as the affine invariant region. The regions are described by color moment invariants and matches are additionally cross-correlated. The method in [64] also selects the initial regions based on intensity extrema. One of the affine normalization methods that is proposed in this work uses two line segments that bridge concave parts of the region's boundary. The segments form an affine frame which is used for cross-correlating the pixel color values.

In a comparative study [58] it was shown that the SIFT [50, 51] descriptor performed best among a variety of descriptors. An extension to SIFT was also given and shown to give a further improvement; it uses a radial array of histograms and a larger number of orientation bins. In addition it was shown that plain cross-correlation is very sensitive to inaccuracies in the interest point position and affine region parameters.

A somewhat different class of features than the discussed "point" features are straight line segments, whose invariance properties have been used for object recognition [31, 32, 34] and correspondence search [80, 81, 4]. Although correspondences between lines are found, the goal is to eventually establish a set of point correspondences. In [80] line segments are extracted between corner points in the images. The segments are assumed to lie on planar surfaces, which allow the transformation to be affine invariant. The intensity profiles along corresponding lines then only differ by a scale factor. To acquire a set of point matches, the line segments originating from a corner are used in a voting framework to find similar corner points. In [81] the method was further improved by taking the ordering of line segments around a corner point into account. In [4] an edge detector was used to define the line segments. Here points were established by intersections of line segments, and the method therefore works well on scenes having little texture.

2.3 Feature comparison and local geometric constraints

The descriptors at the interest points have to be compared across the images to establish a set of initial correspondences. When there are n local features in the

first image and m in the second, then nm feature comparisons will give us the set of best matching pairs. In addition, there exist several local constraints which we may apply to reject correspondences that are potentially outliers. In particular, when two features are spatially close in an image, they are likely to describe parts of an object that are close in 3D. In other images of the object these features should then be close as well. When we encounter two correspondences in a small neighborhood in the first image, they should as well be in a small neighborhood in the second image.

This reasoning is used in [73], where a correspondence is only considered correct when half of its closest features also form correspondences. Furthermore, it is required that the angles between the features are consistent. For example, for the correspondences in Fig. 2.5, we should have that

$$\angle(ab) - \angle(ac) \approx \angle(a'b') - \angle(a'c') \quad (2.8)$$

This only holds when there is no significant shear change between the images.



Figure 2.5: The correspondence $a \leftrightarrow a'$ is likely to be correct if there also exist correspondences $b \leftrightarrow b'$ and $c \leftrightarrow c'$.

In [95] the quality of a correspondence is judged by its feature similarity score and the number of correspondences in its neighborhood. The contribution of a neighboring correspondence depends on both its own similarity score and the difference in feature distances. In particular, the correspondence $b \leftrightarrow b'$ contributes to $a \leftrightarrow a'$ when $d(a, b) \approx d(a', b')$. The contribution also depends on the total distance, so that the influence of $c \leftrightarrow c'$ is smaller than $b \leftrightarrow b'$. In the cited work, multiple correspondences for each feature are possible; a potential correspondence is present when the feature similarity is above a certain threshold. The quality measure is used to select the correct correspondences from this set of available ones.

The method in [9] checks pairs of correspondences by their positions and descriptor properties. In addition to $d(a, b) \approx d(a', b')$ it requires that the difference in scale σ of the features is comparable

$$\sigma(a) - \sigma(b) \approx \sigma(a') - \sigma(b') \quad (2.9)$$

Also, the difference in orientation θ of a feature and the connecting line segment should be comparable

$$\theta(a) - \angle(ab) \approx \theta(a') - \angle(a'b') \quad (2.10)$$

These pairwise constraints are used to group neighboring correspondences and subsequently remove outliers.

After using local constraints the initial set of correspondences has undergone a first filtering stage. However, the drawback of local constraints is that also inliers can be filtered out, since the constraints are only local approximations of the image transformation. The actual transformation is determined by the epipolar geometry, which is discussed next.

2.4 The epipolar geometry

To see what the relation is between correspondences of an object, we will first describe the imaging process that transforms 3D world points into 2D image points. The relation between correspondences will then follow from combining two different imaging processes for the object.

2.4.1 Camera projection

We will consider the projection for an ideal pinhole camera, for which all light rays go through a central point (the lens). In Fig. 2.6 this type of camera is illustrated. The camera is positioned in the origin of the coordinate system, and the principal axis points in the direction of the positive Z axis. The lens or center of projection is located at the camera center \mathbf{C} , and the image plane is positioned at focal length f from the center. The light ray of a point \mathbf{X} passes through the camera center \mathbf{C} , and the intersection with the image plane forms the projection \mathbf{x} .

It is convenient to use homogeneous coordinates (see Appendix A) for the points. So, let the 3D point be denoted by $\mathbf{X} = (X, Y, Z, 1)^\top$ and the projection in the image plane by $\mathbf{x} = (x, y, 1)^\top$. The coordinate system of the image plane is usually mirrored, so that positive X and Y coordinates will also correspond to positive x and y coordinates in the projection. In this way, the image plane can also be thought of as being in front of the camera center at distance f , and having x and y axes pointing in the same direction as the X and Y axes (see for example Fig. 2.7).

The position of the point \mathbf{x} can be calculated by

$$x = \frac{fX}{Z} \quad y = \frac{fY}{Z} \quad (2.11)$$

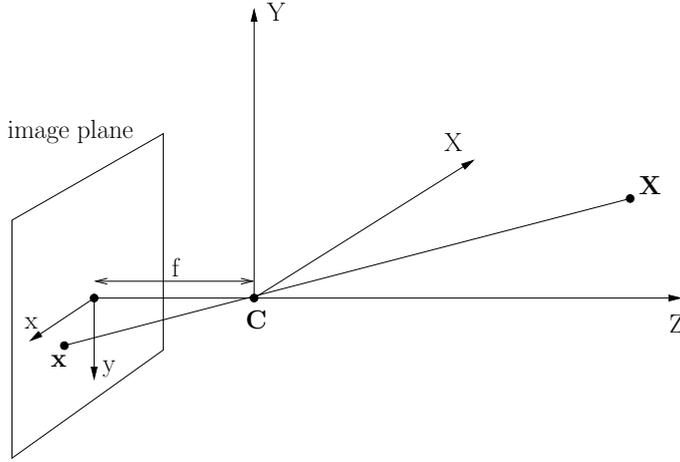


Figure 2.6: The projection \mathbf{x} of a world point \mathbf{X} made by a pinhole camera with focal length f and camera center in the origin.

which can be written in matrix form as

$$\mathbf{x} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.12)$$

since division of \mathbf{x} by Z yields the same point in homogeneous coordinates. In general, a camera may be rotated or translated with respect to its position in Fig. 2.6. To find the projection of a world point in this case, we can first transform the world point to the camera frame (i.e. specific world frame) in Fig. 2.6 and subsequently compute its projection. If the camera is rotated by a matrix R^\top and translated by a vector \mathbf{C} , then the inhomogeneous world point \mathbf{X} is given in the camera frame as $\mathbf{X}_c = R(\mathbf{X} - \mathbf{C})$. In homogeneous coordinates this becomes

$$\mathbf{X}_c = \begin{bmatrix} R & -RC \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{X} \quad (2.13)$$

If we now combine (2.12) and (2.13) we get

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}_c = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} [R \quad -RC] \mathbf{X} \\ &= KR[I \quad -\mathbf{C}] \mathbf{X} \\ &= P\mathbf{X} \end{aligned} \quad (2.14)$$

where K is the diagonal matrix containing the entries $(f, f, 1)$. This invertible matrix contains the internal camera parameters, and is given here in its simplest form which is sufficient for our discussion. The matrix $P = KR[I \ -\mathbf{C}]$ is the 3×4 camera projection matrix, which contains all internal and external (positional) camera parameters.

2.4.2 The fundamental matrix

Having established the relation between world points and their projections, we can now investigate the relation between different projections of a single world point. Let us consider two cameras with projection matrices P and P' , which are making projections of an unspecified world point \mathbf{X} (see Fig. 2.7).

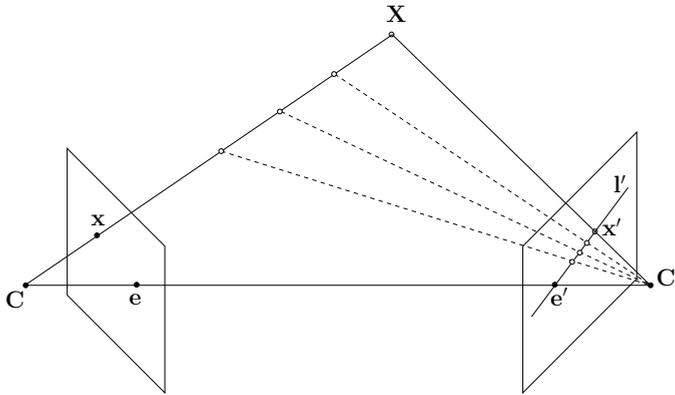


Figure 2.7: Two cameras with centers \mathbf{C} and \mathbf{C}' make projections of the same world point \mathbf{X} . The projection on the first image plane corresponds to a line of possible projections on the second image plane.

Given a projection $\mathbf{x} = P\mathbf{X}$ in the first camera, the projection $\mathbf{x}' = P'\mathbf{X}$ in the second camera cannot be arbitrary. The projection \mathbf{x} constrains the point \mathbf{X} to be somewhere on the line connecting \mathbf{C} and \mathbf{x} . The projection of this line in the second image plane is the epipolar line l' , which contains all possible projections of \mathbf{x}' . The line l' necessarily crosses the epipole $\mathbf{e}' = P'\mathbf{C}$, which is the projection of the first camera's center. Since the line contains \mathbf{x}' we must have that $\mathbf{x}'^\top l' = 0$. The precise expression for l' will depend on \mathbf{x} , and follows from the fundamental matrix relation [15, 24].

Theorem 1. *There is a 3×3 rank 2 matrix F , called the fundamental matrix, which relates projections of a world point by*

$$\mathbf{x}'^\top F \mathbf{x} = 0 \quad (2.15)$$

Proof. We follow the derivation in [94]. Without loss of generality, the first camera is assumed to be placed as in Fig. 2.6 so that $P = K [I \ \mathbf{0}]$. The second camera is rotated by R^\top and translated over \mathbf{C}' resulting in the camera matrix $P' = K'R [I \ -\mathbf{C}']$. Multiplying both sides of $\mathbf{x} = P\mathbf{X}$ with K^{-1} gives

$$K^{-1}\mathbf{x} = [I \ \mathbf{0}]\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (2.16)$$

and multiplying both sides of $\mathbf{x}' = P'\mathbf{X}$ with $R^\top K'^{-1}$ results in

$$R^\top K'^{-1}\mathbf{x}' = [I \ -\mathbf{C}']\mathbf{X} = \begin{pmatrix} X - C'_1 \\ Y - C'_2 \\ Z - C'_3 \end{pmatrix} \quad (2.17)$$

Now we construct the skew-symmetric matrix

$$[\mathbf{C}']_\times = \begin{bmatrix} 0 & -C'_3 & C'_2 \\ C'_3 & 0 & -C'_1 \\ -C'_2 & C'_1 & 0 \end{bmatrix} \quad (2.18)$$

which has the property that $(C'_1, C'_2, C'_3)[\mathbf{C}']_\times = \mathbf{0}^\top$. When we combine this matrix with (2.16) and (2.17) we get

$$\begin{aligned} \mathbf{x}'^\top K'^{-\top} R [\mathbf{C}']_\times K^{-1} \mathbf{x} &= (R^\top K'^{-1} \mathbf{x}')^\top [\mathbf{C}']_\times K^{-1} \mathbf{x} \\ &= (X - C'_1, Y - C'_2, Z - C'_3) [\mathbf{C}']_\times \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \\ &= 0 \end{aligned} \quad (2.19)$$

where the last equality follows from $(C'_1, C'_2, C'_3)[\mathbf{C}']_\times = \mathbf{0}^\top$ and the fact that the quadratic form for any skew-symmetric matrix is zero.

The matrix $F = K'^{-\top} R [\mathbf{C}']_\times K^{-1}$ in the first line of (2.19) is the fundamental matrix. It follows that $\text{rank}(F) \leq \text{rank}([\mathbf{C}']_\times) = 2$. □

The epipolar line in the second image can now be written as $\mathbf{l}' = F\mathbf{x}$, and the epipolar line in the first image as $\mathbf{l} = F^\top \mathbf{x}'$. The fundamental matrix is completely specified by camera matrices P and P' . If the camera parameters are unknown,

which is usually the case, it is also possible to compute the fundamental matrix solely from point projections. When the elements of F are written as

$$F = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \quad (2.20)$$

the relation in (2.15) for a corresponding pair $\mathbf{x} = (x, y, 1) \leftrightarrow \mathbf{x}' = (x', y', 1)$ can be rewritten as

$$x'x f_1 + x'y f_2 + x' f_3 + y'x f_4 + y'y f_5 + y' f_6 + x f_7 + y f_8 + f_9 = 0 \quad (2.21)$$

Since multiplying both sides of (2.15) with a scalar value does not change the relation, the scale of the elements of F can be arbitrary. It is therefore possible to calculate F from eight correspondences [49, 26]. However, because $\det(F) = 0$ the matrix F has only seven degrees of freedom, and thus 7 different correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$ are sufficient to find F (see e.g. [25]). Solving the system of equations

$$\begin{bmatrix} x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots \\ x'_7 x_7 & x'_7 y_7 & x'_7 & y'_7 x_7 & y'_7 y_7 & y'_7 & x_7 & y_7 & 1 \end{bmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_9 \end{pmatrix} = \mathbf{0} \quad (2.22)$$

gives a two-dimensional nullspace for F . Let this nullspace be spanned by matrices F_1 and F_2 , so that $F = \alpha F_1 + (1 - \alpha) F_2$. We can then use the singularity constraint of the fundamental matrix to find the solution. This constraint $\det(\alpha F_1 + (1 - \alpha) F_2) = 0$ gives a cubic polynomial in α . Solving the polynomial for α yields (at maximum) three real solutions for F . The solution which satisfies (2.15) for most correspondences is the final F .

2.4.3 Robust fundamental matrix estimation

The estimation of the fundamental matrix, or epipolar geometry, from a set of correspondences is relatively easy when all of them are inliers. Their positions may then be affected by noise, but they still conform with the ground truth epipolar geometry. In practice, however, some correspondences will be outliers and their image positions may be arbitrarily far from their correct positions. Outliers will result from non-corresponding features which accidentally give good matches. To illustrate the estimation process of the fundamental matrix under the presence of outliers, we will use the case of 2D linear regression. In the data set from Fig. 2.8 we see that all points except one obey a linear model. The single deviating point is clearly an outlier.

We can fit a line through this data by orthogonal regression, that is, minimizing the sum of orthogonal distances from the points to the line. In contrast to least squares on the y coordinates, this type of regression is required when both x and y

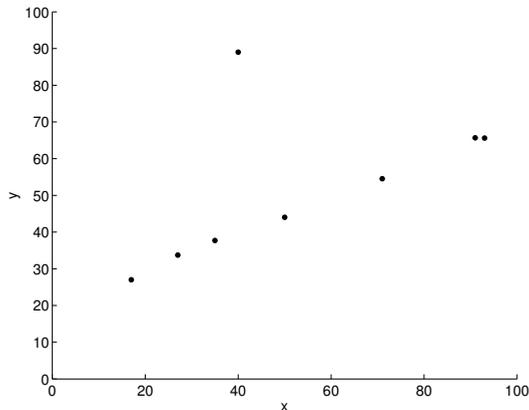


Figure 2.8: A data set with a single outlier.

are noisy. If the normal vector of the line is denoted by \mathbf{n} and its orthogonal distance from the origin as b , the minimization of the sum of squared orthogonal distances can be written as

$$\min_{\mathbf{n}, b} \sum_{\mathbf{x}} \frac{((x, y)\mathbf{n} + b)^2}{\|\mathbf{n}\|^2} \quad (2.23)$$

This can be solved efficiently and results in the line indicated in Fig. 2.9(a). Although the data contains mostly inliers, the single outlier causes the line to deviate substantially from the true or best model. It is clear that when all data is used, the outliers cause the squared error criterion to be inappropriate.

Both in statistics and computer vision a great deal of work has been done on improving the robustness of model estimation methods. Robust estimators are intended to be less susceptible to the influence of outliers. In the following we discuss several such robust estimators.

M-estimators

In the statistics literature it has been proposed to replace the squared error criterion with a different error function. Given the error value e , we may use instead of the squared error $\rho(e) = e^2$ another suitable function $\rho(e)$. For example, in the class of so called M-estimators, Huber's function [30]

$$\rho(e) = \begin{cases} \frac{1}{2}e^2 & \text{if } |e| \leq \gamma \\ \gamma|e| - \frac{1}{2}\gamma^2 & \text{if } |e| > \gamma \end{cases} \quad (2.24)$$

is quadratic for small and linear for large e . As such, it decreases the influence of large outlying values on the minimization. The constant γ is a control parameter

and determines the extent to which the function is quadratic. To choose its value correctly we must know the scale of the data, i.e. have knowledge of the degree of variation caused by noise. The scale either has to be known beforehand or can possibly be estimated from the data. An iterative procedure is then applied to find the best estimate of the model using $\rho(e)$.

The result of (2.24) for the line estimation example using automatic scale selection is shown in Fig. 2.9(b). The outlier does not corrupt the final model anymore. Though the M-estimator performs satisfactory for this linear regression data, the estimation of the epipolar geometry is not handled well by these estimators [87, 94]. In particular, the iterative estimation procedure needs a good initial estimate of the model. For poor initialization and moderate outlier ratios, the procedure may not converge to the true model.

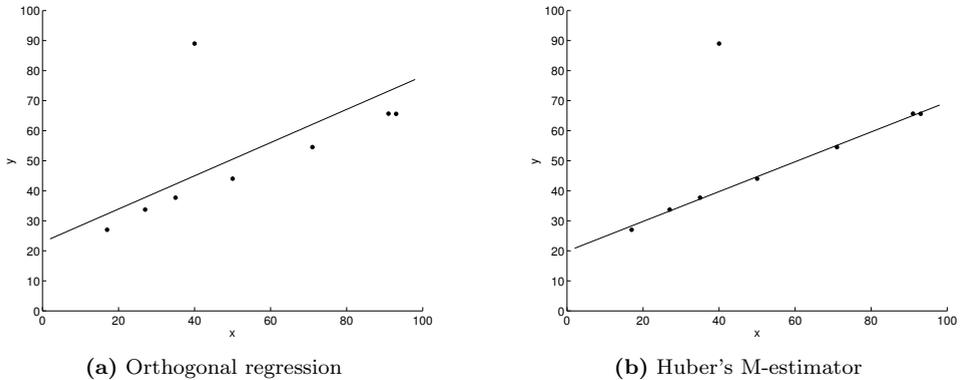


Figure 2.9: Orthogonal regression can not find the true linear relation in the data. The robust M-estimator using Huber's function is able to find the model precisely.

Hough transform

A robust estimation approach which has been heavily used in computer vision is the Hough transform [39, 14]. It has been applied for the detection of lines in images and has later been generalized to detect other shapes as well [2]. The Hough transform finds the model by a voting process in a parameter space, where each combination of parameters corresponds to a different model. The range of possible models is limited by quantizing the parameters; a particular bin in the parameter space will actually correspond to a small range of almost similar models. The parameter space is accumulated by letting the data points vote for each bin whose corresponding model they are supporting. The bin in the parameter space which has received

most votes, will produce the final set of model parameters. Since the quantized parameters lack some accuracy, the model may not precisely pass through the inliers. To improve its accuracy, several extensions to the standard Hough transform have been proposed. For example, in [61] the voting pattern in the parameter space is smoothed before searching for the maximum.

A notable drawback of the Hough transform is the large parameter space when using more complex models like the fundamental matrix. Applying the Hough transform for its estimation requires a 7-dimensional voting array (due to the 7 degrees of freedom), which becomes unmanageable even for a moderate number of quantization levels.

A different approach in the spirit of the Hough transform is tensor voting. This method is less complex and can be used for fundamental matrix estimation [79, 83]. The idea is that data points will vote for structure at neighboring data points. In Fig. 2.10 this is illustrated for the estimation of a line.

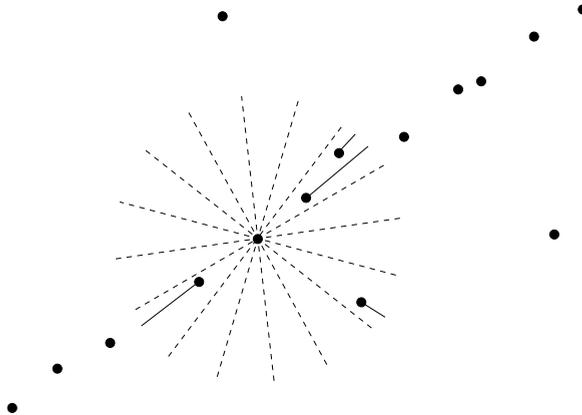


Figure 2.10: Illustration of the tensor voting process for estimating a line. Points in the neighborhood of the voting point will get a vote for a direction (solid line) with a certain weight (length of the line). The weight depends on the distance between the points.

All points close to the voting point receive a direction along with a weight. The farther the points are separated, the smaller the weight will be. After each point has voted for its neighbors, the accumulated directions in each point are used to find any dominant direction. If there is a dominant direction, the point should be an inlier for a line. In case there is no dominant direction, the point is an outlier. Afterwards, the inliers are used to find the model.

The size of the voting neighborhood is an important parameter of the method.

To select it correctly, the size of the model should be known. In case of fundamental matrix estimation, it has therefore been argued [83] that the method performs best when the movement between two images is small.

RANSAC

In [16] the RANSAC (RANdom SAmple Consensus) estimator was proposed. It has been applied for estimation problems in e.g. object recognition [45, 70], range image registration [10] and mosaicing [55, 29]. The method is based on the random selection of a small number of points (a sample) and the instantiation of the model corresponding to these points; the sample contains only the minimally required number of points to fully determine an instance of the model. The remaining data points are then checked to see if they support this model, i.e. are close enough in terms of some distance measure. Since there are outliers in the data, several samples have to be tried to assure with high probability that a sample containing only inliers is selected.

The number of iterations J needed to find such a sample can be determined from the required probability of success, i.e. the probability that at least one all-inlier sample is found in J iterations [16, 71]. Let ε denote the outlier ratio in the data, and d the number of points needed to hypothesize a model. If p is the probability of success, e.g. 0.99, then we have the relation

$$p = 1 - (1 - (1 - \varepsilon)^d)^J \quad (2.25)$$

For the estimation of a line we have $d = 2$, and for the fundamental matrix $d = 7$. We can see in Table 2.1 that the value of J increases rapidly for larger d and ε .

Table 2.1: The number J of RANSAC iterations according to (2.25) for success probability $p = 0.99$ and different values of the sample size d and outlier ratio ε .

$d \backslash \varepsilon$	0.1	0.3	0.5	0.7	0.9
2	3	7	17	49	459
4	5	17	72	567	46,050
7	8	54	588	21,055	46,051,700

In case the value of ε is known, the number of iterations to be executed would be known beforehand. Usually, however, the outlier ratio is unknown and we must use an alternative way for determining the number of iterations. An efficient manner is to begin RANSAC with a conservative estimate of ε , for example 0.99, and adjust it during following iterations [28]. When a certain hypothesized model has the largest number of support points seen so far, we know that there are at least that many

inliers in the data. This imposes a lower bound on $1 - \varepsilon$ for the data set, and we can adjust the remaining number of iterations J according to (2.25). The RANSAC algorithm using this strategy is given in Fig. 2.11. Here S_j is used to indicate the set of support points in iteration j , which are the points within a small distance T from the model. The largest support set found during the algorithm is denoted by S_{max} . The re-estimation step at the end of the algorithm is performed to improve the accuracy of the final model. Since S_{max} will contain only points on a model, an ordinary least squares estimate will be sufficient.

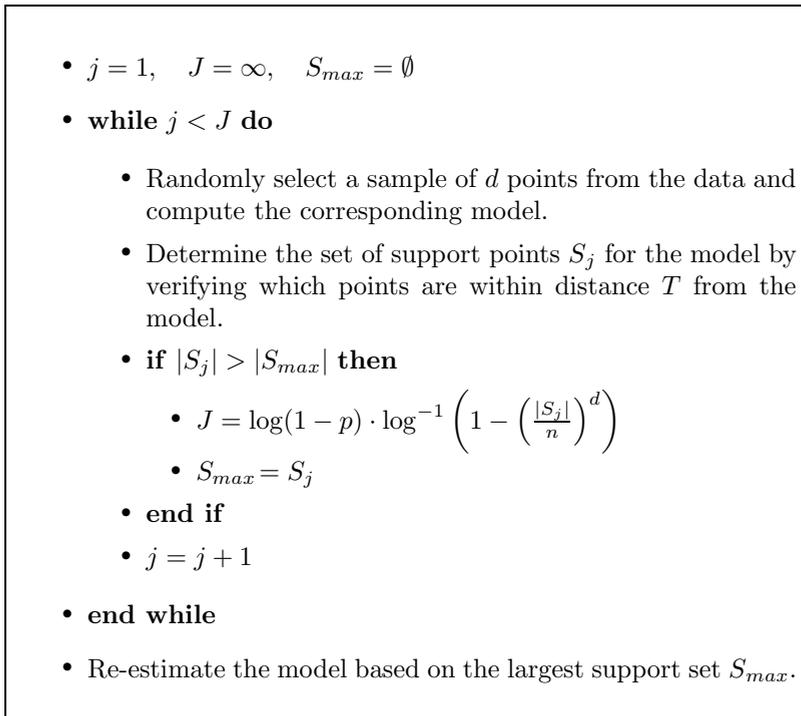


Figure 2.11: The RANSAC algorithm using an adaptive number of iterations.

The RANSAC algorithm provides a suitable way of fundamental matrix estimation [86] and has been used frequently for this purpose, see e.g. [5, 69, 67, 72]. The algorithm has essentially become the standard for epipolar geometry estimation [87, 94, 28], and in the remainder of the thesis we will consider only this particular estimator.

The samples drawn in RANSAC should contain 7 correspondences $\mathbf{x} \leftrightarrow \mathbf{x}'$ for

computing the fundamental matrix. We can then use a threshold on a suitably chosen error criterion to evaluate the support. If the noise on the coordinates of the inliers is assumed to be Gaussian, the Maximum Likelihood estimate of the fundamental matrix follows by minimizing the sum of reprojection errors [28], given by

$$\sum_{\mathbf{x} \leftrightarrow \mathbf{x}'} \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + \|\mathbf{x}' - \hat{\mathbf{x}}'\|^2 \quad (2.26)$$

where the inhomogeneous coordinate representation is used. The quantities $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ are estimated positions of \mathbf{x} and \mathbf{x}' , respectively. These estimates are reprojections of an estimated world point position $\hat{\mathbf{X}}$, and they satisfy the relation $\hat{\mathbf{x}}'^T F \hat{\mathbf{x}} = 0$. A disadvantage of the error measure in (2.26) is that it is computationally complex. Alternatively, an approximation to this error measure can be used: the Sampson error. If we let $\mathcal{A} = \mathbf{x}'^T F \mathbf{x}$ denote the algebraic distance, the Sampson error can be written as

$$\frac{\mathcal{A}^2}{\left(\frac{\partial \mathcal{A}}{\partial x}\right)^2 + \left(\frac{\partial \mathcal{A}}{\partial y}\right)^2 + \left(\frac{\partial \mathcal{A}}{\partial x'}\right)^2 + \left(\frac{\partial \mathcal{A}}{\partial y'}\right)^2} \quad (2.27)$$

where the partial derivatives are evaluated at the positions (x, y) and (x', y') of the correspondence. This error measure is only dependent on the value of F and can therefore be computed much faster.

The threshold T for the error is based on the noise variance of the inliers. The noise variance can be estimated from the data using the median error [71], provided that the outlier ratio is lower than 0.5. Unfortunately, this will not always hold and in general means that we have to set the threshold heuristically.

2.5 Complexity of epipolar geometry estimation

The total complexity of epipolar geometry estimation will be the combination of interest point selection, feature extraction, feature comparison and robust estimation. It is the question which stage can be optimized best for achieving the most complexity reduction.

Apart from the complexity of the interest point algorithm itself, the selection of interest points in the first stage has a profound impact on the complexity of following stages. The number of interest points selected largely determines the amount of data which is to be processed. Usually their number is selected by a threshold on some interest function, which allows the selection of a varying number of interest points. In the second stage, the features are extracted at the interest points, where the chosen feature type determines the complexity of this stage and the following feature comparison stage, see also [41]. The size of the representation and the required similarity measure are important in this respect. The feature comparison in the third stage is quadratic in the number of features, so a large

number of features can slow down this stage significantly. In the fourth stage we use the RANSAC algorithm on the set of initial correspondences from the previous stage to find the epipolar geometry. As was shown in Table 2.1, the number of iterations used by the algorithm is a major cause for its complexity, and it is the outlier ratio that determines this number of iterations.

Reducing the number of features for controlling the complexity in the first three stages, will only have a limited effect in the fourth stage. It is true that fewer correspondences make individual iterations faster, but for an equal outlier ratio the number of iterations remains unchanged. The outlier ratio can not be influenced a priori. Even a very good interest point detector and feature extractor can not guarantee an upper bound on the outlier ratio. In particular, large viewpoint differences between images will inevitably lead to many outlying correspondences. Furthermore, a reduction in the complexity of the first three stages may potentially lead to fewer inliers among the correspondences, and therefore a more complex fourth stage.

The abovementioned arguments motivate our choice of optimizing the fourth robust estimation stage. Our goal will therefore be to improve the RANSAC algorithm such that it arrives at the solution in less iterations. As is clear from (2.25), a reduction in the number of iterations J is achieved by reducing either the outlier ratio ε or sample size d . Although it seems that these quantities are fixed, they can be altered by making some specific adjustments to the RANSAC algorithm. Chapters 3 and 4 will focus on different ways of reducing the outlier ratio. Chapter 5 will concentrate on the use of a smaller sample size.

2.6 Discussion

We have described the extraction of local features at image positions which have some characteristic image structure, so called interest points. Such points can be detected on the same object part even when there has been a significant change in imaging conditions. At the interest points, descriptors of the local image region are computed. Assuming the object is locally planar, the descriptors only need to be affine invariant to yield the same value across different views. However, descriptors invariant to rotation and scaling may still show good performance, like the SIFT descriptor. The descriptors are subsequently compared across two images to establish a set of correspondences. To remove some of the outlying correspondences, local geometric constraints may be applied. They are based on the assumed spatial consistency of features from neighboring correspondences.

The epipolar geometry is the combination of two imaging processes, and governs the position of inlying correspondences in the images. It can be estimated from the correspondences by robust methods, which reduce the influence of outlying correspondences. The RANSAC algorithm is well suited for epipolar geometry estimation, and, since a complexity reduction can best be achieved in the robust estimation stage, it will serve as the basis for our discussion.

In the following chapters we will present several methods, which aim at achieving faster RANSAC execution by lowering the outlier ratio or the model dimensionality.

Chapter 3

A Priori Inlier Probabilities for Correspondences based on Quadrics

In this chapter, we address methods for lowering the outlier ratio of a set of correspondences before RANSAC execution. These methods work by incorporating prior knowledge of the correspondences into the sampling process. We will propose a new algorithm that is able to achieve this goal under a minimal computational complexity. This chapter is based on [35].

3.1 Introduction

The outlier ratio equals the probability of randomly sampling an outlier from the data, since all data points have an equal probability of being sampled. When we would have a reason to assume that a particular data point is an inlier, it is beneficial to increase the sampling probability of this point. The probability of sampling an outlier will then become lower than the outlier ratio. Any confidence regarding points being inliers can be reflected accordingly in the sampling probabilities. Data points with high inlier confidence will then be sampled more often.

The measure of confidence used should be reliable and easy to compute. After using the measure, the resulting probability of sampling an outlier, or the 'effective' outlier ratio, should be lower than the original outlier ratio. If not, the RANSAC algorithm will become even more complex, especially for high outlier ratios. Also, the complexity of computing the measure should be small in comparison to the RANSAC complexity, since an overall complexity reduction is desired.

In [84] a method in the context of motion estimation is proposed, which uses

the matching score of a correspondence to alter the probability of selecting that correspondence. This method is generally applicable since the comparison of local descriptors will usually yield some confidence value about the similarity of a pair of descriptors. The distinction can be further improved if spatial information of the correspondences is taken into account. A method based on using this information is the ROR (Rejection of Outliers by Rotations) algorithm [1]. Here it is shown that there exists a 3D rotation of the points in one image, which makes the directions of all good correspondences in the joint image plane equal. The directions that do not conform to the most prominent direction, which is found by random trials, originate from outlying correspondences. Eliminating these correspondences will decrease the effective outlier ratio. The method shows very good results, but a drawback is that it makes a few assumptions about the data, like a dominant depth value of the world points and the absence of significant camera rotation about the principal axis. It also assumes the camera's focal length to be known. While the assumptions about the scene may be justified for many image pairs, it will rarely be known beforehand. In addition, information about the cameras is usually not available.

In this chapter, we propose a method to decrease the outlier ratio by using spatial correspondence information, without making any assumptions about the cameras or imaged points. The method is based on the comparison of the point correspondences to a series of quadrics. Outlying correspondences tend to have an arbitrary position with respect to a quadric. When we gather statistics about the correspondences' positions, we can use this fact to decrease the effective outlier ratio. The complexity of the proposed algorithm is low, since for each correspondence only a small number of algebraic distances is computed.

In contrast to the ROR algorithm, the image points are not transformed in any way, but rather compared to independent quadric models positioned in the space. Furthermore, there is no search for a specific model which provides the distinction between inliers and outliers. Though some quadrics will provide more information than others in this respect, it is the complete set of quadrics that produces the result.

In Section 3.2 the proposed algorithm for outlier identification is explained. A short discussion of the complexity of the proposed algorithm and ROR is given in Section 3.3. The evaluation of both algorithms for synthetic and real data is given in Section 3.4. Finally, in Section 3.5, we give some concluding remarks.

3.2 Quadric comparisons

Consider a set of points \mathbf{X}_i for $i = 1, \dots, n$ in \mathbb{P}^3 , which are indicated by homogeneous coordinates $\mathbf{X} = (X, Y, Z, 1)^\top$ (see Appendix A). Each point \mathbf{X}_i results in a correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. The fundamental matrix relation essentially defines a quadric relation on the points \mathbf{X} , which can be seen by writing

$$\mathbf{x}'^\top F \mathbf{x} = \mathbf{X}^\top P'^\top F P \mathbf{X} = 0 \quad (3.1)$$

where the quadric is given by $P'^\top FP$. This quadric is special in the sense that the locus, i.e. the points \mathbf{X} obeying (3.1), consists of all world points \mathbf{X} .

Replacing F with an arbitrary 3×3 matrix Q will yield a quadric $P'^\top QP$, for which the locus in general consists of a two-dimensional variety in \mathbb{P}^3 . The quadric will split the points \mathbf{X} into two sets: those with positive and those with negative algebraic distance to the quadric, that is

$$\mathbf{X}_i^\top P'^\top QP \mathbf{X}_i \gtrless 0 \quad i = 1, \dots, n \quad (3.2)$$

If the quadric is shaped in such a way that the space with positive (or negative) algebraic distance is very small, few points \mathbf{X} will yield $\mathbf{X}^\top P'^\top QP \mathbf{X} > 0$ (or $\mathbf{X}^\top P'^\top QP \mathbf{X} < 0$). This is equivalent to having few image points with $\mathbf{x}'^\top Q \mathbf{x} > 0$ (or $\mathbf{x}'^\top Q \mathbf{x} < 0$). In the case of outliers, however, there are not necessarily few points like that. Since the outliers do not correspond to real world points, their algebraic distances do not follow the same subdivision as for the inliers. Ideally, the outliers have an equal probability of yielding a positive or negative algebraic distance. The idea is that by examining statistics of the signs for several different quadrics, we are able to distinguish between inliers and outliers.

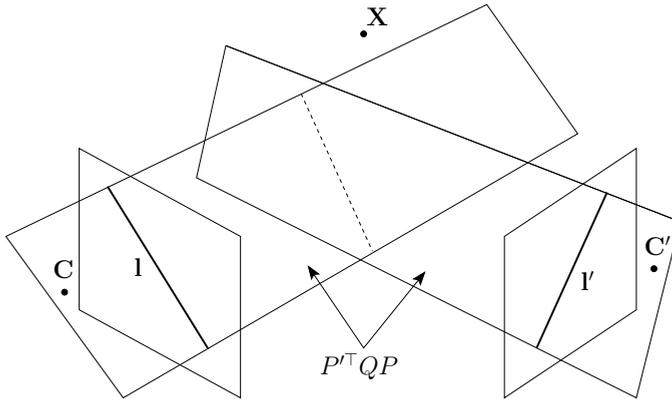


Figure 3.1: The quadric $P'^\top QP$ formed by selection of lines l and l' in the image planes. Both camera centers C and C' lie on a plane of the quadric, which subdivides the points \mathbf{X} in the space.

The type of quadric formed by Q depends on the rank of the symmetric part of $P'^\top QP$, which is given by

$$\frac{P'^\top QP + (P'^\top QP)^\top}{2} \quad (3.3)$$

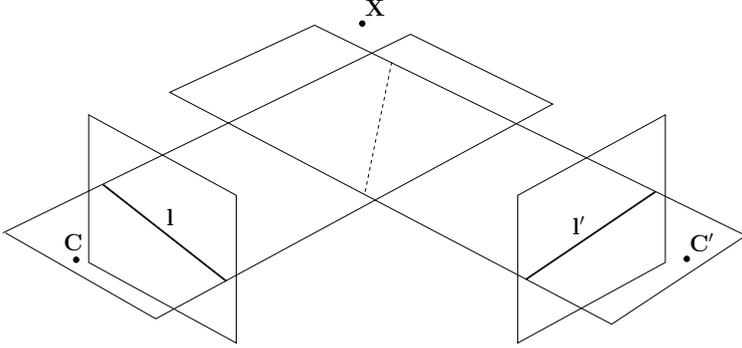


Figure 3.2: A quadric where nearly all world points \mathbf{X} will yield the same sign for the relation $\mathbf{X}^\top P'^\top Q P \mathbf{X}$.

The antisymmetric part, which equals (3.3) for a minus instead of a plus sign, always yields zero in the quadric relation and therefore does not contribute to $\mathbf{X}^\top P'^\top Q P \mathbf{X}$. If we choose Q to have rank 1, then both $P'^\top Q P$ and $(P'^\top Q P)^\top$ from (3.3) have rank 1. As a result, the sum of these matrices in the symmetric part will generally have rank 2. In this case the locus corresponds to two planes, see Fig. 3.1. The camera centers \mathbf{C} and \mathbf{C}' each lie on one of the planes, so that this plane is visible as a line in the image. The quadric of rank 2 can therefore be defined by choosing appropriate lines \mathbf{l} and \mathbf{l}' in the image planes. An example of a highly imbalanced quadric for this configuration is shown in Fig. 3.2. It is possible to choose Q with higher rank, but the placement of lines \mathbf{l} and \mathbf{l}' is the easiest way of letting the quadric pass through the means of the projected points. This assures that points will lie on all sides of the quadric, so that there is enough variation in the sign of the algebraic distances in the point set. We select the lines \mathbf{l} and \mathbf{l}' such that they pass through the means $\bar{\mathbf{x}} = (\bar{x}, \bar{y}, 1)^\top$ and $\bar{\mathbf{x}}' = (\bar{x}', \bar{y}', 1)^\top$ in the first and second image, respectively. If the angles of the lines with the image x-axis are denoted by θ and θ' , the lines are given by

$$\mathbf{l} = (-\sin(\theta), \cos(\theta), \bar{x} \sin(\theta) - \bar{y} \cos(\theta))^\top \quad (3.4)$$

in the first image and

$$\mathbf{l}' = (-\sin(\theta'), \cos(\theta'), \bar{x}' \sin(\theta') - \bar{y}' \cos(\theta'))^\top \quad (3.5)$$

in the second. Then the matrix Q is constructed by

$$Q = \mathbf{l}' \mathbf{l}^\top \quad (3.6)$$

which is a rank 1 matrix.

We do not know the precise position of the planes of the quadric $P'^T Q P$, since the camera matrices P and P' are unknown. Yet, we can pick an arbitrary θ and θ' with resulting Q , and subsequently examine the sign of the algebraic distances of all correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. If the sign of the outliers is more or less random, the inliers should determine the dominant sign. Therefore, a point belonging to the set with the dominant sign is likely to be an inlier. Counting these occurrences for several different Q 's will gather statistics about the probability of dealing with either an inlier or an outlier. The counts will be put in variables c_i for $i = 1, \dots, n$, which are being updated after each quadric used. We select L equally spaced angles from $[0, \pi]$ for both θ and θ' , and for every pair of θ and θ' a quadric is generated. It is also possible to select θ and θ' randomly for each Q , which gives almost comparable results [36]. However, due to the randomness we should run the algorithm several times for averaging and this is more costly. The resulting algorithm for equally spaced angles is given in Fig. 3.3.

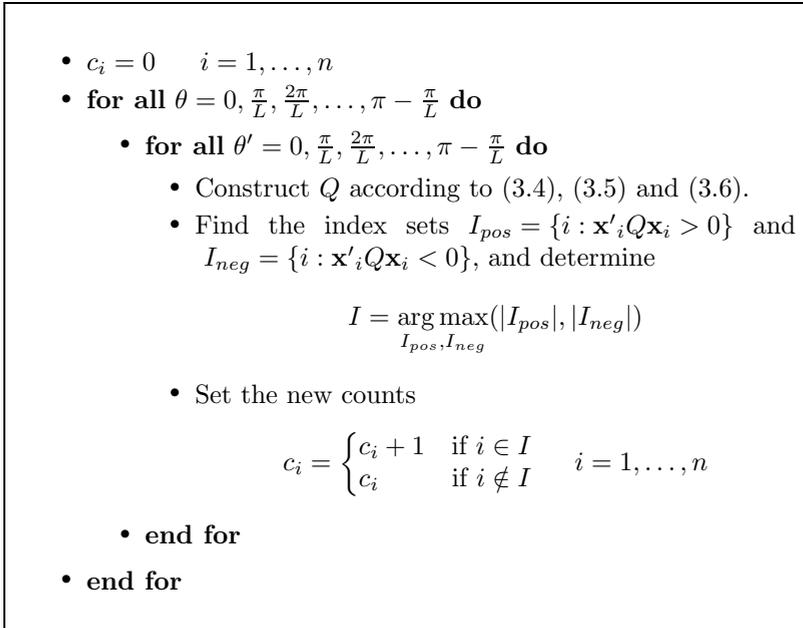


Figure 3.3: The quadric algorithm.

The resulting values of the counts can be incorporated into the sampling probabilities of the correspondences. In particular, when the probability of selecting a correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ is the relative number of counts $c_i / \sum_i c_i$, the effective

outlier ratio ε_{quad} after applying the quadric algorithm has become

$$\varepsilon_{quad} = \frac{\sum_{i: \mathbf{x}_i \leftrightarrow \mathbf{x}'_i} c_i}{\sum_i c_i} \quad (3.7)$$

This measure is used to evaluate the algorithm in the experiments, where the ground truth outlier data is available.

3.3 Complexity

The computational savings for a single iteration of RANSAC are the computation of F and one or three times n distance computations. The quadric algorithm itself should have a low additional complexity to retain most of these savings. We have chosen for $L = 8$ in the experiments, which means that for the quadric algorithm we need only $64n$ algebraic distance computations. This is a minor computational load that justifies the use of the algorithm prior to RANSAC in almost every situation.

The standard implementation of the ROR algorithm, however, uses 1,000 random rotations to reject the outliers [1]. The authors propose to take the majority vote over 10 runs, so that the algorithm eventually requires $10,000n$ computations of segment angles and 10,000 computations of the mode of an angle distribution. This makes ROR much more costly than the quadric algorithm.

To illustrate this complexity difference, we have recorded the running times for the “Valbonne church” data set from Table 3.3 on a Pentium IV 2.8 GHz computer. The ROR algorithm needed approximately 2.0 seconds to complete, while the quadric algorithm finished in 0.74 milliseconds.

3.4 Experimental results

We will evaluate the proposed quadric algorithm by comparing it to RANSAC without any preprocessing and the ROR algorithm [1]. In ROR possible outliers are completely rejected, so that the effective outlier ratio ε_{ROR} is the ratio of the number of retained outliers and the total number of retained points. We have used the standard implementation¹ of the algorithm without adjusting any parameters. In the quadric algorithm we have used $L = 8$ angles for θ and θ' .

3.4.1 Synthetic data

We have generated synthetic data by randomly positioning points in a cube in \mathbb{R}^3 . The cameras with equal internal parameters are randomly positioned on a sphere

¹The code is available at http://www.cs.technion.ac.il/Labs/IsI/Project/Projects_done/ror/.

around the cube. The radius of the sphere is twice the edge length of the cube. The points are between 20 and 60 focal lengths away from the cameras. The image coordinates of the inliers are perturbed by Gaussian distributed noise with a standard deviation of 0.3% of the image size. For the outliers we randomly select two different space points, and use their non-corresponding projections as a data pair. The experiment is run 100 times for a particular outlier ratio ε , and each run will have different point and camera positions. For the ROR algorithm we have scaled the coordinate values to resemble realistic pixel values.

The results for 50 and 200 points are shown in Table 3.1 and 3.2, respectively. In the tables we have shown the average ε_{quad} and ε_{ROR} and their standard deviations over the 100 runs. In addition, we have calculated according to (2.25) for each run the *theoretical* number of iterations J , J_{quad} and J_{ROR} corresponding to ε , ε_{quad} and ε_{ROR} , respectively. The quantities J_{quad} and J_{ROR} were then averaged over all runs and these values are shown in the tables. In Fig. 3.4 also the ratios of J and the average J_{quad} and J_{ROR} are depicted.

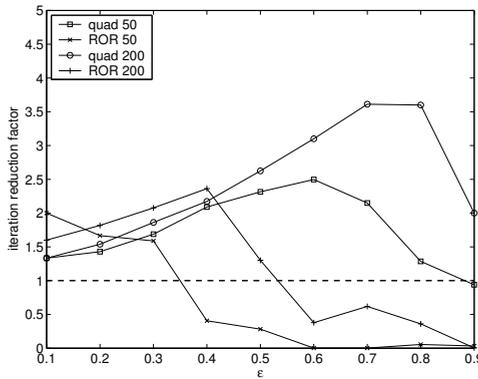


Figure 3.4: The ratios $\frac{J}{\bar{J}_{quad}}$ and $\frac{J}{\bar{J}_{ROR}}$ for J , \bar{J}_{quad} and \bar{J}_{ROR} from Table 3.1 and 3.2.

In general, the results for 200 points are better than for 50 points both in terms of the averages as well as the standard deviations. More points allow a better estimate of the side of the quadric that is largest. We can see that our method reduces the outlier ratio in both cases. An exception is the somewhat extreme case $\varepsilon = 0.9$ for 50 points, where only 5 inliers are present.

The synthetic data essentially violates the assumption of a dominant depth value in the ROR algorithm. Yet, in most runs the algorithm performs well, as can be seen by the lower average outlier ratio that it achieves when compared to our method. However, the standard deviation of ROR is larger, so there are several runs where the

Table 3.1: The results for synthetic data consisting of 50 points. The quantities indicated are the outlier ratio ϵ , the average outlier ratios $\bar{\epsilon}_{quad}$ and ϵ_{ROR} and their standard deviations, the number of iterations J , and the average number of iterations J_{quad} and J_{ROR} .

ϵ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\bar{\epsilon}_{quad}$	0.078	0.158	0.245	0.333	0.433	0.540	0.656	0.773	0.892
$\bar{\epsilon}_{ROR}$	0.046	0.101	0.171	0.199	0.309	0.405	0.525	0.684	0.867
STD ϵ_{quad}	0.013	0.016	0.022	0.021	0.021	0.022	0.025	0.025	0.015
STD ϵ_{ROR}	0.055	0.093	0.122	0.155	0.175	0.188	0.181	0.140	0.075
J	8	20	54	163	588	2,809	$2.1 \cdot 10^4$	$3.6 \cdot 10^5$	$4.6 \cdot 10^7$
J_{quad}	6	14	32	78	254	1,125	9,769	$2.8 \cdot 10^5$	$4.9 \cdot 10^7$
J_{ROR}	4	12	34	401	2,084	$1.7 \cdot 10^6$	$1.3 \cdot 10^7$	$6.6 \cdot 10^6$	$1.4 \cdot 10^9$

Table 3.2: The results for synthetic data consisting of 200 points.

ϵ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\bar{\epsilon}_{quad}$	0.075	0.155	0.237	0.330	0.425	0.528	0.637	0.756	0.883
$\bar{\epsilon}_{ROR}$	0.049	0.098	0.148	0.210	0.293	0.387	0.505	0.657	0.860
STD ϵ_{quad}	0.006	0.011	0.012	0.016	0.017	0.016	0.015	0.015	0.011
STD ϵ_{ROR}	0.042	0.082	0.109	0.134	0.164	0.183	0.178	0.139	0.071
J	8	20	54	163	588	2,809	$2.1 \cdot 10^4$	$3.6 \cdot 10^5$	$4.6 \cdot 10^7$
J_{quad}	6	13	29	75	224	906	5,815	$1.0 \cdot 10^5$	$2.3 \cdot 10^7$
J_{ROR}	5	11	26	69	452	7,430	$3.4 \cdot 10^4$	$1.0 \cdot 10^6$	$1.2 \cdot 10^{10}$

outlier ratio is substantially higher. Since the number of iterations increases quickly with the outlier ratio, the average number of iterations for high outlier ratios is therefore much larger for ROR.

The results for synthetic data illustrate that over a large range of camera configurations, the quadric algorithm essentially performs best. Camera configurations which do not meet ROR’s assumptions make this algorithm increase the original outlier ratio.

3.4.2 Real data

We have also applied the algorithms to some real stereo images, which are shown in Appendix B. The results of applying the algorithms on the images are shown in Table 3.3.

Table 3.3: The results on the real image pairs. The quantities indicated are the total number of correspondences n , the outlier ratios ε , ε_{quad} and ε_{ROR} , and the number of iterations J , J_{quad} and J_{ROR} .

image pair	n	ε	ε_{quad}	ε_{ROR}	J	J_{quad}	J_{ROR}
books	740	0.74	0.679	0.165	$5.7 \cdot 10^4$	$1.3 \cdot 10^4$	14
pile of books	548	0.82	0.788	0.286	$7.5 \cdot 10^5$	$2.4 \cdot 10^5$	47
Valbonne church	299	0.58	0.512	0.056	1,996	697	5
U. British Columbia	911	0.56	0.481	0.089	1,441	452	7
U. British Columbia (rotated 180°)	911	0.56	0.481	0.598	1,441	452	2,712
corridor	262	0.43	0.380	0.082	234	129	6
Wadham college	921	0.71	0.655	0.183	$2.7 \cdot 10^4$	7,914	17
Wadham college (rotated 60°)	921	0.71	0.657	0.737	$2.7 \cdot 10^4$	8,243	$5.3 \cdot 10^4$
tea box	221	0.71	0.698	1	$2.7 \cdot 10^4$	$2.0 \cdot 10^4$	∞

Although the effective outlier ratio ε_{quad} is decreased compared to ε for all image pairs, the ROR algorithm shows impressive results here. Most image pairs apparently meet the assumptions that are needed for the algorithm. However, if we introduce some additional transformation like image rotation, the ROR algorithm is not able to handle it very well. We rotated the right images of all image pairs and noticed that ε_{ROR} generally increases. In two cases, which are indicated in Table 3.3, the increase was such that ε_{ROR} became larger than the original ε .

The image pair “tea box” provides a challenge for the algorithms. The box is rotated between the views, and due to the similar text on both sides many outliers arise from the same regions. As a result, the outlier distribution is somewhat struc-

tured and confuses the ROR algorithm. Our algorithm also has difficulty with this image pair, but it does show a reduction in the outlier ratio.

The given examples of image rotation and a somewhat special object may not occur that frequently in practice. Though, they serve to illustrate that the proposed method is able to perform consistently for various stereo pairs, rather than that it shows a superior performance for a constrained set of stereo images. In addition, we note that the complexity of the ROR algorithm makes it only useful for high outlier ratios. Since the standard RANSAC algorithm finishes well within 2 seconds for the “Valbonne church”, “U. British Columbia” and “Corridor” image pairs, it makes the application of ROR for these pairs ineffective.

3.5 Discussion

We have proposed an algorithm that compares the point correspondences from an image pair to a series of quadrics. Using the fact that outliers in the correspondences tend to have an arbitrary position with respect to a quadric, we can decrease the effective outlier ratio by gathering statistics over multiple quadrics. Subsequent application of RANSAC will therefore be less complex. The quadric algorithm has a low computational complexity and requires no assumptions on the data. Experiments show that the algorithm reduces the original outlier ratio of both synthetic and real data sets. In the synthetic data experiment, the quadric algorithm shows better performance than the method based on random rotations, which itself is much more complex. Although this method produces lower outlier ratios than the quadric algorithm for many real data sets, it is ineffective for small and moderate outlier ratios due to its higher complexity than standard RANSAC.

There is one specific quadric which may be useful in a different way. This is the most imbalanced quadric which is formed when both planes coincide. All inliers will have the same sign with respect to this quadric. The quadric is reached when the rays of the mean projections approximately lie in a single plane. Whether this condition is satisfied depends on the number of correspondences and the number of outliers among them. In general, the more inliers are present, the closer the backprojected means will be to the mean 3D position. The single plane quadric will then be an epipolar plane containing both camera centers and the mean world point. An epipolar plane places two constraints on the fundamental matrix, so fewer inliers are required to fix the remaining degrees of freedom. It has to be investigated whether such a quadric can be found in a reliable manner.

Chapter 4

Weighted Data Sampling in the RANSAC Algorithm

In this chapter we will consider again the adjustment of sampling probabilities for steering the sampling of correspondences. In contrast to the previous chapter, however, the confidence values are not acquired before but during RANSAC execution. The modification of the RANSAC algorithm that is proposed, feeds back information from previous iterations to adjust the sampling probabilities for the current iteration. It is shown to be applicable to different types of models, including the fundamental matrix. This chapter is based on [38].

4.1 Introduction

Several enhancements of the RANSAC algorithm have been proposed, which allow faster execution through a reduction of the number of iterations. In [60] the points are sampled in a small neighborhood of the first sampled point. Given that the first point is an inlier, it is likely that mostly inliers are sampled in its neighborhood. When compared to pure random sampling, this method has a higher probability of sampling inliers. In [12] a model optimization step is performed on the support sets of several hypothesized models. This is motivated by the fact that an all-inlier sample may only provide an approximation to the true model due to noise on the inliers. Optimization of the hypothesized model can compensate for this, and therefore reduces the number of iterations needed.

A different approach is the speeding up of the support evaluation [11]. Here a small number of randomly selected points is initially evaluated for support. Only when the hypothesized model has sufficient support points among this number, the remaining data is tested for support. Since it is unlikely that several data points obey a random model, a significant speedup is obtained during the support evalu-

ation phase. In [62] multiple hypothesized models are tested simultaneously using an incremental number of data points. Models having few supporting points with respect to the others are removed during the process, resulting in efficient support evaluation. In [52] the scheme from [11] was further improved, by optimizing the number of points used for the initial test.

As we have seen in Chapter 3, a change of the sampling probabilities of data points can decrease the effective outlier ratio of the data and likewise the number of RANSAC iterations. In this chapter we will also change the sampling probabilities though not a priori. We propose a weighted sampling strategy for model estimation among uniformly distributed outliers, where the sampling probabilities are changed during RANSAC execution. We will prove that when estimating hyperplanes among uniformly distributed outliers, the weighted sampling strategy can increase the probability of sampling inliers. By experiments we will show that this method can also be successfully applied to fundamental matrix estimation.

The weighted sampling method will reduce the necessary number of iterations under a small amount of additional computation. The additional computation will be compensated by the reduction in iterations, except in case of very “light” iterations where simple models are computed and few data points are observed. The method does not rely on the setting of any additional thresholds like in [60], where the selection of a good neighborhood size depends on the distribution of the data. It may be difficult to select the size for diverse data sets. Our method can potentially use fewer iterations than theoretically necessary; the model optimization step in [12] can only reduce the number of iterations to the theoretically required number, but is not designed to give a further improvement.

In Section 4.2 the proposed sampling strategy is explained and analyzed. The evaluation of this strategy when RANSAC is applied to linear regression and the estimation of the fundamental matrix is described in Section 4.3. Section 4.4 will conclude the chapter.

4.2 Weighted sampling

In this section we introduce the weighted sampling strategy, which is followed by an analysis of the method. Thereafter, the benefit for hyperplane estimation among uniformly distributed outliers is shown.

4.2.1 Method of point weighting

Since the proposed method is not only limited to fundamental matrix estimation, we use a more general formulation of the estimation problem. The data points are denoted by \mathbf{x}_i for $i = 1, \dots, n$ in \mathbb{R}^N and the points in a sample of size d by $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_d\}$. The number of inliers among these d points is indicated by d^{in} . We

also denote with H^T the subspace of \mathbb{R}^N containing all points within distance T of a particular model. This subspace necessarily includes $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_d\}$.

In the standard RANSAC algorithm the sampling of points is uniform; each data point has an equal probability of being selected. To reduce the complexity, we propose a weighted sampling strategy, which aims at increasing the sampling probabilities of inliers during RANSAC execution. Data points are sampled according to an evolving probability distribution, which favors the sampling of points found in previous support sets. Since the model is most probably being hypothesized in a dense point area, sampling from this area will therefore be further stimulated.

The probability distribution used for sampling is formed by a set of weights $\{w_1, \dots, w_n\}$ for each $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. In every iteration the weights of points in the support set S_j are increased, except for the sample $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_d\}$ (the reason for this exclusion will be explained later on). The probability distribution in iteration j will depend on the weights $\{w_1^{j-1}, \dots, w_n^{j-1}\}$ from the previous iteration $j-1$. The RANSAC algorithm with the proposed weighted sampling strategy is given in Fig. 4.1. Note that when the weights are not updated, standard uniform sampling is obtained.

4.2.2 Analysis of the effective inlier ratio

Weighted sampling should increase the probability of sampling an inlier from the data. In case of uniform sampling this probability is constant, namely $1 - \varepsilon$. In the proposed algorithm it will depend on the weights $\{w_1, \dots, w_n\}$, and is likely to be different after each iteration. Let us denote by ρ_j the probability of sampling an inlier *after* iteration j has finished. We can write ρ_j as

$$\rho_j = \frac{\sum_{i:\mathbf{x}_i \in \text{inliers}} w_i^j}{\sum_i w_i^j} = \frac{n(1 - \varepsilon) + \sum_{l=1}^j k_l}{n + \sum_{l=1}^j q_l + \sum_{l=1}^j k_l} \quad (4.1)$$

where k_j and q_j denote the number of inliers and outliers, respectively, in the support set S_j . Necessarily, we have that $|S_j| = k_j + q_j$. Also note that $\rho_0 = 1 - \varepsilon$, which is the original inlier probability of the data set. Examining (4.1) shows us that ρ_j will be difficult to evaluate directly, since the probability distribution of k_j is essentially unknown. To make the analysis tractable, we will use the expected value of ρ_j as a criterion for evaluation. In particular, weighted sampling is readily preferred over uniform sampling if for every iteration the expected inlier probability is non-decreasing, that is

$$E[\rho_j] \geq E[\rho_{j-1}] \quad j = 1, \dots, J \quad (4.2)$$

- $S_{max} = \emptyset$, $J = \infty$, $j = 1$, $w_i^0 = 1$ $i = 1, \dots, n$
- **while** $j < J$ **do**
 - Randomly select a sample $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_d\}$ from $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ according to

$$p(\mathbf{x}_i) = \frac{w_i^{j-1}}{\sum_{i=1}^n w_i^{j-1}} \quad i = 1, \dots, n$$
 and compute the corresponding model.
 - Determine the set of support points S_j for the model by verifying which points are within H^T .
 - Adjust the weights by

$$w_i^j = \begin{cases} w_i^{j-1} + 1 & \text{if } \mathbf{x}_i \in S_j \setminus \{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_d\} \\ w_i^{j-1} & \text{else} \end{cases}$$
 - **if** $|S_j| > |S_{max}|$ **then**
 - $J = \log(1 - p) \cdot \log^{-1} \left(1 - \left(\frac{|S_j|}{n} \right)^d \right)$
 - $S_{max} = S_j$
 - **end if**
 - $j = j + 1$
- **end while**
- Re-estimate the model based on the largest support set S_{max} .

Figure 4.1: The RANSAC algorithm with weighted sampling.

This condition will assure that the expected total number of iterations for the algorithm can only decrease during execution.

Before we evaluate (4.2) we will make a few assumptions. The first assumption is frequently made and concerns the distribution of the outliers, namely that the outliers have a uniformly distributed distance over a fixed range to any hypothesized model [77, 88]. For a given threshold T , we therefore have that the expected number of outliers within H^T is equal for all models, and we can write $E[q_j] = E[q]$ for all j . In case of fundamental matrix estimation, the uniform distribution is an acceptable approximation when the outlying feature matches have random positions in the images. For many image pairs this will hold indeed, but some image pairs may produce more structured outliers with clustered positions. Though, in case the structured outliers do not dominate the outliers, the assumption may still be a good approximation. The second assumption is that the inliers are influenced by Gaussian noise with standard deviation σ on each coordinate independently. The third assumption we make is formally stated below.

Assumption 1. *The expected number of inliers $E[k]$ in the support set S is a non-decreasing function of the number of inliers in the sample d^{in} .*

The most probable position for sampling an inlier is from a dense inlier area. Since the hypothesized model passes through the selected inlier, the subspace H^T is likely to contain some of the surrounding inliers. When more inliers are sampled, i.e. for larger d^{in} , this effect will be more prominent and motivates the assumption. Several experiments that we have performed validate the assumption.

The following lemma shows under what condition (4.2) is fulfilled.

Lemma 1. *$E[\rho_j] \geq E[\rho_{j-1}]$ for $j = 1, \dots, J$ if in the first iteration the expected number of inliers in the support set satisfies*

$$E[k_1] \geq \frac{1-\varepsilon}{\varepsilon} E[q] \quad (4.3)$$

Proof. In order to simplify the analysis, the expected values $E[\rho_j]$ and $E[\rho_{j-1}]$ will be approximated by the ratio of expected nominator and denominator values in (4.1). In this way, $E[\rho_j] \geq E[\rho_{j-1}]$ can be written as

$$\frac{n(1-\varepsilon) + \sum_{l=1}^j E[k_l]}{n + \sum_{l=1}^j E[q_l] + \sum_{l=1}^j E[k_l]} \geq \frac{n(1-\varepsilon) + \sum_{l=1}^{j-1} E[k_l]}{n + \sum_{l=1}^{j-1} E[q_l] + \sum_{l=1}^{j-1} E[k_l]} \quad (4.4)$$

After some algebraic manipulation and using $E[q_l] = E[q]$, this reduces to

$$E[k_j] (n\varepsilon + (j-1)E[q]) - E[q] \left(n(1-\varepsilon) + \sum_{l=1}^{j-1} E[k_l] \right) \geq 0 \quad (4.5)$$

Evaluating (4.5) for $j = 1$ results in (4.3). Using an inductive argument, we can show that if (4.5) holds for $j = m$ then it also holds for $j = m + 1$. The left side of (4.5) for $j = m + 1$ can be written as

$$\begin{aligned} & E[k_{m+1}] (n\varepsilon + mE[q]) - E[q] \left(n(1-\varepsilon) + \sum_{l=1}^m E[k_l] \right) \\ &= E[k_{m+1}] (n\varepsilon + (m-1)E[q]) - E[q] \left(n(1-\varepsilon) + \sum_{l=1}^{m-1} E[k_l] \right) \\ &\quad + E[q] (E[k_{m+1}] - E[k_m]) \\ &\geq E[k_m] (n\varepsilon + (m-1)E[q]) - E[q] \left(n(1-\varepsilon) + \sum_{l=1}^{m-1} E[k_l] \right) \\ &\geq 0 \end{aligned} \quad (4.6)$$

The first inequality follows from the induction hypothesis $E[\rho_m] \geq E[\rho_{m-1}]$ and $E[d_j^{in}] = dE[\rho_{j-1}]$, which results in $E[d_{m+1}^{in}] \geq E[d_m^{in}]$ and together with Assumption 1 this gives $E[k_{m+1}] \geq E[k_m]$. The second inequality follows directly from the induction hypothesis. \square

As shown by the Lemma, weighted sampling will perform at least as well as uniform sampling when (4.3) holds. The exclusion of the sample $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_d\}$ from increasing its weights, is because for the points in the sample (4.2) is always satisfied with equality (due to $E[d_j^{in}] = dE[\rho_{j-1}]$). In case the remaining support points satisfy (4.2) by a strict inequality, the sample $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_d\}$ will actually decrease the value of $E[\rho_j]$.

4.2.3 Hyperplane estimation

Next, we give a result related to the estimation of hyperplanes using the weighted sampling method.

Theorem 2. *When the model is a hyperplane and assuming uniformly distributed outliers and Gaussian noise with $\sigma < 0.21$ on the inliers, then $E[\rho_j] \geq E[\rho_{j-1}]$ for $j = 1, \dots, J$.*

Proof. Let a point in \mathbb{R}^N be denoted by $\mathbf{x} = (x_1, x_2, \dots, x_N)^\top$. A hyperplane with normal vector $\mathbf{N} = (a_1, a_2, \dots, a_N)^\top$ is given by $a_1x_1 + a_2x_2 + \dots + a_Nx_N + b = 0$. A hyperplane is therefore specified by (\mathbf{N}, b) , where we assume that \mathbf{N} has unit norm. The position of the inlying points \mathbf{x}_i^{in} , $i = 1, \dots, n(1-\varepsilon)$ in \mathbb{R}^N (under the

constraint that they are on the true model) can be described by a probability density function. We will denote by $p^{in}(b|\mathbf{N})$ the probability that an inlier is found on the hyperplane with offset b for a given normal \mathbf{N} . The inlying points are assumed to be independently distributed.

Since the inliers are influenced by Gaussian noise with standard deviation σ on each coordinate independently, the noise distribution in the direction of \mathbf{N} is also described by a Gaussian distribution with the same standard deviation. The distribution $p^{in}(b|\mathbf{N})$ can therefore be written as

$$\begin{aligned} p^{in}(b|\mathbf{N}) &= \int_u p(b|u)p(u) du \\ &= \frac{1}{\sqrt{2\pi}\sigma} \int_u e^{-\frac{(b-u)^2}{2\sigma^2}} p(u) du \end{aligned} \quad (4.7)$$

where u is the true inlier position (i.e. the true b) on the model.

Similarly, we denote by $p^{out}(b|\mathbf{N})$ the probability that an outlier \mathbf{x}_i^{out} , $i = 1, \dots, n\varepsilon$ is found on the hyperplane with offset b for a given normal \mathbf{N} . Due to the uniformly distributed distance of the outliers to any model, the probability $p^{out}(b|\mathbf{N})$ is equal for all models and we indicate it by $p^{out}(b|\mathbf{N}) \equiv p_o$. Note that $\int p^{in}(b|\mathbf{N}) db = 1$ for all \mathbf{N} and that $\int p_o db = 1$.

Let us first assume that $d^{in} = 0$. For a particular model to be selected, the points in the sample must all lie on the model. Therefore, for a given normal direction \mathbf{N} , the probability that offset b is selected is

$$\begin{aligned} p(b|\mathbf{N}) &= \frac{(p_o)^d}{\int (p_o)^d db} \\ &= p_o \end{aligned} \quad (4.8)$$

The number of inliers in H^T in the first iteration for a hyperplane with parameters \mathbf{N} and b is

$$\begin{aligned} E[k_1|\mathbf{N}, b] &= \sum_{i=1}^{n(1-\varepsilon)} p(\mathbf{x}_i^{in} \in H^T) \\ &= n(1-\varepsilon) \int_{b-T}^{b+T} p^{in}(B|\mathbf{N}) dB \end{aligned} \quad (4.9)$$

Using (4.8) we can now write

$$\begin{aligned} E[k_1|\mathbf{N}, d^{in} = 0] &= \int E[k_1|\mathbf{N}, b]p(b|\mathbf{N}) db \\ &= n(1-\varepsilon) \int \int_{b-T}^{b+T} p^{in}(B|\mathbf{N})p_o dB db \\ &= n(1-\varepsilon)2Tp_o \end{aligned} \quad (4.10)$$

The expected number of outliers $E[q|\mathbf{N}]$ in each H^T is equal, and we have $E[q|\mathbf{N}] = n\varepsilon 2T p_o$. As a result, we can write $E[k_1|\mathbf{N}, d^{in} = 0] = \frac{1-\varepsilon}{\varepsilon} E[q|\mathbf{N}]$ and averaging over all \mathbf{N} proves the claim for $d^{in} = 0$ by Lemma 1.

Let us now assume that $d^{in} = 1$. For a hyperplane with given normal direction \mathbf{N} , the probability that offset b is selected equals

$$\begin{aligned} p(b|\mathbf{N}) &= \frac{p^{in}(b|\mathbf{N})(p_o)^{d-1}}{\int p^{in}(b|\mathbf{N})(p_o)^{d-1} db} \\ &= p^{in}(b|\mathbf{N}) \end{aligned} \quad (4.11)$$

Combining (4.9) and (4.11) gives us

$$\begin{aligned} E[k_1|\mathbf{N}, d^{in} = 1] &= \int E[k_1|\mathbf{N}, b] p(b|\mathbf{N}) db \\ &= n(1-\varepsilon) \iint_{b-T}^{b+T} p^{in}(B|\mathbf{N}) p^{in}(b|\mathbf{N}) dB db \\ &= n(1-\varepsilon) \int_{-T}^T \int p^{in}(b+\tau|\mathbf{N}) p^{in}(b|\mathbf{N}) db d\tau \\ &\geq n(1-\varepsilon) \int_{-T}^T \int \int p(b+\tau|u) p(u) p(b|u) p(u) du db d\tau \\ &= n(1-\varepsilon) \int_{-T}^T \int \int p(u)^2 \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(b+\tau-u)^2}{2\sigma^2}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(b-u)^2}{2\sigma^2}} du db d\tau \\ &= n(1-\varepsilon) \frac{1}{2\pi\sigma^2} \int_{-T}^T \int p(u)^2 e^{-\frac{\tau^2}{4\sigma^2}} \int e^{-\frac{(b-(u-\frac{1}{2}\tau))^2}{\sigma^2}} db du d\tau \\ &= n(1-\varepsilon) \frac{1}{2\sqrt{\pi}\sigma} \int_{-T}^T e^{-\frac{\tau^2}{4\sigma^2}} d\tau \int p(u)^2 du \end{aligned} \quad (4.12)$$

where the inequality follows from $p(b_2 - b_1 = \tau) \geq p(b_2 - b_1 = \tau, u_2 = u_1)$ for the two inlier densities. The last equality follows from integrating the rightmost Gaussian over the full range of b , which yields $\sqrt{\pi}\sigma$.

Threshold T is usually chosen as 1.96σ , so that the probability that an inlier is in the support set of the true model is 0.95 (following a χ^2 distribution). The Gaussian integral in the last line of (4.12) for this choice of T yields

$$\frac{1}{2\sqrt{\pi}\sigma} \int_{-1.96\sigma}^{1.96\sigma} e^{-\frac{\tau^2}{4\sigma^2}} d\tau = 0.83 \quad (4.13)$$

The integral of $p(u)^2$ can be bound below using the Cauchy-Schwarz inequality [43]

$$\int p(u)^2 du \geq \frac{\left(\int p(u)p_o du\right)^2}{\int p_o^2 du} = \frac{p_o^2}{p_o} = p_o \quad (4.14)$$

and combining (4.12), (4.13) and (4.14) we have that

$$\begin{aligned} E[k_1 | \mathbf{N}, d^{in} = 1] &\geq n(1 - \varepsilon)0.83p_o \\ &\geq n(1 - \varepsilon)2Tp_o \quad \text{for } \sigma < 0.21 \end{aligned} \quad (4.15)$$

where T is 1.96σ . From (4.15) we therefore have that $E[k_1 | \mathbf{N}, d^{in} = 1] > \frac{1-\varepsilon}{\varepsilon} E[q | \mathbf{N}]$ for small $\sigma < 0.21$, and after averaging over all \mathbf{N} this completes the proof for $d^{in} = 1$ by Lemma 1.

In case $d^{in} > 1$ we can use Assumption 1, showing that $E[k_1]$ should increase for a larger given value of d^{in} . Since the expected number of outliers in H^T is still the same as for $d^{in} = 1$, (4.3) will also hold for $1 < d^{in} \leq d$. \square

Although the resulting range of σ is rather small, the use of lower bounds in the Theorem may suggest that it also holds for larger σ values.

We will show in the experiments that for hyperplane estimation, weighted sampling indeed requires fewer iterations than uniform sampling.

4.3 Experimental results

The RANSAC algorithm with uniform and weighted data sampling will be compared for linear regression and fundamental matrix estimation on both synthetic and real data sets. The re-estimation in the last step of RANSAC is not performed, but this will not affect the comparison. Most algorithms were implemented in C and ran on Intel Xeon 3.07 GHz / 3.2 GHz computers. The singular value decomposition needed for plane and fundamental matrix computation was taken from [68].

First the estimation of a simple hyperplane, a line, is considered. The number of required iterations from (2.25) is already relatively small due to $d = 2$. The reduction in number of iterations by weighted sampling is also small, since there are few iterations for the weighting to take effect. For the estimation of a plane ($d = 3$) the reduction will therefore be larger, and for the fundamental matrix ($d = 7$) even more.

In practice, the weighted sampling of points according to weights $\{w_1, \dots, w_n\}$ is achieved by means of cumulative weights $\{c_1, \dots, c_n\}$, which are given by $c_i = c_{i-1} + w_i$ with $c_1 = w_1$. The values of the cumulative weights need to be recalculated in every iteration, which requires one pass of additions over all $\{c_1, \dots, c_n\}$. A uniformly distributed random number between 0 and c_n is then drawn, and the

smallest index i_s for which c_{i_s} is larger than the random number corresponds to the selected point \mathbf{x}_{i_s} [40]. The search for i_s can be done efficiently with a binary search over $\{c_1, \dots, c_n\}$, which has on average a $\log(n)$ complexity.

4.3.1 Line fitting

Consider the example of line fitting in Fig. 4.2(a), where the model has slope 0.5 and intercept 20. The outliers are uniformly distributed and the outlier ratio is 75%. Running the weighted sampling results in the weight distribution as shown in Fig. 4.2(b)- 4.2(d). After 30 iterations the line already shows many high weights when compared to the outliers. Due to the ratio of initial and incremental weights, there is no incorrect clustering of high weights at a specific position in the plane.

For this type of dataset, we have determined the average performance of the sampling methods for outlier ratios from $\varepsilon = 0$ to $\varepsilon = 0.9$. For each outlier ratio 500 runs are performed with random line parameters and point distributions. The inliers are uniformly distributed along the line, and the outliers uniformly distributed in 2D-space. The coordinates x and y are bounded to the interval $[0, 100]$. We add unit variance Gaussian noise independently to the x and y coordinates of the inliers. We set the threshold for the orthogonal distance to the line to $T = 1.96$, so that 95% of the inliers will be accepted given the true model. Table 4.1 shows for $n = 50$ points the resulting number of iterations J , the size of the maximum support set $|S_{max}|$ and the running time t . Alternatively, in [33] some initial experiments have been performed recording the number of iterations for finding a fixed number of inliers.

From Table 4.1 it can be seen that weighted sampling reduces on average the required number of iterations and shows a smaller variation. The sizes of the maximal support sets found are comparable; weighted sampling gives a slightly larger support set with less variation, which indicates that there is a gain in precision of the estimated model. However, the average running times show that the additional complexity of weighted sampling cancels the reduction in number of iterations. A single RANSAC iteration is here relatively fast; a simple model is computed and there are few data points to evaluate.

It may appear that the weighted sampling method is equivalent to a fixed density-based prior, since weighted sampling stimulates the sampling from dense point areas. To show the difference in results, we have additionally shown in Table 4.1 the number of iterations needed for sampling according to density-based probabilities. It concerns a simple method where the prior point probabilities are proportional to the number of points in their local neighborhoods. The threshold for the neighborhood radius was set to 15, which gave the best results for the case $\varepsilon = 0.5$. Note that in practice, in the absence of a tuning stage, it may be difficult to come up with a good threshold value. In Fig. 4.3 a graphical comparison is shown of the iteration data in Table 4.1. Overall, the average number of iterations over 500 runs for the density-based method turns out to be higher than for weighted sampling. The case where the hypothesized model is nearly optimal, and many inlier weights are increased, is

where weighted sampling improves over the purely density-based method. The total complexity of the latter method will in general depend on the specific type of density estimation procedure, and requires just like weighted sampling a non-uniform sampling procedure.

Structured outliers may cause the weighted sampling to be ineffective or give worse performance than uniform sampling. This can be seen in Fig. 4.4 where a line is shown in data with 60% outliers. Half of the outliers lie around a point, and the resulting high point density causes increased point weights. If we run RANSAC with both sampling schemes for 500 times on this particular point set, we get as average and standard deviation 27.1 ± 4.4 iterations for uniform sampling, and 30.4 ± 13.1 iterations for weighted sampling. The average size and standard deviation of the support sets found are 40.6 ± 2.4 for uniform sampling, and 40.1 ± 4.1 for weighted sampling. The performance of weighted sampling is here inferior to uniform sampling. Although point weighting helps here in discerning structure from the rest of the data (see Fig. 4.4(d)), it does not discriminate well between the structures.

When the structures are evenly distributed in the space and have an equal point density, weighted sampling does not necessarily require more iterations than uniform sampling. This is illustrated by line fitting in real images for the purpose of straight edge detection. An edge detector will generally return multiple structures of equal pixel density. Applying the Sobel edge detector to the test images of size 256×256 pixels in Fig. 4.5 and subsequent thresholding, gives the corresponding edge maps. The RANSAC algorithm will now be repeatedly used to extract all straight edges in the images. Each edge that is returned by the algorithm is deleted from the image, and the extraction process continues until an edge with less than 50 pixels is returned. The whole process is repeated 500 times for averaging. We set the threshold for the orthogonal distance to the edge to $T = \sqrt{2}$ pixels. The results are shown in Table 4.3. Weighted sampling requires slightly fewer or the same number of iterations when compared to uniform sampling. The distribution of the pixels largely determines whether there is any improvement possible. A sparse set of lines like in Fig. 4.5(f) and 4.5(j) is far from being uniformly distributed, and therefore no improvement is obtained. A distribution like in Fig. 4.5(i) is more uniform and likewise suitable for the weighting of samples. Unfortunately, this only concerns the theoretical benefit of the algorithm, since in practice it does not imply a faster execution of the algorithm. The improvement is too small to compensate for the additional complexity of weighted sampling.

4.3.2 Plane fitting

In a similar way as for fitting a line, we have generated points in 3D for testing RANSAC on plane fitting. The inliers are uniformly distributed along the plane and the outliers uniformly distributed in the space. The x , y and z coordinates of the points are bounded to the interval $[0, 100]$. We add unit variance Gaussian noise independently to the x , y and z coordinates of the inliers. The threshold for the

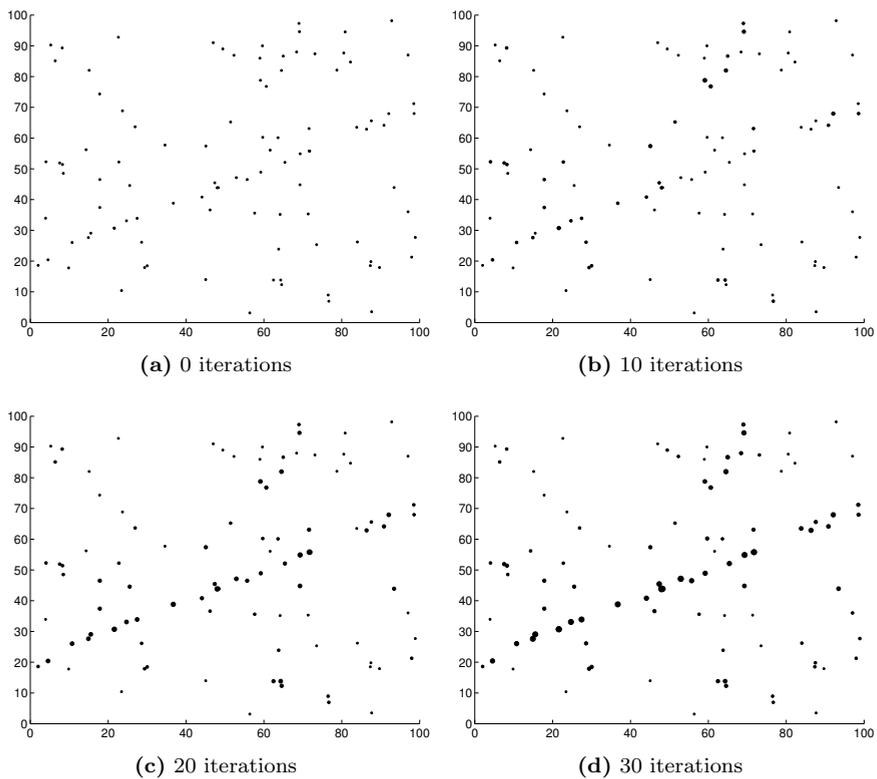


Figure 4.2: A data set with 75% outliers. The sizes of the points reflect their relative weights after several iterations.

Table 4.1: The results for uniform (U), density-based (D) and weighted (W) sampling in the synthetic line fitting experiment for 50 points and different outlier ratios ε . Indicated are the averages and standard deviations (\pm) for the executed number of iterations J and the maximum number of support points $|S_{max}|$, and the average running time t in milliseconds.

ε	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
J_U	3.6 ± 1.1	5.0 ± 1.2	6.7 ± 1.6	9.5 ± 2.2	13.3 ± 2.6	19.3 ± 4.3	29.2 ± 6.1	50.4 ± 12.0	98.4 ± 26.9	171 ± 39.1
J_D	3.8 ± 1.2	4.9 ± 1.0	6.5 ± 1.5	8.8 ± 1.5	12.5 ± 2.1	18.0 ± 2.7	28.1 ± 5.1	46.2 ± 9.7	95.1 ± 25.3	172 ± 40.3
J_W	3.6 ± 1.2	4.8 ± 1.1	6.6 ± 1.5	9.0 ± 1.8	12.5 ± 2.0	18.0 ± 3.0	27.0 ± 4.8	45.5 ± 9.9	92.4 ± 24.8	164 ± 38.3
$ S_{max} _U$	45.2 ± 2.9	40.9 ± 2.6	37.0 ± 2.4	32.6 ± 2.3	28.3 ± 2.0	23.9 ± 1.9	19.7 ± 1.7	15.3 ± 1.6	11.1 ± 1.5	8.4 ± 1.0
$ S_{max} _W$	45.4 ± 2.8	41.3 ± 2.3	37.2 ± 2.2	33.1 ± 2.0	28.8 ± 1.7	24.5 ± 1.7	20.3 ± 1.5	15.9 ± 1.5	11.4 ± 1.5	8.6 ± 1.1
t_U	0.06	0.08	0.08	0.12	0.16	0.22	0.32	0.56	1.04	1.76
t_W	0.06	0.08	0.10	0.14	0.16	0.22	0.34	0.58	1.16	2.02

Table 4.2: The results for the synthetic plane fitting experiment using 100 points.

ε	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
J_U	5.0 ± 1.8	7.6 ± 2.2	11.1 ± 3.0	17.3 ± 4.4	28.2 ± 7.3	45.0 ± 11.1	86.3 ± 23.0	178 ± 50.4	483 ± 159	$1,256 \pm 282$
J_D	5.3 ± 1.9	7.5 ± 2.0	10.7 ± 2.7	15.9 ± 3.2	24.4 ± 4.5	40.7 ± 7.6	75.9 ± 15.9	164 ± 39.4	450 ± 149	$1,256 \pm 279$
J_W	4.8 ± 1.7	7.0 ± 1.8	10.1 ± 2.3	15.0 ± 2.8	23.7 ± 4.3	38.9 ± 6.4	72.5 ± 12.7	155 ± 36.8	429 ± 141	$1,186 \pm 270$
$ S_{max} _U$	88.8 ± 5.6	80.4 ± 5.0	72.4 ± 4.3	63.5 ± 3.8	54.8 ± 3.6	47.2 ± 3.3	38.3 ± 2.9	30.4 ± 2.5	21.9 ± 2.3	15.7 ± 1.3
$ S_{max} _W$	89.3 ± 5.3	82.0 ± 4.3	74.0 ± 3.5	65.7 ± 3.1	57.3 ± 2.6	48.9 ± 2.3	40.0 ± 2.1	31.4 ± 2.3	22.7 ± 2.3	16.0 ± 1.3
t_U	0.22	0.32	0.40	0.60	0.90	1.40	2.60	5.22	13.92	35.20
t_W	0.24	0.28	0.40	0.54	0.80	1.28	2.30	4.92	12.90	34.60

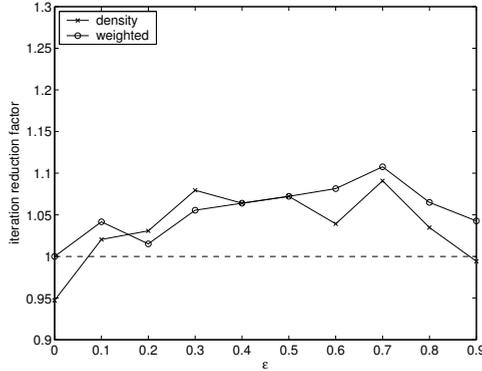


Figure 4.3: The ratios $\frac{J_U}{J_D}$ and $\frac{J_U}{J_W}$ for J_U , J_D and J_W from Table 4.1.

Table 4.3: The results for extracting all lines in the edge maps of Fig. 4.5. Indicated are the averages and standard deviations (\pm) for the total number of iterations $\sum J$ per image, and the total size of the maximum support sets $\sum |S_{max}|$, in case of uniform (U) and weighted (W) sampling.

	$\sum J_U (\cdot 10^4)$	$\sum J_W (\cdot 10^4)$	$\sum S_{max} _U (\cdot 10^3)$	$\sum S_{max} _W (\cdot 10^3)$
4.5(f)	1.45 ± 0.065	1.45 ± 0.069	1.41 ± 0.039	1.40 ± 0.042
4.5(g)	2.53 ± 0.18	2.48 ± 0.16	2.22 ± 0.045	2.22 ± 0.046
4.5(h)	5.95 ± 0.26	5.93 ± 0.25	2.01 ± 0.052	2.01 ± 0.055
4.5(i)	6.32 ± 0.39	6.09 ± 0.34	3.03 ± 0.068	3.05 ± 0.067
4.5(j)	0.32 ± 0.0061	0.32 ± 0.013	0.81 ± 0.012	0.81 ± 0.020

orthogonal distance to the plane is set to $T = 1.96$. The plane parameters are chosen randomly for each run. The results over 500 runs are shown in Table 4.2 for $n = 100$ points, and are partly depicted in Fig. 4.6.

Weighted sampling requires fewer iterations and has smaller variation than uniform sampling; the decrease is also larger than with line fitting. The sizes of the maximal support sets found are very much identical, though weighted sampling shows here a slight improvement over uniform sampling. The running times indicate that a single RANSAC iteration is already sufficiently complex for allowing weighted sampling to be faster.

We have also used a fixed density-based probability distribution for the sampling

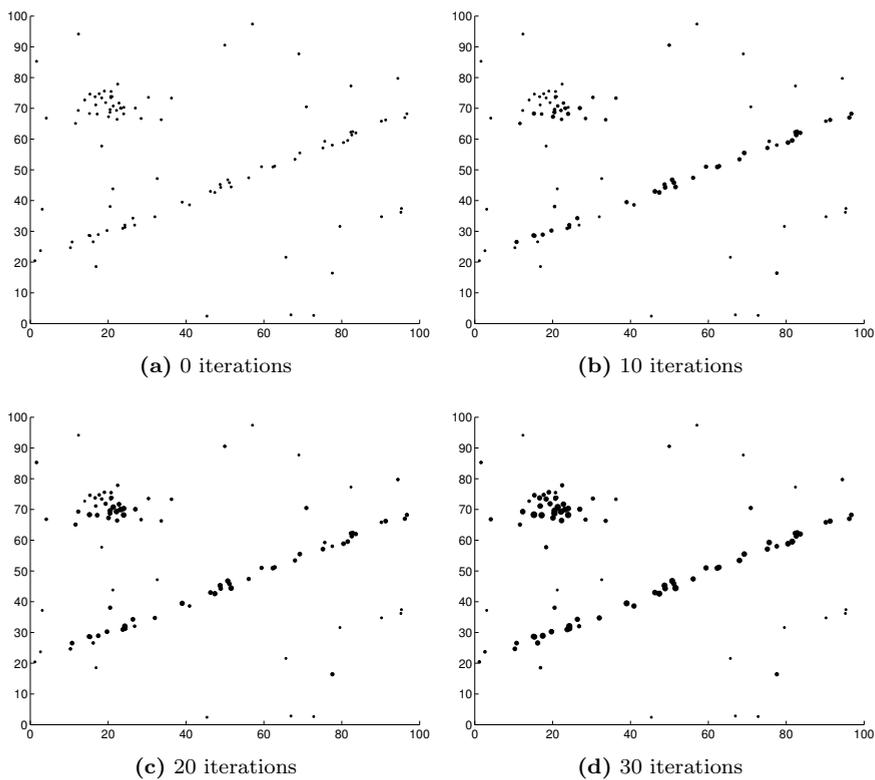


Figure 4.4: A data set with 60% outliers that are partly structured. The sizes of the points reflect their relative weights after several iterations.

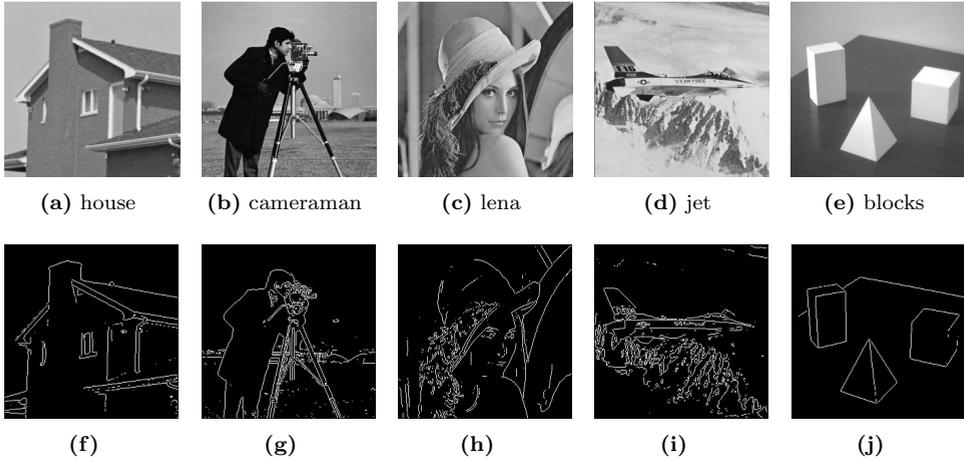


Figure 4.5: Test images and their corresponding edge maps.

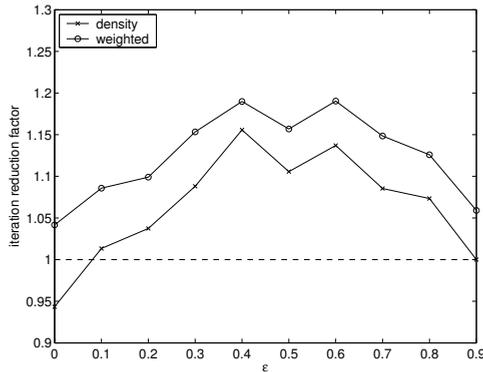


Figure 4.6: The ratios $\frac{J_U}{J_D}$ and $\frac{J_U}{J_W}$ for J_U , J_D and J_W from Table 4.2.

stage. The radius of the neighborhood was set to 15, which gave best results for the case $\varepsilon = 0.5$. The number of iterations for this method is also shown in Table 4.2. It can be seen that fewer iterations are required for weighted sampling when compared to the density-based method.

Additionally, we have applied RANSAC for the fitting of planes in range image

data to see the effects of structured outliers on the algorithm. Plane fitting in a range image can be used to find a suitable segmentation of the image [44, 91]. For our experiments we have used 10 images (“train” 0 to 9) from the ABW structured light scanner in the USF database¹. The images contain several different planar objects, and the intensity values correspond to the measured depth by the scanner. An example image is shown in Fig. 4.7. We have subsampled the images with a factor 2 to obtain 256×256 sized images, since each pixel in the image is a separate data point and using all of them makes computation very complex. We search with RANSAC for planes in the images and subsequently delete them. The repeated application of RANSAC is stopped when a plane is returned with support smaller than 500 pixels. As an example, the number of planes that can be extracted this way from Fig. 4.7 is about 11. The experiment is repeated 500 times for each image. The threshold for the orthogonal distance to the plane is set to $T = 2.5$. Table 4.4 shows the results of the experiments. Weighted sampling requires for most images on average fewer iterations, while the number of points returned on the planes is comparable to uniform sampling. However, as the running times indicate, the reduction is too small to make the algorithm of practical use for this type of data.

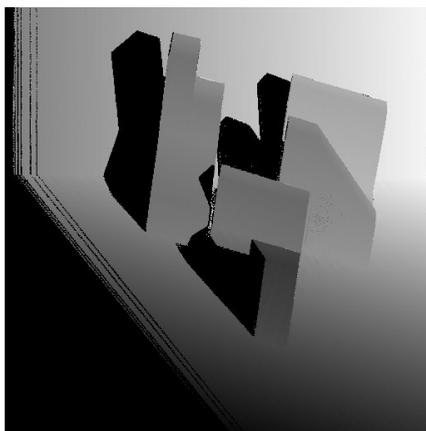


Figure 4.7: An example of the range images used in the experiments.

4.3.3 Fundamental matrix estimation

Next, we consider the estimation of the fundamental matrix using synthetic data, which is created by randomly generating points in a 3D cube. The cameras are

¹Available at <http://marathon.csee.usf.edu/range/seg-comp/images.html>.

Table 4.4: Uniform (U) and weighted (W) sampling results for the extraction of all planes in the ABW range images. Indicated are the averages and standard deviations (\pm) for the total number of iterations $\sum J$ per image, the running time t in seconds and the total size of the maximum support sets $\sum |S_{max}|$.

	$\sum J_U (\cdot 10^3)$	$\sum J_W (\cdot 10^3)$	t_U	t_W	$\sum S_{max} _U (\cdot 10^4)$	$\sum S_{max} _W (\cdot 10^4)$
image 0	9.50 \pm 1.42	9.45 \pm 1.40	19.9 \pm 2.37	23.1 \pm 3.82	6.29 \pm 0.032	6.29 \pm 0.033
image 1	3.41 \pm 0.48	3.25 \pm 0.49	5.47 \pm 0.64	5.61 \pm 0.78	6.37 \pm 0.029	6.37 \pm 0.029
image 2	5.36 \pm 0.75	5.39 \pm 0.74	9.58 \pm 1.53	10.2 \pm 1.87	6.41 \pm 0.034	6.41 \pm 0.034
image 3	2.12 \pm 0.37	2.03 \pm 0.36	4.13 \pm 0.70	4.49 \pm 0.81	6.41 \pm 0.028	6.41 \pm 0.026
image 4	2.88 \pm 0.50	2.51 \pm 0.43	5.71 \pm 0.73	5.55 \pm 0.73	6.41 \pm 0.029	6.41 \pm 0.027
image 5	2.55 \pm 0.34	2.45 \pm 0.39	4.32 \pm 0.57	4.45 \pm 0.70	6.39 \pm 0.029	6.39 \pm 0.029
image 6	5.03 \pm 0.68	4.72 \pm 0.64	9.80 \pm 1.18	9.52 \pm 1.21	6.36 \pm 0.029	6.36 \pm 0.032
image 7	1.62 \pm 0.29	1.55 \pm 0.27	3.74 \pm 0.39	4.00 \pm 0.51	6.44 \pm 0.026	6.44 \pm 0.025
image 8	2.00 \pm 0.36	1.98 \pm 0.37	3.61 \pm 0.57	3.95 \pm 0.69	6.40 \pm 0.028	6.39 \pm 0.030
image 9	1.91 \pm 0.37	1.78 \pm 0.33	4.09 \pm 0.52	4.26 \pm 0.58	6.44 \pm 0.029	6.44 \pm 0.032

placed at random positions on a sphere around the cube, where the radius of the sphere is two times the cube size. The points are between 20 and 60 focal lengths away from the cameras. For the inliers we add Gaussian distributed noise to both coordinates with standard deviation of about 0.3% of the image size. For the outliers we select two different space points, and use their non-corresponding projections as a pair. The distance measure used is the Sampson approximation (2.27) to the reprojection error, which is compared to the squared threshold T^2 . The threshold has been set to $T^2 = 3.84\sigma^2$ where σ^2 is the measured variance of the error with respect to the true fundamental matrix. The variance is averaged over several runs with each run having different point and camera positions. We also examined the behavior for two larger threshold values $5T^2$ and $20T^2$, which correspond to an increase in point distance by roughly a factor $\sqrt{5} = 2.24$ and $\sqrt{20} = 4.47$. Due to practical considerations, we stopped the RANSAC algorithm whenever the number of iterations exceeded 1,000,000. In these runs the number of iterations was set to one million. The results over 500 runs for 200 correspondences are given in Table 4.5 and Fig. 4.8, and shown graphically in Fig. 4.9.

It can be seen from these results that on the whole weighted sampling saves iterations and that it obtains better statistics for the maximum support sets. Though, there are some cases where the performance is worse. For example, in case there are no outliers ($\varepsilon = 0$), uniform sampling is better because weighted sampling is then discriminating between the inliers. Also for high outlier ratios ($\varepsilon \geq 0.7$) uniform sampling requires fewer iterations in some cases. This is probably caused by the low density of inliers in the space. For increasing threshold values, we see an improvement for high ε but at the same time worse results for moderate ε . Since in real imaged scenes the inliers are usually much more close when compared to a uniform distribution in space, better results can be expected from weighted sampling for real image pairs.

To verify this, we have used some of the images in Appendix B for the estimation of the fundamental matrix using both sampling methods. The square root of the threshold for the Sampson distance is set to $T = 1.5$ pixels. The results of 500 runs on the image pairs are shown in Table 4.6 and 4.7. Weighted sampling clearly improves upon uniform sampling in all aspects. Also note that weighted sampling approaches the true number of inliers much more closely. An exception is the image pair “pile of books” where the improvement is minimal. The explanation for this is that the inlying correspondences are very close to the outliers in the images, and the outliers are therefore frequently weighted too.

Also shown in Table 4.6 and 4.7 are the results for a density-based sampling method. This method follows the same principle as used in line and plane fitting, except that the distance is now computed in the 4D joint image space (x, y, x', y') . We have manually selected an appropriate threshold value in this space, namely 50 pixels. Although the number of iterations is slightly higher than for weighted sampling, the difference in running times is negligible. For image pair “pile of books”

Table 4.5: The results for fundamental matrix estimation from synthetic data with 200 correspondences and different outlier ratios ε . Indicated are the averages and standard deviations (\pm) for the executed number of iterations J , in case of uniform (U) and weighted (W) sampling.

ε	T^2		$5T^2$		$20T^2$	
	J_U	J_W	J_U	J_W	J_U	J_W
0.0	86.4 ± 35.8	95.3 ± 39.4	12.6 ± 6.1	14.2 ± 7.2	4.9 ± 2.5	5.5 ± 2.9
0.1	172 ± 64.2	171 ± 69.3	26.4 ± 12.5	25.3 ± 11.6	11.6 ± 4.4	11.3 ± 4.8
0.2	389 ± 149	326 ± 124	57.5 ± 24.9	48.1 ± 18.7	26.2 ± 9.7	23.3 ± 6.5
0.3	920 ± 387	663 ± 257	146 ± 64.9	103 ± 31.3	61.7 ± 21.7	52.6 ± 11.9
0.4	$2.74 \pm 1.15 \cdot 10^3$	$1.76 \pm 0.69 \cdot 10^3$	390 ± 161	270 ± 81.5	179 ± 68.1	136 ± 27.9
0.5	$8.84 \pm 3.81 \cdot 10^3$	$6.16 \pm 2.97 \cdot 10^3$	$1.35 \pm 0.63 \cdot 10^3$	967 ± 353	561 ± 208	439 ± 122
0.6	$3.64 \pm 1.62 \cdot 10^4$	$3.30 \pm 1.70 \cdot 10^4$	$5.49 \pm 2.57 \cdot 10^3$	$4.84 \pm 2.23 \cdot 10^3$	$2.18 \pm 0.89 \cdot 10^3$	$1.92 \pm 0.69 \cdot 10^3$
0.7	$2.16 \pm 1.05 \cdot 10^5$	$2.46 \pm 1.28 \cdot 10^5$	$3.26 \pm 1.43 \cdot 10^4$	$3.37 \pm 1.65 \cdot 10^4$	$1.18 \pm 0.52 \cdot 10^4$	$1.15 \pm 0.53 \cdot 10^4$
0.8	$9.87 \pm 0.63 \cdot 10^5$	$9.88 \pm 0.69 \cdot 10^5$	$3.63 \pm 1.81 \cdot 10^5$	$4.22 \pm 2.10 \cdot 10^5$	$9.01 \pm 3.14 \cdot 10^4$	$9.16 \pm 3.36 \cdot 10^4$
0.9	$1.00 \pm 0 \cdot 10^6$	$1.00 \pm 0 \cdot 10^6$	$9.70 \pm 0.72 \cdot 10^5$	$9.45 \pm 1.01 \cdot 10^5$	$1.38 \pm 0.33 \cdot 10^5$	$1.29 \pm 0.33 \cdot 10^5$

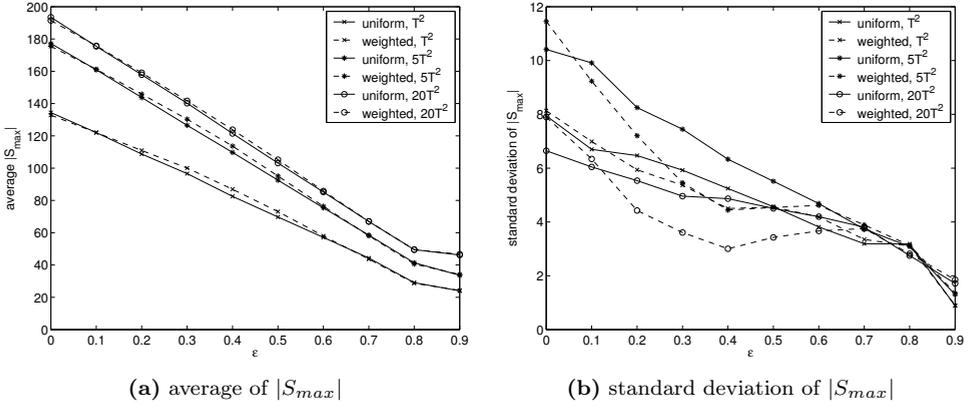


Figure 4.8: The statistics of $|S_{max}|$ for fundamental matrix estimation from synthetic data with 200 correspondences.

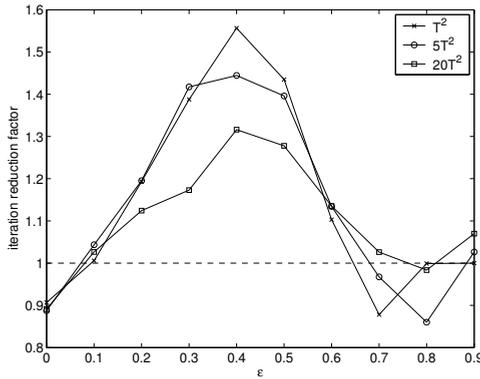


Figure 4.9: The ratio $\frac{J_U}{J_W}$ for the results from Table 4.5.

the performance is even better. This raises the question whether there is a reason to prefer weighted sampling over the density-based method.

There are two reasons which can be given in this respect. The first, already mentioned before, is that a density-based prior needs an appropriate value for the neighborhood threshold. In our method, this threshold was selected manually in

such a way that good results were obtained. When the threshold is chosen very small, e.g. equal to the threshold $T = 1.5$, the density-based method gives much worse results than weighted sampling. The weighted sampling method does not require tuning of an additional threshold value. The second issue is that the running times indicated do not include the computation of the density-based prior. This operation is of the order $O(n^2)$ and needs to be included in the computation time. Although depending on the specific implementation chosen, it can make density-based sampling less favourable than weighted sampling.

Recently it was shown that for having many inliers on a plane, the fundamental matrix computed from a sample of five such inliers and two outliers covers all the remaining plane inliers [13]. Such a fundamental matrix may mistakenly be considered correct due to its large support. One may suspect that this same phenomenon is responsible for the success of weighted sampling in the images containing planes. For example, in Fig. B.5 the walls of the building form two planes containing many inliers. However, the probability that five inliers are selected on the same wall is relatively low, since the inliers are evenly distributed over both walls. The probability this would happen is thus approximately 2^{-5} given that the sample contains five inliers. This indicates that it does not occur frequently enough to explain the improvement of weighted sampling. Evenmore, when we ran the experiment with the explicit requirement that weights can only be updated in case less than 5 inliers are sampled, we still noticed an increase in speed for weighted sampling.

4.4 Discussion

We have presented a weighted sampling scheme for the RANSAC algorithm, which adjusts the sampling probabilities of points to increase the probability of sampling inliers. The method is applicable for different types of models. It is proven that for estimation of a hyperplane among uniformly distributed outliers, the expected inlier ratio in the samples obtained this way is non-decreasing. Experiments on line and plane fitting in synthetic data generated under these conditions, confirm that weighted sampling reduces the number of RANSAC iterations. Experiments on real data with multiple lines or planes, show that weighted sampling roughly gives comparable performance in terms of the number of iterations but takes more time to complete. Furthermore, it has been shown that fundamental matrix estimation can benefit from the application of weighted sampling.

If we use sampling probabilities proportional to the local point densities, we need only slightly more RANSAC iterations when compared to the weighted sampling method. Weighted sampling seems to follow a similar strategy as the density-based method, since points are sampled independently and are most likely to originate from dense point areas. Though, the method not only benefits these particular points, but also points which together form a good model. This holds in particular for point subsets of the true model, which are then sampled more frequently and can steer

Table 4.6: The results in case of uniform (U), weighted (W) and density-based (D) sampling for fundamental matrix estimation from the real images. Indicated are the averages and standard deviations (\pm) for the executed number of iterations J and the running time t in seconds.

image pair	ε	J_U	J_D	J_W	t_U	t_D	t_W
Univ. BC	0.56	$2,649 \pm 562$	$1,757 \pm 154$	$1,654 \pm 104$	1.13 ± 0.28	0.73 ± 0.07	0.74 ± 0.09
Corridor	0.43	466 ± 160	329 ± 50.7	300 ± 30.5	0.08 ± 0.03	0.05 ± 0.01	0.05 ± 0.01
Boat	0.49	549 ± 53.4	524 ± 37.8	511 ± 31.0	0.11 ± 0.02	0.10 ± 0.01	0.10 ± 0.02
Wadham	0.71	$70,822 \pm 29,311$	$32,446 \pm 3,379$	$30,788 \pm 2,599$	29.0 ± 12.4	13.4 ± 1.41	13.7 ± 1.90
Valbonne church	0.58	$2,559 \pm 628$	$1,768 \pm 66.2$	$1,714 \pm 51.7$	0.49 ± 0.13	0.32 ± 0.01	0.34 ± 0.05
pile of books	0.82	$4.03 \pm 0.71 \cdot 10^5$	$3.46 \pm 0.47 \cdot 10^5$	$3.78 \pm 0.49 \cdot 10^5$	114 ± 24.2	98.5 ± 13.5	113 ± 19.0

Table 4.7: The averages and standard deviations (\pm) for the maximum support sets $|S_{max}|$ found in the real images.

image pair	# inliers	$ S_{max} _U$	$ S_{max} _D$	$ S_{max} _W$
Univ. BC	399	372 ± 11.0	391 ± 4.9	394 ± 3.7
Corridor	150	139 ± 5.8	143 ± 2.8	145 ± 2.0
Boat	184	181 ± 2.6	183 ± 2.0	183 ± 1.6
Wadham	264	241 ± 13.8	261 ± 3.7	263 ± 3.1
Valbonne church	127	123 ± 3.7	128 ± 0.7	129 ± 0.6
pile of books	97	109 ± 2.9	111 ± 2.4	109 ± 2.2

the sampling towards the true model. A disadvantage of the density-based approach is that it requires some scale estimate of the data. Only then a good measure of proximity is available to compute the point density.

Although the chosen weighting scheme gives good results, it may not be optimal; different weighting strategies could be explored to find the most useful one. We can think of weight values that depend on the size of the support set or the number of executed iterations so far. Care must be taken, however, that the sampling does not get stuck at a specific point in the model space. Additional measures may be taken to prevent this by examining the variation in hypothesized models or the current weight distribution.

A possible adjustment of the algorithm is to use weighting only in a subset of iterations. Although this may seem to hamper the speed, it will reduce the chance of the algorithm getting stuck at some point. The weight distribution is basically prevented to evolve so far that only a few models remain possible. Especially when multiple models are present, this issue becomes more important. One possibility is to let the update take place when the largest support set so far is found [12], which happens approximately $\log(J)$ times. This may still give a reasonable performance, while the number of weight updates is limited.

Chapter 5

A Six-Point RANSAC Algorithm for Fundamental Matrix Estimation

In this chapter we will consider a combination of the RANSAC algorithm and the Hough transform. The model will be computed by sampling a smaller than minimal subset, followed by a voting process of the remaining data. To be able to use the combined method for this purpose, an adequate parameterization of the model in the Hough space is required. We will show that in case of hyperplane and fundamental matrix estimation, there is a similar and very general parameterization possible. This allows both models to be estimated in a very efficient manner. This chapter is based on [37].

5.1 Introduction

The Hough transform determines for every data point the parameter subspace of models it supports, and increases the votes in the Hough space for all these models. An extension of this principle is to vote for sets of data points instead of single points. The subspace of supported models is then smaller, while the number of different point sets is larger. For example, a hyperplane in \mathbb{R}^N is specified by N points, and a single point imposes a $N - 1$ dimensional subspace of supported models in the Hough space. When a pair of points is considered, the set of supported models is a $N - 2$ dimensional subspace. The voting process is then faster, but we have to consider $\binom{n}{2}$ different pairs instead of only n points. The limiting case is when precisely sets of N points are selected, which then results in a single point in the Hough space. This is the principle of the randomized Hough transform [93]. Instead of the total

number of possible sets $\binom{n}{N}$, only a small number of random sets is selected which is sufficient to find the best model.

In contrast to the Hough transform, the RANSAC algorithm samples N points and verifies the amount of support for the corresponding model. In view of the above, it is also possible to sample less than N points and verify the support for each supported model in the parameter subspace. This use of RANSAC in combination with the Hough transform has been proposed in [65, 66] to improve the efficiency and quality of model estimation. It was argued that using sets of $N - 1$ points is probably the best choice in terms of efficiency. This results in a one-dimensional subspace of models, which may be parameterized by a single quantity. Then there is no need to accumulate the large N dimensional Hough space.

The necessary number of iterations of the combined RANSAC and Hough method is clearly lower than for standard RANSAC; only sets of $N - 1$ instead of N points are sampled for forming model hypotheses. The quality of the model can be improved by using an error propagation mechanism for the data. Uncertainty in the point positions is translated into an error for the supported models. This error is then projected onto the one-dimensional subspace of models.

As discussed in Section 2.4.3, an explicit parameterization of the fundamental matrix in the Hough space is impractical. To be able to use the method in [65], we propose a new parameterization for hyperplanes which can also be applied to the fundamental matrix. The parameterization is based on the nullspace of a sample, where the sample will contain one point less than the minimally required number. For hyperplane estimation, we can include the threshold for the support set directly into the voting process. As a result, the whole range of models supported by the remaining data is taken into account. For fundamental matrix estimation, the correspondences will vote for single models. The resulting estimation by 6-point samples will be very efficient due to the reduced number of iterations. Error propagation is not incorporated in our method, since no explicit parameterization of the model is used. Note that the standard RANSAC algorithm also neglects noise effects of points in the sample [12].

In Section 5.2 the proposed parameterization technique is discussed for hyperplane estimation. Section 5.3 describes the application of the method to fundamental matrix estimation, and Section 5.4 contains specific implementation issues. In Section 5.5, hyperplane estimation is evaluated for synthetic data and range data. Fundamental matrix estimation is evaluated on both synthetic and real correspondences. Section 5.6 will conclude the chapter.

5.2 Hyperplane estimation

The data points \mathbf{x}_i for $i = 1, \dots, n$ in \mathbb{R}^N will be denoted by $\mathbf{x} = (x_1, x_2, \dots, x_N)^\top$. A hyperplane with normal vector $\mathbf{n} = (a_1, a_2, \dots, a_N)^\top$ and offset b is given by $a_1x_1 + a_2x_2 + \dots + a_Nx_N + b = 0$. In short, the parameters of the hyperplane will

be indicated by $\mathbf{h} = (\mathbf{n}^\top b)^\top$. The random samples that will be drawn consist of $N - 1$ points $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_{N-1}\}$, and solving for the hyperplane

$$\begin{bmatrix} \tilde{\mathbf{x}}_1^\top & 1 \\ \tilde{\mathbf{x}}_2^\top & 1 \\ \vdots & \vdots \\ \tilde{\mathbf{x}}_{N-1}^\top & 1 \end{bmatrix} \mathbf{h} = \mathbf{0} \quad (5.1)$$

yields a two-dimensional space $\{\mathbf{h}_1, \mathbf{h}_2\}$ for \mathbf{h} . This nullspace can in practice be computed by a singular value decomposition of the lefthand-side matrix. If the sample $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_{N-1}\}$ contains only inliers, then the true hyperplane can be given by a linear combination of the nullspace vectors as

$$\mathbf{h} = \alpha \mathbf{h}_1 + (1 - \alpha) \mathbf{h}_2 \quad (5.2)$$

The value of α can be found by solving $(\mathbf{x}^\top \mathbf{1}) \mathbf{h} = 0$ for another inlying point \mathbf{x} , and should be the same for all other inliers. The outliers will produce different values for α .

To find the true value of α we use a Hough-based voting mechanism for the remaining $n - N + 1$ data points [65]. We could use the projections of \mathbf{x} onto \mathbf{h}_1 and \mathbf{h}_2 directly for computing α , but this may result in α values which are difficult to quantize. In particular, the nullspace vector with the largest singular value, say \mathbf{h}_1 , is likely to constitute the largest part of \mathbf{h} and therefore $\alpha \approx 1$. The binning of many values close to 1 and possibly some values far from 1 is impractical. It would be more convenient to have an α with equiprobable values over a large range.

For this purpose, we will make use of an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2\}$ for the space spanned by \mathbf{n}_1 and \mathbf{n}_2 , which are the normals in \mathbf{h}_1 and \mathbf{h}_2 from (5.2). We will take a point $\tilde{\mathbf{x}}_1$ from the sample, and project all vectors $\mathbf{x}_i - \tilde{\mathbf{x}}_1$ for $i = 1, \dots, n$ (except those from the sample) onto this basis. The point $\tilde{\mathbf{x}}_1$ can be seen as the origin for the space spanned by $\{\mathbf{u}_1, \mathbf{u}_2\}$, which is shown in Fig. 5.1 for a line in 2D. From (5.2) we have that $\mathbf{n} = \alpha \mathbf{n}_1 + (1 - \alpha) \mathbf{n}_2$, and since \mathbf{n}_1 and \mathbf{n}_2 are linear combinations of $\{\mathbf{u}_1, \mathbf{u}_2\}$ we can write

$$\mathbf{n} = c_1 \mathbf{u}_1 + c_2 \mathbf{u}_2 \quad (5.3)$$

for certain values c_1 and c_2 . It then follows, that for the inliers the ratio of projections onto \mathbf{u}_2 and \mathbf{u}_1 becomes

$$\begin{aligned} \frac{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_2}{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1} &= \frac{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top (\mathbf{n} - c_1 \mathbf{u}_1) \frac{1}{c_2}}{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1} \\ &= \frac{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1 \frac{-c_1}{c_2}}{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1} \\ &= \frac{-c_1}{c_2} \end{aligned} \quad (5.4)$$

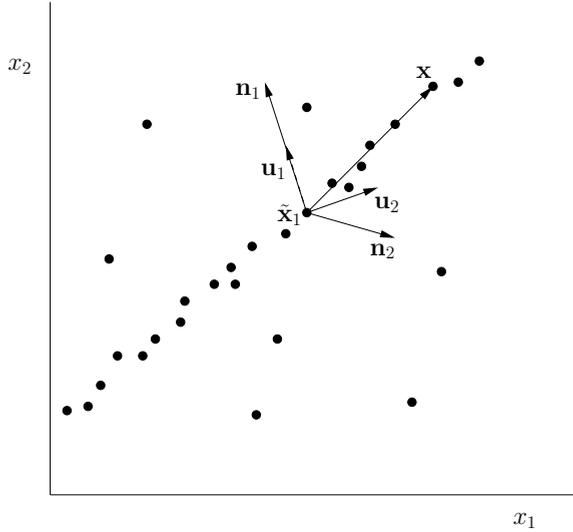


Figure 5.1: The sampled point $\tilde{\mathbf{x}}_1$ will serve as the origin for the space spanned by $\{\mathbf{n}_1, \mathbf{n}_2\}$. Each point \mathbf{x} is projected onto this space by projecting the vector $\mathbf{x} - \tilde{\mathbf{x}}_1$ onto the orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2\}$.

since $(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{n} = -b - (-b) = 0$. The outliers will produce different values for the projection ratio since in that case $\mathbf{x}^\top \mathbf{n} \neq -b$. The projection ratio in (5.4) will cover a relatively large range of values, and the angle γ of the projected vector $\mathbf{x} - \tilde{\mathbf{x}}_1$ with respect to the basis $\{\mathbf{u}_1, \mathbf{u}_2\}$

$$\gamma = \arctan \left(\frac{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_2}{(\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1} \right) \quad (5.5)$$

offers a quantity which can conveniently be used in a voting space.

In principle, not only the hyperplane which crosses a point \mathbf{x} should receive a vote, but all possible hyperplanes that are within allowable distance of the point. A data point will support a hyperplane if its orthogonal distance to the hyperplane is smaller than T , see Fig. 5.2. Here the angle β determines for which models the indicated point can possibly be a support point, and we have $\sin(\beta) = \frac{T}{\delta}$ where $\delta = \sqrt{((\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1)^2 + ((\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_2)^2}$. A data point will vote for all angles in the range $[\gamma - \beta, \gamma + \beta]$. We note that the distance from point \mathbf{x} to the hyperplane is equal to the projected distance in the space spanned by $\{\mathbf{u}_1, \mathbf{u}_2\}$, since the component of \mathbf{x} that lies outside this space is orthogonal to it.

The angle γ will be measured in degrees and we choose to use a voting space of 180 bins; one bin for each degree from -90 to 89. After calculating this angle for

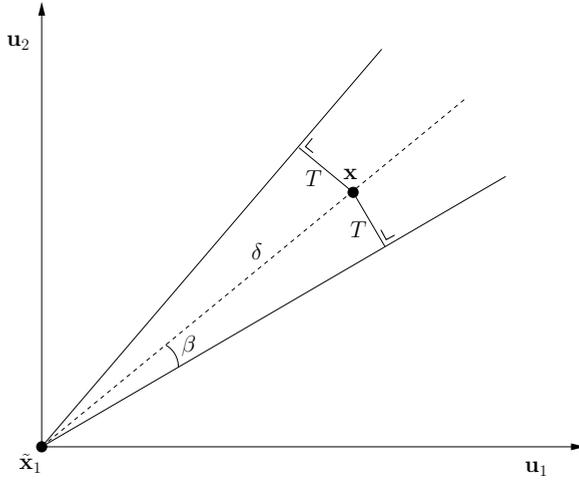


Figure 5.2: The projection of the point \mathbf{x} in the frame $\{\mathbf{u}_1, \mathbf{u}_2\}$. There is a range of hyperplanes which the point supports. The maximum angle β for the range depends on threshold T and the length δ of the projection of the vector $\mathbf{x} - \tilde{\mathbf{x}}_1$.

all points, it should result in a large number of votes in the bin of the true angle. A drawback of using all data points for voting, is that the voting operation may become quite complex for large data sets. Following the concept of the probabilistic Hough transform [42], we can also examine a subset of randomly sampled data points and calculate the best angle for this subset. This should give a sufficiently accurate estimate of the angle γ while making the voting process much faster. In the experiments we have chosen for a total of 100 randomly sampled data points, and only in case $n \leq 100$ we use all data.

The bin containing most votes determines the angle γ^* , for which the final hyperplane is calculated according to

$$\mathbf{h} = \begin{pmatrix} \mathbf{u}_1 + \tan(\gamma^* + \frac{1}{2}\pi)\mathbf{u}_2 \\ -\tilde{\mathbf{x}}_1^\top(\mathbf{u}_1 + \tan(\gamma^* + \frac{1}{2}\pi)\mathbf{u}_2) \end{pmatrix} \quad (5.6)$$

where point $\tilde{\mathbf{x}}_1$ is taken from the sample.

The complete estimation algorithm, incorporating all steps, is shown in Fig. 5.3.

5.3 Fundamental matrix estimation

The fundamental matrix can be estimated by following roughly the same technique as for hyperplane estimation. However, there are two major differences with the

- $j = 1, \quad J = \infty, \quad S_{max} = \emptyset$
- **while** $j < J$ **do**
 - Randomly select $N - 1$ points $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_{N-1}\}$ and compute the nullspace $\{\mathbf{h}_1, \mathbf{h}_2\}$ by solving (5.1).
 - Determine the orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2\}$ for the space spanned by the normals in $\{\mathbf{h}_1, \mathbf{h}_2\}$.
 - **if** $n > 100$ **then**
 - form the set C by randomly selecting 100 points from $\{\mathbf{x}_1, \dots, \mathbf{x}_n\} \setminus \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_{N-1}\}$
 - **else**
 - form $C = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \setminus \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_{N-1}\}$
 - **for each** \mathbf{x} **in** C **do**
 - Compute $\delta = \sqrt{((\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_1)^2 + ((\mathbf{x} - \tilde{\mathbf{x}}_1)^\top \mathbf{u}_2)^2}$ and $\beta = \lfloor \arcsin(\frac{T}{\delta}) \rfloor$
 - Determine γ according to (5.5) and round the angle to the nearest degree.
 - Add one vote for the angles $[\gamma - \beta, \gamma + \beta]$.
 - Determine the angle γ^* for which the number of votes is maximum.
 - Construct \mathbf{h} from (5.6) using γ^* .
 - Determine the set of support points S_j for \mathbf{h} by verifying which points are within distance T .
 - **if** $|S_j| > |S_{max}|$ **then**
 - $J = \log(1 - p) \cdot \log^{-1} \left(1 - \left(\frac{|S_j|}{n} \right)^{N-1} \right)$
 - $S_{max} = S_j$
 - $j = j + 1$
- Re-estimate the hyperplane based on the largest support set S_{max} .

Figure 5.3: The RANSAC-Hough algorithm for hyperplane estimation using a one-dimensional voting space.

preceding scenario.

First, the 7-point algorithm uses the singularity constraint to determine the fundamental matrix. After seven correspondences are selected, solving (2.22) yields a two-dimensional nullspace. Then the singularity constraint of the fundamental matrix needs to be used to find the solution. If we sample six points, the resulting nullspace is three-dimensional. We would like to use the singularity constraint for removing one dimension and use voting to find the final solution. Unfortunately, the singularity constraint on the three-dimensional nullspace is a cubic polynomial in two variables, which does not allow voting with respect to a fixed pair of nullspace vectors. As a result, we have to solve the singularity constraint for each correspondence individually. The complexity of the algorithm will therefore increase, but as we have already indicated, a subset of the data points will suffice in the voting process.

Second, since there is no fixed two-dimensional nullspace during voting, we can not calculate the range of allowable models as in Fig. 5.2. The number of fundamental matrices consistent with a seventh correspondence will be either one or three, just like for the 7-point algorithm. Therefore, there is no range of matrices for which the Sampson distance in (2.27) can be evaluated, and votes are cast for either one or three separate angles.

To start the estimation process we sample 6 correspondences $\{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \dots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$, and solve

$$\begin{bmatrix} \tilde{x}'_1 \tilde{x}_1 & \tilde{x}'_1 \tilde{y}_1 & \tilde{x}'_1 & \tilde{y}'_1 \tilde{x}_1 & \tilde{y}'_1 \tilde{y}_1 & \tilde{y}'_1 & \tilde{x}_1 & \tilde{y}_1 & 1 \\ \vdots & \vdots \\ \tilde{x}'_6 \tilde{x}_6 & \tilde{x}'_6 \tilde{y}_6 & \tilde{x}'_6 & \tilde{y}'_6 \tilde{x}_6 & \tilde{y}'_6 \tilde{y}_6 & \tilde{y}'_6 & \tilde{x}_6 & \tilde{y}_6 & 1 \end{bmatrix} \mathbf{f} = \mathbf{0} \quad (5.7)$$

which results in a three-dimensional space of solutions

$$\mathbf{f} = \alpha \mathbf{f}_1 + \beta \mathbf{f}_2 + (1 - \alpha - \beta) \mathbf{f}_3 \quad (5.8)$$

If we take a single correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ and solve

$$\begin{pmatrix} x'x & x'y & x' & y'x & y'y & y' & x & y & 1 \end{pmatrix} \mathbf{f} = 0 \quad (5.9)$$

for \mathbf{f} from (5.8) we get a linear constraint in α and β . When the correspondence is an inlier, the true values for α and β will satisfy this constraint. Let the resulting linear relation be $\beta = r\alpha + g$. Then we use the singularity constraint

$$\det(\alpha F_1 + (r\alpha + g)F_2 + (1 - \alpha - (r\alpha + g))F_3) = 0 \quad (5.10)$$

where F_1 , F_2 , and F_3 are the 3×3 matrices containing the elements of \mathbf{f}_1 , \mathbf{f}_2 and \mathbf{f}_3 , respectively. This will result in either one or three real solutions for α and thus for \mathbf{f} . Now, writing the vectors \mathbf{f}_1 , \mathbf{f}_2 and \mathbf{f}_3 in (5.8) as $\mathbf{f}_1 = (\mathbf{n}_1^\top \ b_1)^\top$, $\mathbf{f}_2 = (\mathbf{n}_2^\top \ b_2)^\top$ and $\mathbf{f}_3 = (\mathbf{n}_3^\top \ b_3)^\top$, we construct an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ from $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$.

This basis is used for the projection of the solutions for \mathbf{f} . In particular, we calculate the angles

$$\gamma_1 = \arctan \left(\frac{(f_1 \cdots f_8) \mathbf{u}_2}{(f_1 \cdots f_8) \mathbf{u}_1} \right) \quad \gamma_2 = \arctan \left(\frac{(f_1 \cdots f_8) \mathbf{u}_3}{(f_1 \cdots f_8) \mathbf{u}_2} \right) \quad (5.11)$$

and use them to cast a vote in a two-dimensional array. The angles will be rounded towards full degrees in the range -90 to 89.

As in hyperplane estimation, we do not use all data points during voting. When the data set contains more than 100 correspondences, only 100 randomly selected correspondences are considered. Examples of the votes casted in a 180×180 bins array for the image pair “Wadham college” (see Appendix B) are shown Fig. 5.4 and 5.5. The image pair contains in total 921 correspondences. Fig. 5.4 shows the votes resulting from a 6-points sample containing outliers, in case of 100 randomly selected correspondences and all 921 correspondences. It can be seen that in both cases the votes are rather spread in the array. In Fig. 5.5 the sample consists of only inliers; there is a clear peak for the true pair of angles in case of 921 correspondences. This peak coincides with the one for 100 randomly selected correspondences. One hundred correspondences are here clearly sufficient to arrive at the right pair of angles.

After having located the values of γ_1^* and γ_2^* for the bin containing most votes, we can find the first eight elements of the corresponding \mathbf{f} by

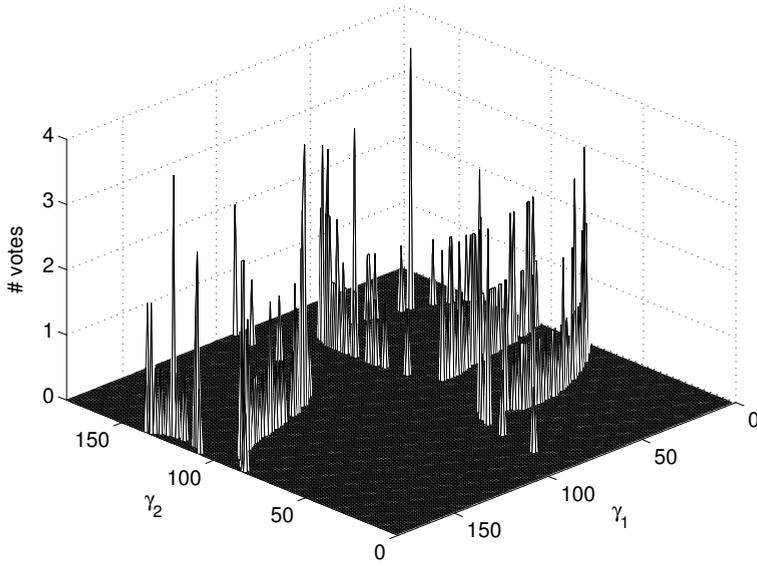
$$\begin{pmatrix} f_1 \\ \vdots \\ f_8 \end{pmatrix} = \mathbf{u}_1 + \tan(\gamma_1^*) \mathbf{u}_2 + \tan(\gamma_1^*) \tan(\gamma_2^*) \mathbf{u}_3 \quad (5.12)$$

and the last element by

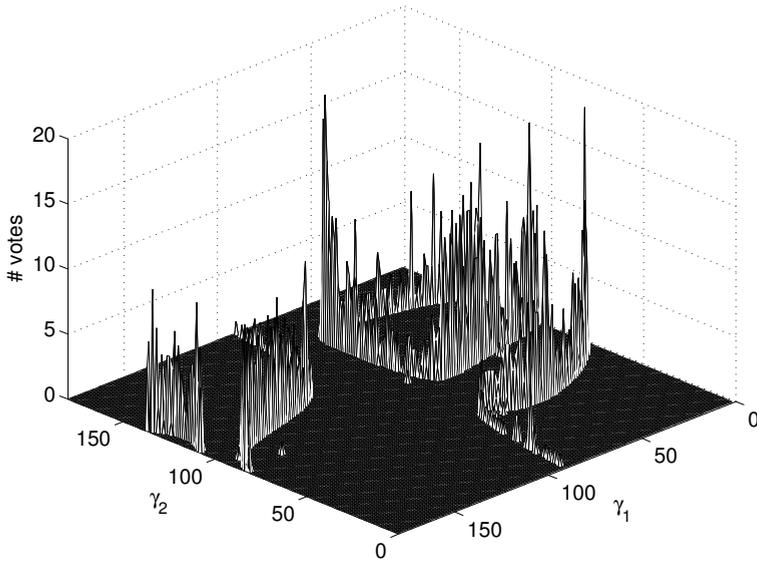
$$f_9 = - \begin{pmatrix} \tilde{x}'_1 \tilde{x}_1 & \tilde{x}'_1 \tilde{y}_1 & \tilde{x}'_1 & \tilde{y}'_1 \tilde{x}_1 & \tilde{y}'_1 \tilde{y}_1 & \tilde{y}'_1 & \tilde{x}_1 & \tilde{y}_1 \end{pmatrix} \begin{pmatrix} f_1 \\ \vdots \\ f_8 \end{pmatrix} \quad (5.13)$$

The correspondence $\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1$ is part of the 6-point sample, and therefore lies on the final \mathbf{f} .

The fundamental matrix that is found this way does not automatically satisfy the singularity constraint. Due to the rounding effect in the voting array, the matrix will slightly deviate from a singular one. We can solve this by applying the SVD to this matrix, and setting the smallest singular value to zero [28]. A prerequisite for this to work properly is a normalization of the correspondences. This entails a translation which results in zero means for the (x, y) coordinates, followed by a scaling which makes their average distance to the origin equal to $\sqrt{2}$. The transformation is applied to both images' correspondences independently. Before the support of the

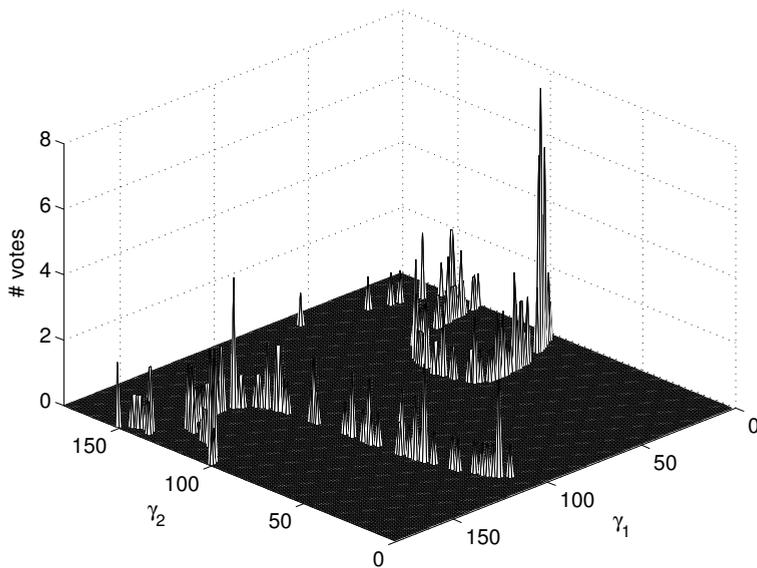


(a) 100 randomly selected correspondences

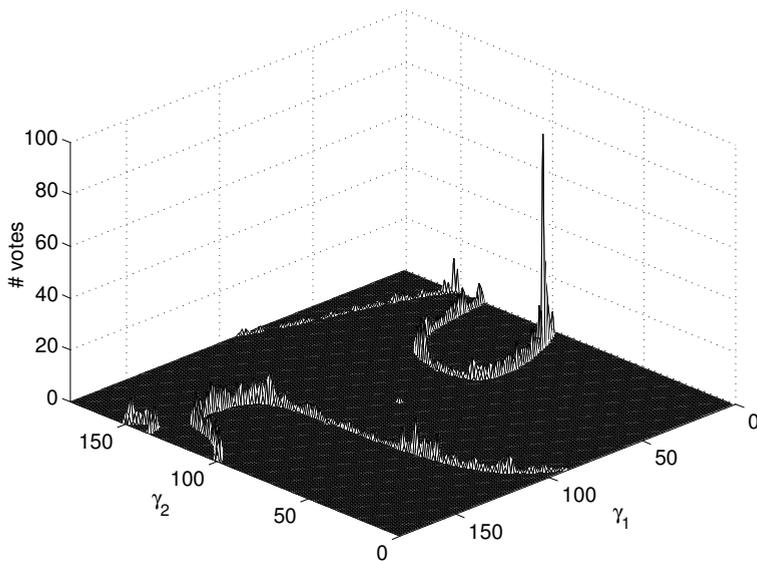


(b) 921 correspondences

Figure 5.4: Fundamental matrix estimation for “Wadham college” pair. Votes for a sample containing outliers.



(a) 100 randomly selected correspondences



(b) 921 correspondences

Figure 5.5: Fundamental matrix estimation for “Wadham college” pair. Votes for an all-inlier sample.

fundamental matrix is evaluated, the coordinates are transformed back again to their original values.

The whole sequence of steps in the estimation process is listed in Fig. 5.6.

5.4 Implementation details

The implementations of the algorithms closely follow the descriptions in Fig. 5.3 and 5.6. The singularity constraint for the fundamental matrix in (5.10), however, requires special care. This equation represents a cubic polynomial in α and can be solved using the explicit solution for a cubic [92]. Though, to efficiently compute the coefficients of the polynomial, we would like to put most operations outside of the data loop in Fig. 5.6. We therefore split the coefficients in polynomials of r and g , which gives

$$\boxed{1} \cdot \alpha^3 + \boxed{2} \cdot \alpha^2 + \boxed{3} \cdot \alpha + \boxed{4} \cdot 1 = 0$$

$$\begin{aligned} \boxed{1} &= \dots \cdot r^3 + \dots \cdot r^2 + \dots \cdot r + \dots \cdot 1 \\ \boxed{2} &= \dots \cdot gr^2 + \dots \cdot r^2 + \dots \cdot gr + \dots \cdot r + \dots \cdot g + \dots \cdot 1 \\ \boxed{3} &= \dots \cdot rg^2 + \dots \cdot g^2 + \dots \cdot rg + \dots \cdot g + \dots \cdot r + \dots \cdot 1 \\ \boxed{4} &= \dots \cdot g^3 + \dots \cdot g^2 + \dots \cdot g + \dots \cdot 1 \end{aligned} \quad (5.14)$$

The coefficients for the polynomials in r and g are functions of the elements of F_1 , F_2 and F_3 in (5.10). If we let $A = F_1 - F_3$ and $B = F_2 - F_3$, then the coefficients of the powers in $\boxed{1}$ are

$$\boxed{5} \cdot r^3 + \boxed{6} \cdot r^2 + \boxed{7} \cdot r + \boxed{8} \cdot 1$$

$$\begin{aligned} \boxed{5} &= \det(B) \\ \boxed{6} &= A_{11}C_{11}^B + A_{12}C_{12}^B + A_{13}C_{13}^B + B_{11}(C_{11}^{A+B} - C_{11}^A - C_{11}^B) \\ &\quad + B_{12}(C_{12}^{A+B} - C_{12}^A - C_{12}^B) + B_{13}(C_{13}^{A+B} - C_{13}^A - C_{13}^B) \\ \boxed{7} &= B_{11}C_{11}^A + B_{12}C_{12}^A + B_{13}C_{13}^A + A_{11}(C_{11}^{A+B} - C_{11}^A - C_{11}^B) \\ &\quad + A_{12}(C_{12}^{A+B} - C_{12}^A - C_{12}^B) + A_{13}(C_{13}^{A+B} - C_{13}^A - C_{13}^B) \\ \boxed{8} &= \det(A) \end{aligned} \quad (5.15)$$

Here A_{ij} represents the element in matrix A at row i and column j , and C_{ij}^A is the (i, j) -cofactor of matrix A . The coefficients in $\boxed{2}$, $\boxed{3}$ and $\boxed{4}$ can be derived in a similar manner. It is clear that the values in (5.15) are independent of individual

- $j = 1, \quad J = \infty, \quad S_{max} = \emptyset$
- Normalize the image correspondences.
- **while** $j < J$ **do**
 - Randomly select 6 correspondences $\{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \dots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$ and use them to compute their nullspace $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3\}$ by solving (5.7).
 - Determine the orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ for the space spanned by the normals in $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3\}$.
 - **if** $n > 100$ **then**
 - form the set C by randomly selecting 100 points from $\{\mathbf{x}_1 \leftrightarrow \mathbf{x}'_1, \dots, \mathbf{x}_n \leftrightarrow \mathbf{x}'_n\} \setminus \{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \dots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$
 - **else**
 - form $C = \{\mathbf{x}_1 \leftrightarrow \mathbf{x}'_1, \dots, \mathbf{x}_n \leftrightarrow \mathbf{x}'_n\} \setminus \{\tilde{\mathbf{x}}_1 \leftrightarrow \tilde{\mathbf{x}}'_1, \dots, \tilde{\mathbf{x}}_6 \leftrightarrow \tilde{\mathbf{x}}'_6\}$
 - **for each** $\mathbf{x} \leftrightarrow \mathbf{x}'$ **in** C **do**
 - Find the possible solutions for $\mathbf{x} \leftrightarrow \mathbf{x}'$ by solving (5.9) and (5.10).
 - Determine γ_1 and γ_2 according to (5.11) for each solution, and round the angles to the nearest degree.
 - Add one vote for each pair of angles (γ_1, γ_2) in the voting array.
 - Determine the pair (γ_1^*, γ_2^*) with the maximum number of votes.
 - Construct \mathbf{f} from (5.12) and (5.13) for γ_1^* and γ_2^* .
 - Find the closest approximation $\hat{\mathbf{f}}$ to \mathbf{f} with $\det(\hat{F}) = 0$ using the SVD.
 - Determine the set of support points S_j for the denormalized $\hat{\mathbf{f}}$, by verifying which points are within distance T .
 - **if** $|S_j| > |S_{max}|$ **then**
 - $J = \log(1 - p) \cdot \log^{-1} \left(1 - \left(\frac{|S_j|}{n} \right)^6 \right)$
 - $S_{max} = S_j$
 - $j = j + 1$
- Re-estimate the fundamental matrix based on the largest support set S_{max} .

Figure 5.6: The RANSAC-Hough algorithm for fundamental matrix estimation using a two-dimensional voting space.

correspondences, and can be computed at the moment that the nullspace of the sample is found in (5.7). This makes that the coefficients for the polynomial in α are quickly found by multiplying the precomputed values with powers of r and g .

5.5 Experimental results

We will compare the proposed RANSAC-Hough method with the standard RANSAC algorithm for plane fitting and fundamental matrix estimation. For all experiments we report the averages and standard deviations over a number of runs of both the executed number of iterations J and the size of the maximum support set $|S_{max}|$. Furthermore, the average running time for a single run is listed. The final re-estimation step in RANSAC will be omitted. The algorithms were implemented in C and ran on Intel Xeon 3.07 GHz / 3.2 GHz computers.

5.5.1 Plane fitting

Synthetic data was generated in \mathbb{R}^3 for the estimation of planes. The inliers are uniformly distributed along the plane and the outliers uniformly distributed in the space. Each point coordinate is bounded to the interval $[0, 100]$. We have added Gaussian unit variance noise to the coordinates of the inliers. The orthogonal distance threshold is set to $T = 1.96$, which makes sure that 95 % of the inliers are within T for the true hyperplane. The plane parameters are chosen randomly for each run.

The results of this experiment over 500 runs are shown in Table 5.1 for a total of 100 data points and different outlier ratios ε . It can be seen that the number of iterations in RANSAC-Hough is clearly lower than in RANSAC. Also, the variation over several runs is smaller. However, a single RANSAC iteration takes here only little time to complete, and an improvement in running time for RANSAC-Hough is therefore only visible for $\varepsilon \geq 0.7$. An iteration of the voting process taking into account 100 points is quite complex when compared to calculating a plane from three points and evaluating its support. Though, the cases $\varepsilon \geq 0.7$ are the most interesting because the absolute running time is here the largest.

Considering the quality of the planes that are found, an increase in the number of support points can be noticed while the standard deviation of this number is smaller. It indicates that the RANSAC-Hough method produces more accurate planes by using a larger part of the data, instead of only an additional third sample point.

Table 5.2 shows the same experiment but now with 300 data points. In this case 100 randomly selected points are used in the voting process. Here the increase in speed can already be seen for $\varepsilon = 0.4$, and the difference grows larger for higher ε . Although the values of J_R and J_{RH} are comparable to those in the 100 point experiment, an iteration for 300 points takes more time and causes a larger difference

Table 5.1: The results for plane fitting using RANSAC (R) and RANSAC-Hough (RH) in synthetic data with 100 points. Indicated are the averages and standard deviations (\pm) for the executed number of iterations J and the maximum number of support points $|S_{max}|$, and the average running time t in milliseconds.

ϵ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
J_R	5.0 ± 1.5	7.6 ± 2.1	11.5 ± 3.6	17.9 ± 5.0	26.6 ± 6.8	43.6 ± 10.6	83.3 ± 21.4	179 ± 53.6	478 ± 154	$1,224 \pm 278$
J_{RH}	3.1 ± 0.7	4.7 ± 0.8	6.5 ± 0.9	8.8 ± 1.1	11.9 ± 1.5	17.6 ± 2.1	27.2 ± 3.7	45.1 ± 7.0	90.5 ± 18.0	169 ± 24.9
$ S_{max} _R$	88.9 ± 5.0	80.6 ± 4.8	71.9 ± 4.6	63.2 ± 4.2	55.9 ± 3.6	47.8 ± 3.1	38.8 ± 2.8	30.2 ± 2.7	21.9 ± 2.3	15.8 ± 1.4
$ S_{max} _{RH}$	91.7 ± 3.5	82.8 ± 3.4	74.3 ± 3.1	66.1 ± 2.8	58.1 ± 2.6	49.0 ± 2.4	40.2 ± 2.4	31.7 ± 2.3	22.7 ± 2.2	16.6 ± 1.3
t_R	0.24	0.30	0.44	0.60	0.86	1.36	2.54	5.40	13.78	34.76
t_{RH}	0.60	0.80	0.98	1.24	1.56	2.16	3.20	5.14	10.08	18.46

Table 5.2: The results for plane fitting in synthetic data with 300 points.

ϵ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
J_R	5.1 ± 1.8	7.4 ± 2.0	11.3 ± 2.7	17.2 ± 4.5	27.2 ± 6.2	46.5 ± 11.5	87.2 ± 20.5	182 ± 43.0	485 ± 122	$2,142 \pm 608$
J_{RH}	3.2 ± 0.6	4.6 ± 0.7	6.3 ± 0.7	8.8 ± 1.1	12.3 ± 1.4	18.2 ± 2.0	28.0 ± 3.2	47.1 ± 5.3	94.9 ± 13.7	263 ± 54.0
$ S_{max} _R$	266 ± 15.0	241 ± 13.1	216 ± 11.4	191 ± 10.8	165 ± 9.4	140 ± 8.6	114 ± 6.9	89.9 ± 5.8	64.8 ± 4.7	39.6 ± 3.7
$ S_{max} _{RH}$	275 ± 9.8	249 ± 8.8	224 ± 7.5	198 ± 6.9	172 ± 6.4	145 ± 6.3	119 ± 5.3	92.7 ± 4.5	66.2 ± 4.3	40.3 ± 4.0
t_R	0.72	0.84	1.14	1.48	2.32	3.90	6.66	13.66	35.20	141.76
t_{RH}	1.06	1.38	1.52	1.74	2.30	3.18	4.60	8.12	14.86	49.18

in running time. To illustrate this point, we have plotted the ratio of t_R and t_{RH} in Fig. 5.7 for both experiments.

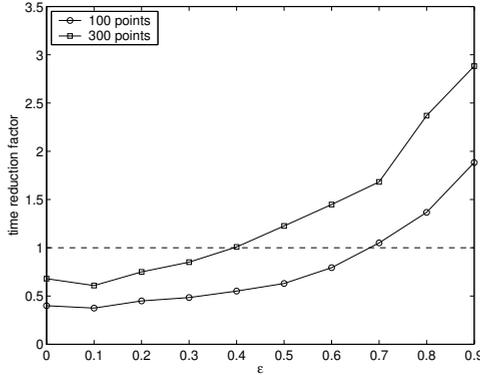


Figure 5.7: The ratio $\frac{t_R}{t_{RH}}$ for the data from Table 5.1 and 5.2.

Next, we consider the fitting of planes in range image data. Like in Chapter 4, we have used 10 images from the ABW structured light scanner for which one of the images is shown in Fig. 5.8. We have subsampled the images with a factor 2 to obtain 256×256 sized images. We search with RANSAC for planes in the images, and subsequently delete the points from the data set which belong to a plane. The repeated application of RANSAC is stopped when a plane is returned with support smaller than 500 pixels. For the shown example image, the number of planes extracted this way will be about 12. The experiment is repeated 500 times for each image. The threshold for the orthogonal distance to the plane is set to $T = 2.5$, which is large enough to capture noisy variations of the inliers. Table 5.3 shows the results of the experiments.

The RANSAC-Hough method outperforms RANSAC in all aspects; in some cases it is up to a factor 5 faster. The total number of points on the extracted planes is comparable, while the number of planes is a bit smaller. This means that the extracted planes are actually better fits, since they contain a larger part of the data.

5.5.2 Fundamental matrix estimation

The synthetic data is produced by randomly generating points in a 3D cube. The cameras are randomly positioned around the cube on a sphere, which has a radius of two times the cube size. The points are between 20 and 60 focal lengths away from the cameras.

Table 5.3: The results for finding all planes using RANSAC (R) and RANSAC-Hough (RH) in the ABW range images. Indicated are the averages and standard deviations (\pm) for the total number of iterations $\sum J$ per image, the running time t in seconds, the total size of the maximum support sets $\sum |S_{max}|$ and the number of planes $\#h$ found.

	$\sum J_R \cdot (10^3)$	$\sum J_{RH} \cdot (10^3)$	t_R	t_{RH}	$\sum S_{max} _R \cdot (10^4)$	$\sum S_{max} _{RH} \cdot (10^4)$	$\#h_R$	$\#h_{RH}$
image 0	9.50 ± 1.42	1.63 ± 0.17	19.9 ± 2.37	3.84 ± 0.31	6.29 ± 0.032	6.29 ± 0.033	14.0 ± 0.7	13.8 ± 0.6
image 1	3.41 ± 0.48	0.72 ± 0.09	5.47 ± 0.64	1.39 ± 0.13	6.37 ± 0.029	6.37 ± 0.028	10.8 ± 0.6	10.4 ± 0.6
image 2	5.36 ± 0.75	1.05 ± 0.11	9.58 ± 1.53	2.06 ± 0.24	6.41 ± 0.034	6.41 ± 0.034	13.0 ± 0.7	12.7 ± 0.6
image 3	2.12 ± 0.37	0.51 ± 0.07	4.13 ± 0.70	1.13 ± 0.14	6.41 ± 0.028	6.41 ± 0.026	10.0 ± 0.6	9.7 ± 0.6
image 4	2.88 ± 0.50	0.63 ± 0.08	5.71 ± 0.73	1.45 ± 0.14	6.41 ± 0.029	6.40 ± 0.028	11.1 ± 0.7	10.6 ± 0.6
image 5	2.55 ± 0.34	0.58 ± 0.07	4.32 ± 0.57	1.14 ± 0.13	6.39 ± 0.029	6.39 ± 0.030	10.0 ± 0.6	9.7 ± 0.6
image 6	5.03 ± 0.68	1.00 ± 0.11	9.80 ± 1.18	2.18 ± 0.20	6.36 ± 0.029	6.36 ± 0.031	12.3 ± 0.6	12.0 ± 0.6
image 7	1.62 ± 0.29	0.41 ± 0.05	3.74 ± 0.39	1.07 ± 0.08	6.44 ± 0.026	6.44 ± 0.024	9.7 ± 0.6	9.3 ± 0.5
image 8	2.00 ± 0.36	0.47 ± 0.07	3.61 ± 0.57	1.00 ± 0.11	6.40 ± 0.028	6.40 ± 0.028	9.2 ± 0.6	8.9 ± 0.6
image 9	1.91 ± 0.37	0.46 ± 0.07	4.09 ± 0.52	1.14 ± 0.10	6.44 ± 0.029	6.44 ± 0.029	10.2 ± 0.7	9.7 ± 0.7

Table 5.4: Fundamental matrix estimation using RANSAC (R) and RANSAC-Hough (RH) in synthetically generated data for different outlier ratios ϵ . Indicated are the averages and standard deviations (\pm) for the executed number of iterations J , the maximum number of support points $|S_{max}|$ and the running time t in seconds.

ϵ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
J_R	99.0 ± 38.0	206 ± 80.4	451 ± 185	$1,133 \pm 445$	$2,923 \pm 1,215$	$9,714 \pm 3,992$	$4.08 \pm 1.79 \cdot 10^4$	$2.52 \pm 1.22 \cdot 10^5$
J_{RH}	66.7 ± 23.4	126 ± 44.4	255 ± 89.9	537 ± 211	$1,278 \pm 495$	$3,881 \pm 1,656$	$1.35 \pm 0.61 \cdot 10^4$	$0.66 \pm 0.34 \cdot 10^5$
$ S_{max} _R$	132 ± 7.6	119 ± 6.9	106 ± 6.3	93.8 ± 5.5	82.0 ± 5.3	69.1 ± 4.5	56.3 ± 3.9	43.6 ± 3.2
$ S_{max} _{RH}$	131 ± 7.8	119 ± 7.5	106 ± 6.6	93.4 ± 6.4	80.9 ± 5.3	67.4 ± 4.9	54.8 ± 4.1	42.5 ± 3.5
t_R	0.015 ± 0.008	0.033 ± 0.014	0.077 ± 0.033	0.20 ± 0.079	0.52 ± 0.22	1.73 ± 0.71	7.27 ± 3.18	45.0 ± 21.9
t_{RH}	0.032 ± 0.014	0.064 ± 0.027	0.134 ± 0.055	0.29 ± 0.143	0.68 ± 0.30	2.07 ± 0.99	7.19 ± 3.68	35.1 ± 19.3

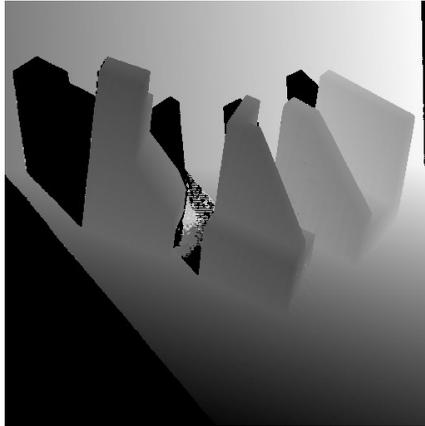


Figure 5.8: A range image used for plane fitting.

For the inliers we add Gaussian distributed noise to both coordinates with standard deviation of about 0.3% of the image size. For the outliers we select two different space points, and use their non-corresponding projections as data points. The distance measure used is the Sampson distance (2.27), which is compared to the squared threshold T^2 . The threshold has been set to $T^2 = 3.84\sigma^2$ where σ^2 is the measured variance of the error with respect to the true fundamental matrix. The variance is averaged over several runs with each run having different point and camera positions.

Since for $\varepsilon \geq 0.8$ the number of iterations for standard RANSAC grows very large ($> 10^6$) on occasion and we would like to record the running time, only outlier ratios up to 0.7 will be considered. In case the number of iterations exceeded 10^6 for $\varepsilon \leq 0.7$, it was clipped at 10^6 .

The results over 500 runs are shown in Table 5.4 for 200 correspondences. The number of iterations is reduced when using RANSAC-Hough and for $\varepsilon = 0.6$ and $\varepsilon = 0.7$ the running time is shorter. Considering the support sets, we can see that they are on average slightly smaller. This is a result of the rank 2 enforcement which finds an approximation of the fundamental matrix computed from the data. Since the data is not considered in finding this approximation, some inliers are lost in the process.

The real images we have used for testing are shown in Appendix B. Again the the Sampson distance is chosen as error measure, and we have set the square root of the threshold to $T = 1.5$ pixels.

The results of running RANSAC 500 times on the image pairs are shown in Table 5.5. Just as in case of synthetic data, the difference in running times is

best noticeable for higher outlier ratios. The number of iterations is here reduced considerably and the additional complexity of the voting process does not prohibit a speedup anymore. The support sets found are slightly smaller than those for RANSAC, due to the same reason as for synthetic data. In addition, we have performed some experiments where the votes were not cast in a single bin but in a small region of bins. This approach can in general provide a more accurate position of the maximum in the Hough space, but it did not yield noticeable better results in our case. The strict one-dimensional voting pattern in the array (see Fig. 5.4 and 5.5) assures that neighboring bin positions do not have to be considered.

For image pair “Valbonne church” the 6-point algorithm runs slower than the standard 7-point algorithm. At the same time, the image pair “Univ. British Columbia” with a comparable outlier ratio shows the opposite. This can be explained by the fact that the latter image pair has in total 911 correspondences and the former 299. The standard RANSAC algorithm therefore takes less time to complete for the “Valbonne church” pair.

5.6 Discussion

The combination of RANSAC and the Hough transform, that has been advocated in the past, is made applicable to hyperplanes and the fundamental matrix by a new parameterization of the model. For hyperplanes, the result is an efficient one-dimensional voting space and a reduction of the sample size by one point. For the fundamental matrix, a two-dimensional voting space is applied because of the singularity constraint. Instead of sampling 7 correspondences per model, we now only need to take 6-point samples. This makes it much easier to find an all-inlier sample by random trials. In addition, we use for both models randomly selected subsets of the data to speed up the voting stage.

Experimental results were shown on plane fitting in synthetic data and in range data. The synthetic data showed faster estimation for high outlier ratios or a large number of data points. Also the consecutive extraction of planes in range images took considerably less time using the RANSAC-Hough method. The quality of the solutions is either equal or better than standard RANSAC. The fundamental matrix was estimated using synthetic and real image data. The results indicated that for high outlier ratios a much faster estimation is achieved, with only a minor decrease in the size of the support.

A further improvement of the algorithm may be the finding of a better approximation to the fundamental matrix. In particular, the loss of support points is caused by the enforcement of the singularity constraint, since an approximation to the model is found independently of the data. Other methods to enforce singularity may be tried to retain the same level of support, possibly at the cost of a higher complexity.

Since the number of RANSAC iterations increases almost exponentially with the

Table 5.5: Fundamental matrix estimation using RANSAC (R) and RANSAC-Hough (RH) on real image pairs. Indicated are the averages and standard deviations (\pm) for the executed number of iterations J , the maximum number of support points $|S_{max}|$ and the running time t in seconds.

image pair	ε	# inliers	J_R	J_{RH}	$ S_{max} _R$	$ S_{max} _{RH}$	t_R	t_{RH}
books	0.74	189	$7.2 \pm 0.64 \cdot 10^4$	$1.92 \pm 0.17 \cdot 10^4$	187 ± 2.5	185 ± 2.9	25.1 ± 3.6	11.7 ± 1.9
pile of books	0.82	97	$4.03 \pm 0.71 \cdot 10^5$	$0.93 \pm 0.18 \cdot 10^5$	109 ± 2.9	106 ± 3.5	114 ± 24.2	52.5 ± 12.0
Wadham college	0.71	264	$7.08 \pm 2.93 \cdot 10^4$	$1.96 \pm 0.73 \cdot 10^4$	241 ± 13.8	236 ± 14.4	29.0 ± 12.4	12.4 ± 4.9
Univ. British Columbia	0.56	399	$2.65 \pm 0.56 \cdot 10^3$	$1.14 \pm 0.25 \cdot 10^3$	372 ± 11.0	369 ± 12.8	1.13 ± 0.28	0.72 ± 0.20
Corridor	0.43	150	466 ± 160	261 ± 95.5	139 ± 5.8	138 ± 6.8	0.08 ± 0.031	0.13 ± 0.055
Valbonne church	0.58	127	$2.56 \pm 0.63 \cdot 10^3$	$1.23 \pm 0.40 \cdot 10^3$	123 ± 3.7	121 ± 5.4	0.49 ± 0.13	0.64 ± 0.23

number of points in the sample, it is tempting to try to reduce the sample size even further. Obviously, the voting stage becomes more complex for smaller sample sizes. The voting stages for hyperplanes and the fundamental matrix have already illustrated that the latter is less straightforward due to the singularity constraint. Several computations have been taken outside the data loop of the algorithm, making it possible to achieve the reported results. It therefore does not seem fruitful to pursue a further reduction in sample size. If such an implementation would be faster, then probably only for very high outlier ratios, while performing worse for moderate values.

Chapter 6

Discussion

In this thesis we have described techniques for local feature extraction and matching, and the subsequent estimation of the epipolar geometry. Several methods have been proposed to expedite the estimation of the epipolar geometry. Below we will discuss possible extensions of the presented work and indicate directions for future research.

6.1 Correspondence search

The extraction of local features from images has been described as a necessary step for finding image correspondences. Interest points from an image pair will likely cover some common world points in the scene, so that comparison of local features computed at the interest points will yield a set of correspondences. These correspondences allow the computation of the epipolar geometry, even when many of them are erroneous.

Unfortunately, in case of large viewpoint changes, feature matching may yield very few inliers due to significant changes of the local image structure. It is then the question whether the set of best matches still forms the most logical choice for the correspondences. When being optimistic about the possibilities for fast fundamental matrix estimation, it is even the question whether feature matching is still necessary. The interest points in the first image can simply be combined with all interest points in the second image. Although the inliers are then hidden in a large amount of data, every possible inlier for the interest points will be included since all point combinations are considered. It will be interesting to see whether such an approach to establishing correspondences can be useful, possibly under the requirement that only the most salient interest points are selected.

6.2 Non-uniform sampling

Two methods have been proposed in this thesis for calculating sampling probabilities for the correspondences. Both methods achieve this goal without regard to the original image content. Yet, the sampling probabilities for correspondences may also be altered by incorporating specific image properties. For example, when a segmentation of the images is available, the local features can be grouped accordingly. A number of correspondences connecting two regions may indicate that all features originating from these regions should form correspondences. The connecting correspondences may then receive high sampling probabilities, while other correspondences from the regions are assigned low probabilities. Furthermore, it is possible to start the sampling in certain regions, and see if computed epipolar geometries extend to the rest of the data.

In a way, the option of segmentation seems to be the opposite of what has been achieved by using local features: the possibility to deal with occlusion and clutter in the images. However, the locality of the features is not at stake when using a segmentation. Only the selection of feature pairs is affected.

6.3 Zero or multiple epipolar geometries

The methods in this thesis have been directed towards finding a single epipolar geometry. Although this is a situation of significant practical interest, the number of epipolar geometries in a scene may be different. When we have two completely unrelated images then no epipolar geometry will be present. As discussed in Chapter 1, this may occur in the visual retrieval scenario. In case several objects are moving independently in a scene, multiple epipolar geometries will be present in images taken at different time instants. The correspondences will then consist of distinct sets of inliers and a number of outliers.

In contrast to the non-uniform sampling methods, the six-point method in Chapter 5 can be applied directly to the abovementioned scenarios. The principle of sampling fewer correspondences will always reduce the required number of iterations. In both scenarios we may expect it to be faster than RANSAC because the method works best for high outlier ratios. In case there is no epipolar geometry we have the maximum outlier ratio, namely 1. For the sequential extraction of multiple epipolar geometries, the initial outlier ratio is also high since it is determined by the inliers of a single epipolar geometry. The remaining correspondences are all outliers with respect to this geometry. The outlier ratio will decrease as more geometries are extracted and have their inliers removed from the data.

An interesting possibility that the quadric algorithm in Chapter 3 may offer is the detection of an epipolar geometry. Inliers are included in the dominant set a large number of times, while the outliers are included only half of the time. There should therefore be two clusters of count values in the set of correspondences. If we

are able to detect the presence of these two clusters, we may affirm the presence of an epipolar geometry. This would be very helpful in the retrieval scenario where we are essentially not interested in the precise geometrical relation between the images, but merely in the fact whether the images are from the same scene or not.

6.4 Algorithm evaluation

Finally, which algorithm in this thesis is to be preferred? Considering the non-uniform sampling methods, the quadric algorithm is perhaps most reliable. On the other hand, weighted sampling can potentially make a better distinction between inliers and outliers, which will eventually find the epipolar geometry faster. The particular feature of weighted sampling is that it offers little control over the algorithm. This may not be necessary in a scenario which meets the assumptions, as the algorithm will generally find the desired model. In other scenarios, like the presence of multiple epipolar geometries or structured outliers, it would be desirable to have more control. Ideally, the algorithm should be able to check whether the weight updates have been of any use up to the current iteration. If not, it may stop producing updates and let RANSAC run unhampered. In this respect, the quadric method is more suitable for a selective use, due to its low complexity and the fact that it is only a preprocessor; the results may be evaluated before running RANSAC. Furthermore, the quadric method is not random, which makes the evaluation easier than for weighted sampling.

The six-point method is more general than the methods based on non-uniform sampling, as it may be used for searching multiple epipolar geometries. A disadvantage of the method is that it performs worse for moderate outlier ratios in combination with few correspondences. However, this may be improved by further optimization of the method. The six-point method therefore belongs to the class of RANSAC variations, which is most likely to provide the next standard for epipolar geometry estimation.

Appendix A

Homogeneous Coordinates

The coordinates of a point $\mathbf{x} = (x, y)^\top$ in 2D can be extended with a scalar s to produce the homogeneous coordinates $\mathbf{x} = (x, y, s)^\top$. Homogeneous coordinates [28] describe the projective space \mathbb{P}^2 as opposed to Euclidean space \mathbb{R}^2 . Coordinates differing by a non-zero scalar value m are identical in \mathbb{P}^2 , i.e. $\mathbf{x} = (x, y, s)^\top$ and $\mathbf{x} = (mx, my, ms)^\top$ represent the same point. It is convenient to select $s = 1$, so that the original values x and y can be retrieved when dividing by the third coordinate. We therefore adopt $\mathbf{x} = (x, y, 1)^\top$.

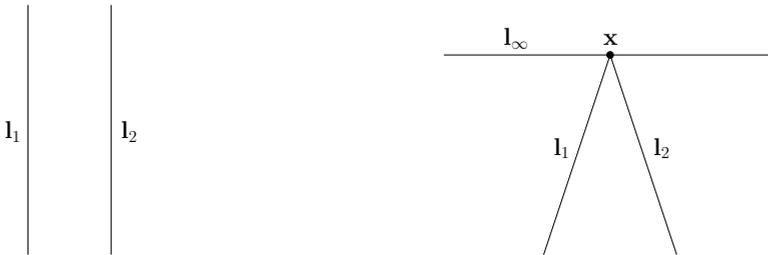


Figure A.1: After projective transformation of the plane containing two parallel lines l_1 and l_2 (left), their intersection point \mathbf{x} on the line at infinity l_∞ becomes visible (right).

The projective space \mathbb{P}^2 has an explicit notion of infinity. For example, two lines always meet in a point, but when the lines are parallel they “meet in infinity”. If we take two parallel lines $l_1 = (a, b, c_1)^\top$ and $l_2 = (a, b, c_2)^\top$ with defining equations $ax + by + c_1 = 0$ and $ax + by + c_2 = 0$, then their point of intersection is $\mathbf{x} = (b, -a, 0)^\top$ since $(b, -a, 0)l_1 = 0$ and $(b, -a, 0)l_2 = 0$. All points with their last coordinate equal

to zero, are said to form the line at infinity $\mathbf{l}_\infty = (0, 0, 1)^\top$. The line \mathbf{l}_∞ can be made visible by a projective transformation of the space, as shown in Fig. A.1. Instead of looking at two parallel lines representing for instance the top view of a road, we will then see a road disappearing at the horizon which is represented by \mathbf{l}_∞ . Note that the representation of lines is also homogeneous, since $m\mathbf{l}$ and \mathbf{l} will contain the same points.

Higher dimensional spaces can be extended in the same way to form projective spaces. For example, points in a 3D space can be denoted as $\mathbf{X} = (X, Y, Z, 1)^\top$. Points whose fourth coordinate is zero then belong to the plane at infinity $\boldsymbol{\pi}_\infty = (0, 0, 0, 1)^\top$.

Appendix B

Two-View Data Sets

Several real image pairs have been used for experimentation in this thesis, and they are shown in Fig. B.1 to B.8. There are differences in viewpoint and/or zoom factor between the left and right images. The image pairs in Fig. B.1 to B.3 are obtained from the INRIA institutes Rhône-Alpes and Sophia-Antipolis (FR), and those in Fig. B.4 and B.5 from the Visual Geometry Group in Oxford (UK). Pair B.6 is an object from the Amsterdam Library of Object Images [19], and pairs B.7 and B.8 are made by the author himself.

The SIFT keypoint detector¹ [51] has been applied for establishing correspondences between the image pairs. As indicated in Chapter 2, the SIFT detector is one of the most reliable methods for finding interest points and deriving local features. For every feature in the left image of the pairs, we found the nearest feature in the right image using Euclidean distance. The correspondence is retained if the distance is smaller than 0.8 times the distance to the second-nearest feature [51]. The inliers among the resulting correspondences are found by robustly estimating the fundamental matrix. We manually identified any remaining incorrect correspondences among the inliers and labeled them as outliers. The left images in Fig. B.1 to B.8 show the final sets of inlying feature points, and the right images the outlying feature points. Also indicated for every image pair are the total number of correspondences n and the outlier ratio ε .

¹The code is obtained from <http://www.cs.ubc.ca/~lowe/keypoints/>.



Figure B.1: Valbonne church: $n = 299$ and $\varepsilon = 0.58$.



Figure B.2: Boat: $n = 358$ and $\varepsilon = 0.49$.



Figure B.3: University of British Columbia: $n = 911$ and $\varepsilon = 0.56$.

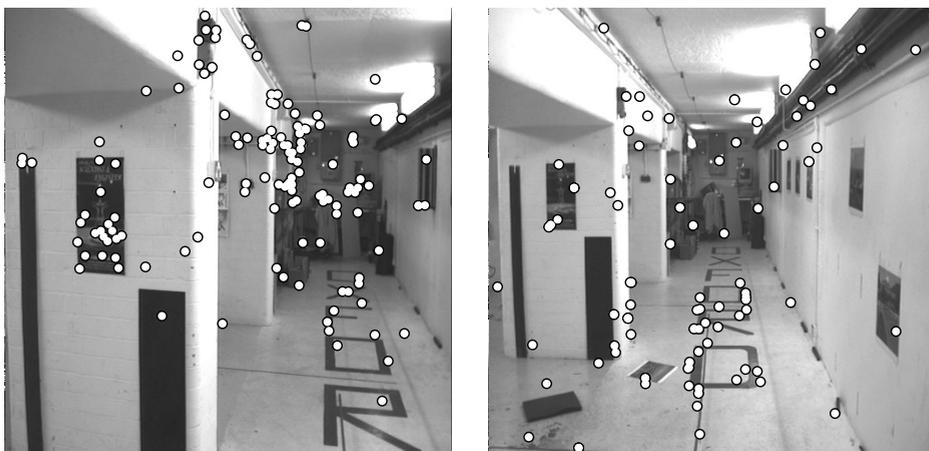


Figure B.4: Corridor: $n = 262$ and $\varepsilon = 0.43$.

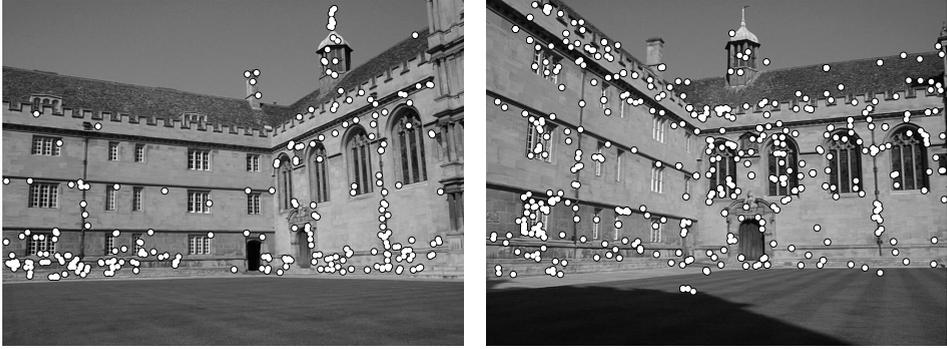


Figure B.5: Wadham college: $n = 921$ and $\varepsilon = 0.71$.

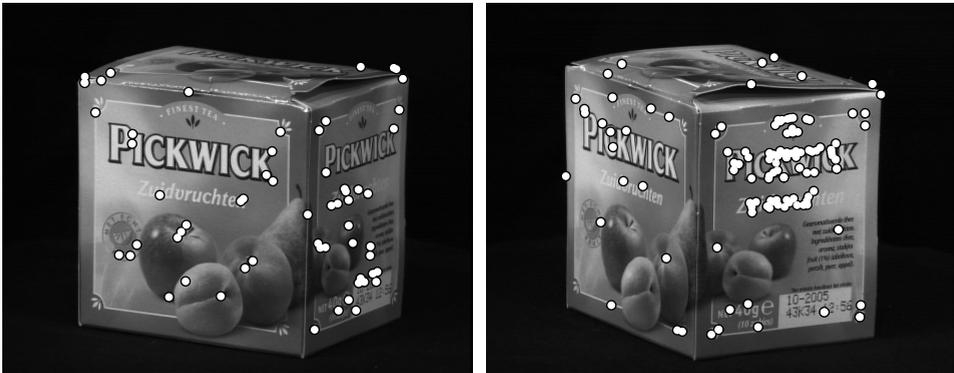


Figure B.6: Tea box: $n = 221$ and $\varepsilon = 0.71$.

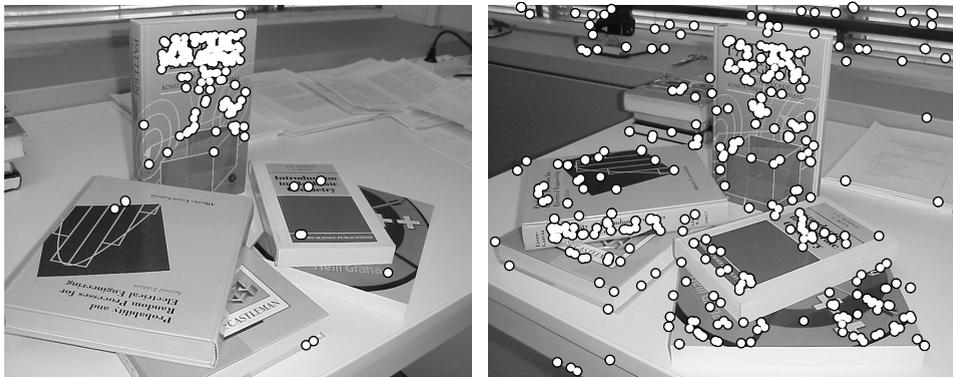


Figure B.7: Books: $n = 740$ and $\varepsilon = 0.74$.

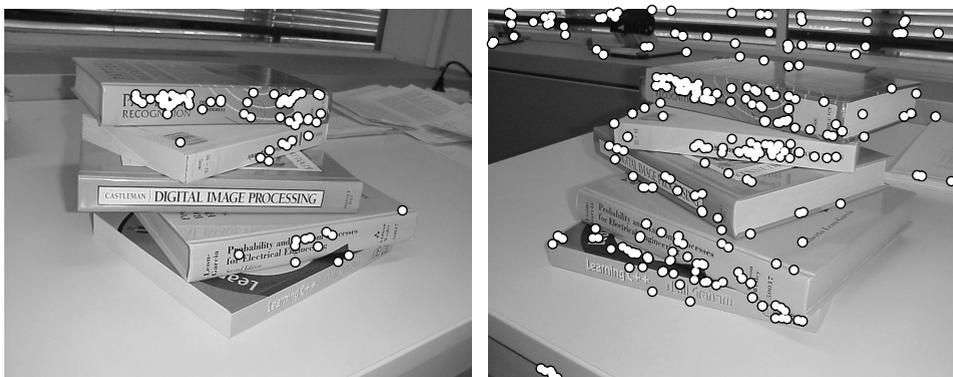


Figure B.8: Pile of books: $n = 548$ and $\varepsilon = 0.82$.

Appendix C

List of Symbols

n	number of elements in data set
ε	outlier ratio of data set
j	iteration counter
J	number of iterations
S	support set which contains all data points close to a model
$ S $	size of the support set
S_{max}	largest support set found
k	number of inliers in the support set
q	number of outliers in the support set
T	threshold for determining the support set
H^T	subspace within distance T from the model
d	number of points in a sample
d^{in}	number of inliers in a sample
$I(x, y)$	image intensity at pixel coordinates (x, y)
\mathbf{x}	data or image point
\dots'	quantity belonging to the second image
$\mathbf{x} \leftrightarrow \mathbf{x}'$	corresponding pair of image points
$\tilde{\mathbf{x}}$	data or image point included in a sample
\mathbf{X}	3D world point
\mathbf{C}	camera center
P	camera projection matrix
F	fundamental matrix
\mathbf{l}	line in the image plane
\mathbf{n}	normal vector of hyperplane
b	offset of hyperplane
t	running time of algorithm

Bibliography

- [1] A. Adam, E. Rivlin, and I. Shimshoni. ROR: Rejection of outliers by rotations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):78–84, 2001.
- [2] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [3] A. Baumberg. Reliable feature matching across widely separated views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 774–781, 2000.
- [4] H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 329–336, 2005.
- [5] P.A. Beardsley, P.H.S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Proc. European Conference on Computer Vision*, pages 683–695, 1996.
- [6] P.A. Beardsley, A. Zisserman, and D.W. Murray. Navigation using affine structure from motion. In *Proc. European Conference on Computer Vision*, pages 85–96, 1994.
- [7] A. Del Bimbo. *Visual Information Retrieval*. Morgan Kaufmann Publishers, 1999.
- [8] M. Brown and D.G. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *International Conference on 3-D Digital Imaging and Modeling*, 2005.
- [9] G. Carneiro and A.D. Jepson. Flexible spatial models for grouping local image features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 747–754, 2004.

-
- [10] C.-S. Chen, Y.-P. Hung, and J.-B. Cheng. RANSAC-Based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1229–1234, 1999.
- [11] O. Chum and J. Matas. Randomized ransac with $T_{d,d}$ test. In *Proc. British Machine Vision Conference*, 2002.
- [12] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *Proc. DAGM*. Springer-Verlag, 2003.
- [13] O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 772–779, 2005.
- [14] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. Wiley, 1973.
- [15] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc. European Conference on Computer Vision*, pages 563–578, 1992.
- [16] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [17] A.W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. European Conference on Computer Vision*, pages 311–326, 1998.
- [18] L.M.J. Florack, B.M. ter Haar Romeny, J.J. Koenderink, and M.A. Viergever. General intensity transformations and differential invariants. *Journal of Mathematical Imaging and Vision*, 4(2):171–187, 1994.
- [19] J.M. Geusebroek, G.J. Burghouts, and A.W.M. Smeulders. The Amsterdam library of object images. *Int. Journal of Computer Vision*, 61(1):103–112, 2005.
- [20] T. Goedemé, T. Tuytelaars, and L. Van Gool. Fast wide baseline matching for visual navigation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 24–29, 2004.
- [21] A. Gupta and R. Jain. Visual information retrieval. *Communications of the ACM*, 40(5):70–79, 1997.
- [22] A. Hanjalić. *Content-Based Analysis of Digital Video*. Kluwer Academic Publishers, 2004.

-
- [23] C.J. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference*, pages 147–151, 1988.
- [24] R.I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *Proc. European Conference on Computer Vision*, pages 579–587, 1992.
- [25] R.I. Hartley. Projective reconstruction and invariants from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1036–1041, 1994.
- [26] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.
- [27] R.I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–764, 1992.
- [28] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003.
- [29] M. Heikkilä and M. Pietikäinen. An image mosaicing module for wide-area surveillance. In *Proc. ACM Int. Workshop on Video Surveillance & Sensor Networks*, pages 11–18, 2005.
- [30] D.C. Hoaglin, F. Mosteller, and J.W. Tukey, editors. *Understanding Robust and Exploratory Data Analysis*. Wiley, 1983.
- [31] R.J.M. den Hollander and A. Hanjalić. Evaluation of a string matching technique for logo recognition in video stills. In *Proc. Annual Conference of the Advanced School for Computing and Imaging*, pages 252–257, 2003.
- [32] R.J.M. den Hollander and A. Hanjalić. Logo recognition in video stills by string matching. In *Proc. IEEE International Conference on Image Processing*, pages 517–520, 2003.
- [33] R.J.M. den Hollander and A. Hanjalić. Faster RANSAC by inlier probability adjustment. In *Proc. Annual Conference of the Advanced School for Computing and Imaging*, pages 69–74, 2004.
- [34] R.J.M. den Hollander and A. Hanjalić. Logo recognition in video by line profile classification. In *Proc. IS&T/SPIE Storage and Retrieval Methods and Applications for Multimedia*, pages 300–306, 2004.
- [35] R.J.M. den Hollander and A. Hanjalić. Outlier identification in stereo correspondences using quadrics. In *Proc. British Machine Vision Conference*, 2005.

-
- [36] R.J.M. den Hollander and A. Hanjalić. Outlier identification in stereo matches. In *Proc. Annual Conference of the Advanced School for Computing and Imaging*, pages 130–136, 2005.
- [37] R.J.M. den Hollander and A. Hanjalić. A six-point RANSAC algorithm for fundamental matrix estimation. Technical Report ICT-2007-02, Delft University of Technology, 2007. Submitted to *Computer Vision and Image Understanding*.
- [38] R.J.M. den Hollander, A. Hanjalić, and R.L. Lagendijk. Weighted data sampling for faster RANSAC. Technical Report ICT-2007-01, Delft University of Technology, 2007.
- [39] P.V.C. Hough. Methods and means for recognizing complex patterns. U.S. Patent 3069654, Dec. 1962.
- [40] M. Isard and A. Blake. CONDENSATION—Conditional density propagation for visual tracking. *Int. Journal of Computer Vision*, 29(1):5–28, 1998.
- [41] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 506–513, 2004.
- [42] N. Kiryati, Y. Eldar, and A.M. Bruckstein. A probabilistic Hough transform. *Pattern Recognition*, 24(4):303–316, 1991.
- [43] H. Kwakernaak and R. Sivan. *Modern Signals and Systems*. Prentice-Hall, 1991.
- [44] K.-M. Lee, P. Meer, and R.-H. Park. Robust adaptive segmentation of range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):200–205, 1998.
- [45] A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78:99–118, 2000.
- [46] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.
- [47] T. Lindeberg. Feature detection with automatic scale selection. *Int. Journal of Computer Vision*, 30(2):79–116, 1998.
- [48] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. *Image and Vision Computing*, 15(6):415–434, 1997.
- [49] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

-
- [50] D.G. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE International Conference on Computer Vision*, pages 1150–1157, 1999.
- [51] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.
- [52] J. Matas and O. Chum. Randomized RANSAC with sequential probability ratio test. In *Proc. IEEE International Conference on Computer Vision*, pages 1727–1732, 2005.
- [53] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. British Machine Vision Conference*, pages 384–393, 2002.
- [54] J. McBride, M. Snorrason, T. Goodsell, R. Eaton, and M.R. Stevens. Single camera stereo for mobile robot surveillance. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
- [55] P.F. McLauchlan and A. Jaenicke. Image mosaicing using sequential bundle adjustment. In *Proc. British Machine Vision Conference*, 2000.
- [56] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. IEEE International Conference on Computer Vision*, pages 525–531, 2001.
- [57] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. European Conference on Computer Vision*, pages 128–142, 2002.
- [58] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.
- [59] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schafalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. Journal of Computer Vision*, 65(1-2):43–72, 2005.
- [60] D.R. Myatt, P.H.S. Torr, S.J. Nasuto, J.M. Bishop, and R. Craddock. Napsac: high noise, high dimensional robust estimation - it's in the bag. In *Proc. British Machine Vision Conference*, 2002.
- [61] W. Niblack and D. Petkovic. On improving the accuracy of the Hough transform: theory, simulations, and experiments. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 574–579, 1988.
- [62] D. Nistér. Preemptive RANSAC for live structure and motion estimation. In *Proc. IEEE International Conference on Computer Vision*, pages 199–206, 2003.

-
- [63] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.
- [64] S. Obdržálek and J. Matas. Object recognition using local affine frames on distinguished regions. In *Proc. British Machine Vision Conference*, pages 113–122, 2002.
- [65] C.F. Olson. Constrained Hough transforms for curve detection. *Computer Vision and Image Understanding*, 73(3), 1999.
- [66] C.F. Olson. A general method for geometric feature matching and model extraction. *Int. Journal of Computer Vision*, 45(1), 2001.
- [67] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Hand-held acquisition of 3D models with a video camera. In *Proc. International Conference on 3-D Imaging and Modeling*, pages 14–23, 1999.
- [68] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition, 1992.
- [69] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *Proc. IEEE International Conference on Computer Vision*, pages 754–760, 1998.
- [70] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *Int. Journal of Computer Vision*, pages 231–259, 2006.
- [71] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. Wiley, New York, 1987.
- [72] F. Schaffalitzky and A. Zisserman. Automated location matching in movies. *Computer Vision and Image Understanding*, 92:236–264, 2003.
- [73] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–535, 1997.
- [74] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *Int. Journal of Computer Vision*, 37(2):151–172, 2000.
- [75] Y. Shan, B. Matei, H.S. Sawhney, R. Kumar, D. Huber, and M. Hebert. Linear model hashing and batch RANSAC for rapid and accurate object recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 121–128, 2004.

-
- [76] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.
- [77] C.V. Stewart. MINPRAN: a new robust estimator for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):925–938, 1995.
- [78] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proc. European Conference on Computer Vision*, pages 709–720, 1996.
- [79] C.-K. Tang, G. Medioni, and M.-S. Lee. N-dimensional tensor voting and application to epipolar geometry estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):829–844, 2001.
- [80] D. Tell and S. Carlsson. Wide baseline point matching using affine invariants computed from intensity profiles. In *Proc. European Conference on Computer Vision*, pages 814–828, 2000.
- [81] D. Tell and S. Carlsson. Combining appearance and topology for wide baseline matching. In *Proc. European Conference on Computer Vision*, pages 68–81, 2002.
- [82] D.W. Thompson and J.L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *Proc. ICRA*, pages 208–220, 1987.
- [83] W.-S. Tong, C.-K. Tang, and G. Medioni. Simultaneous two-view epipolar geometry estimation and motion segmentation by 4D tensor voting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.
- [84] B. Tordoff and D.W. Murray. Guided sampling and consensus for motion estimation. In *Proc. European Conference on Computer Vision*, 2002.
- [85] P.H.S. Torr, A.W. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated image sequences. *Int. Journal of Computer Vision*, 32(1):27–44, 1999.
- [86] P.H.S. Torr and D.W. Murray. Outlier detection and motion segmentation. In *Proc. SPIE Sensor Fusion VI*, pages 432–443, 1993.
- [87] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int. Journal of Computer Vision*, 24(3):271–300, 1997.
- [88] P.H.S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

-
- [89] T. Tuytelaars and L. Van Gool. Content-based image retrieval based on local affinely invariant regions. In *Visual Information and Information Systems*, pages 493–500, 1999.
- [90] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. British Machine Vision Conference*, pages 412–425, 2000.
- [91] H. Wang and D. Suter. Robust adaptive-scale parametric model estimation for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1459–1474, 2004.
- [92] E.W. Weisstein. Cubic formula. From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/CubicFormula.html>.
- [93] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough Transform (RHT). *Pattern Recognition Letters*, 11(5), 1990.
- [94] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *Int. Journal of Computer Vision*, 27(2):161–198, 1998.
- [95] Z. Zhang, R. Deriche, O.D. Faugeras, and Q.T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119, 1995.

Summary

The model underlying two views of the same scene or object is the epipolar geometry. It is the combined model of two differently positioned cameras that will reveal the relation between the camera projections. Since it is such a basic element in the analysis of multiple views of a scene, its computation by way of the fundamental matrix is important for many vision algorithms.

The standard way of computing the epipolar geometry is finding image correspondences by local feature matching. In Chapter 2 it is described how features are commonly computed at interest point locations, which are positions where the image shows some characteristic structure. Interest points are likely to be located in both images at projections of some conspicuous scene point. The local features at the interest points are then mutually compared to obtain a set of image-to-image correspondences. However, some of the correspondences may be erroneous or outlying, so that they yield useless information for the epipolar geometry. As a consequence, we need to use robust estimation methods which can cope with the presence of outliers. The RANSAC (RANdom SAmple Consensus) algorithm is particularly suitable for epipolar geometry estimation and has been frequently used for this purpose. It takes random samples of correspondences and evaluates the support for the corresponding fundamental matrices. This process continues until sufficient samples are taken to assure with high probability that an all-inlier sample has been found.

Unfortunately, the complexity of the algorithm increases substantially for higher outlier ratios. It is therefore necessary to devise methods which make faster execution of RANSAC possible. In this thesis we propose several methods for achieving this goal.

In general, the RANSAC algorithm samples correspondences uniformly, i.e. each correspondence has an equal probability of being sampled. In Chapter 3, the sampling probabilities of correspondences are deliberately adjusted to speed up the finding of an all-inlier sample. In particular, confidence values indicating the probability of dealing with an inlying correspondence are used to alter their frequency of selection. These confidence values are acquired by comparing the point correspondences to a series of planar quadrics. The distribution of outliers around the quadrics will be more or less random, while the inliers will have an imbalanced distribution for

imbalanced quadrics. This distinction can be made apparent by gathering statistics over a series of different planar quadrics. The effective outlier ratio that results after incorporating these statistics into the sampling probabilities, is likely to be lower than the original outlier ratio. Experiments on both synthetic and real data sets show that the effective outlier ratio is indeed smaller.

The adjustment of sampling probabilities is not necessarily performed prior to RANSAC execution. It is also possible to acquire the confidence values during execution of the algorithm. In Chapter 4 a weighted sampling scheme is proposed, which adjusts the sampling probabilities after each RANSAC iteration to increase the probability of sampling inliers. The probabilities are adjusted by positively weighting points from previous support sets. The algorithm that is obtained this way is not limited to epipolar geometry estimation. In particular, it is theoretically shown that the algorithm decreases the effective outlier ratio for hyperplane estimation among uniformly distributed outliers. Experiments on line and plane fitting in synthetic data confirm this fact. However, it is also shown that the algorithm is not well suited for multiple models, implying that there is no or only a minimal reduction in the number of iterations. The tested image pairs contain a single fundamental matrix, and weighted sampling shows an improvement for this type of data.

In Chapter 5, fundamental matrix estimation is performed using a combination of RANSAC and the Hough transform. It is shown that the nullspace of a smaller-than-minimal RANSAC sample can be used for the Hough voting stage. This principle is demonstrated on both hyperplanes and the fundamental matrix. No explicit parameterization of the model is required, which would be very difficult to obtain for the fundamental matrix. The combined method allows the drawing of 6-point samples instead of 7-point samples, so that it is much easier to find an all-inlier sample. The votes are cast in a two-dimensional space since six correspondences leave two remaining degrees of freedom. One degree can be removed by applying the singularity constraint, but voting in the two-dimensional space is more convenient than the parameterized one-dimensional space. A drawback of the combined method is the increased complexity of an iteration due to the voting stage. Fortunately, by following the principle of the Randomized Hough Transform, we can partly compensate for this by using just a subset of the data for voting. Experiments on both plane fitting and fundamental matrix estimation show, as expected, a reduction in the number of iterations. Faster execution is reached when the number of 6-point samples is much smaller than the number of 7-point samples, i.e. for large outlier ratios. A speedup is also obtained when the size of the subset used for voting is small in comparison to the number of correspondences.

The adjustment of sampling probabilities can be done in several different ways, e.g. by quadric comparisons or weighted sampling. Alternatively, the probabilities may be adjusted by using certain image properties, like regional similarity as applied for segmentation. This would allow grouping of correspondences and may enable the further identification of outliers.

In comparison to altering sampling probabilities, the use of smaller samples is more generally applicable. The search is then not hampered by possible other epipolar geometries. This is an important property which can make epipolar geometry estimation in complex scenes more efficient as well.

Summary of the thesis: “*Random Sampling Methods for Two-View Geometry Estimation*”.

R.J.M. den Hollander, Koudekerk aan den Rijn, March 2007.

Samenvatting

Het onderliggende model van twee afbeeldingen van dezelfde scène of van hetzelfde object is de epipolaire geometrie. Dit is het gecombineerde model van twee verschillend gepositioneerde camera's dat de relatie tussen hun projecties aangeeft. Aangezien het de basis vormt voor de analyse van meerdere afbeeldingen van een scène, is haar berekening door middel van de fundamentele matrix van belang voor vele beeldalgoritmes.

De gebruikelijke manier om de epipolaire geometrie te berekenen is het vinden van beeldcorrespondenties door vergelijking van lokale features. In hoofdstuk 2 wordt beschreven hoe deze features gewoonlijk worden berekend nabij interessante posities, waar de afbeelding een typische structuur vertoont. Deze interessante posities doen zich waarschijnlijk in beide plaatjes voor bij de projecties van een opvallend punt in de scène. De lokale features nabij de interessante posities worden daarna onderling vergeleken om een verzameling beeld-naar-beeld correspondenties te verkrijgen. Sommige correspondenties kunnen echter fout zijn of een outlier, zodat ze geen informatie geven over de epipolaire geometrie. Dit betekent dat we robuuste schattingsmethoden moeten gebruiken die geen last hebben van outliers. Het RANSAC (RANDOM SAMPLE CONSENSUS) algoritme is met name geschikt voor epipolaire geometrieschatting en wordt hiervoor vaak gebruikt. Het neemt random samples van een aantal correspondenties en bepaalt de support voor de bijbehorende fundamentele matrices. Dit proces herhaalt zich totdat voldoende samples zijn getrokken om een grote kans op een sample met alleen inliers te garanderen.

Helaas neemt de complexiteit van het algoritme aanzienlijk toe voor hogere outlierpercentages. Daarom is het noodzakelijk om methoden te ontwikkelen die RANSAC sneller kunnen maken. In dit proefschrift stellen we verschillende methoden voor om dit doel te bereiken.

In het algemeen trekt het RANSAC algoritme correspondenties op een uniforme manier, oftewel elke correspondentie heeft een gelijke kans om getrokken te worden. In hoofdstuk 3 worden de trekkingskansen van de correspondenties opzettelijk veranderd om het vinden van een sample met alleen inliers te bespoedigen. Betrouwbaarheidswaarden omtrent de kans met een inlier van doen te hebben worden gebruikt om hun trekkingskans te veranderen. Deze betrouwbaarheidswaarden worden verkregen

door de puntcorrespondenties te vergelijken met een aantal vlakke quadrics. De verdeling van de outliers rondom de quadrics zal min of meer willekeurig zijn, terwijl de inliers een ongebalanceerde verdeling zullen hebben voor ongebalanceerde quadrics. Dit verschil kan zichtbaar gemaakt worden door statistiek te bedrijven voor verschillende vlakke quadrics. Het gebruik van deze gegevens voor aanpassing van de trekkingskansen leidt tot een effectief outlierpercentage dat waarschijnlijk lager is dan het originele outlierpercentage. Resultaten van artificiële en echte datasets tonen aan dat het effectieve outlierpercentage inderdaad kleiner is.

Het is niet noodzakelijk om de trekkingskansen vóór uitvoering van het RANSAC algoritme aan te passen. Het is ook mogelijk om de betrouwbaarheidswaarden tijdens het uitvoeren van dit algoritme te verkrijgen. In hoofdstuk 4 wordt een gewogen trekkingsmethode voorgesteld die na elke RANSAC iteratie de trekkingskansen aanpast met het doel de inlier trekkingskans te verhogen. De aanpassing bestaat hier uit het toekennen van extra gewicht aan de data punten uit vorige support sets. Het resulterende algoritme is echter niet alleen toepasbaar voor het schatten van de epipolaire geometrie. Om precies te zijn, er wordt theoretisch aangetoond dat het algoritme het effectieve outlierpercentage verlaagt voor hypervlakschatting temidden van uniform verdeelde outliers. Dit wordt bevestigd door experimenten van lijn- en vlakschatting in artificiële data. Echter, er wordt tevens aangetoond dat het algoritme niet geschikt is voor meerdere modellen, wat inhoudt dat er geen of slechts een minimale besparing van het aantal iteraties is. De geteste beeldparen bevatten een enkele fundamentele matrix, en de gewogen trekkingsmethode laat een verbetering zien voor dit type data.

In hoofdstuk 5 wordt de fundamentele matrix geschat met een combinatie van RANSAC en de Hough transformatie. Er wordt aangetoond dat de nulruimte van een kleiner-dan-minimaal RANSAC sample gebruikt kan worden voor het accumuleren van de Hough ruimte. Dit principe wordt gedemonstreerd voor hypervlakken en de fundamentele matrix. Er is hierbij geen expliciete parameterisatie van het model nodig, dat ook zeer moeilijk te verkrijgen zou zijn voor de fundamentele matrix. De gecombineerde methode maakt het mogelijk om 6-punts samples in plaats van 7-punts samples te trekken, zodat het veel makkelijker wordt om een sample met alleen inliers te vinden. Het accumuleren gebeurt in een tweedimensionale ruimte omdat er twee vrijheidsgraden overblijven bij zes correspondenties. Eén vrijheidsgraad kan worden weggenomen door de singulariteitsvoorwaarde toe te passen, maar het accumuleren in een tweedimensionale ruimte is toch makkelijker dan in een geparameteriseerde eendimensionale ruimte. Een nadeel van de gecombineerde methode is dat een iteratie complexer wordt door het accumuleren. Gelukkig kan dit gedeeltelijk gecompenseerd worden door hiervoor een subset van de data te gebruiken, in navolging van de Randomized Hough Transformatie. Zoals verwacht laten experimenten voor vlak- en fundamentele matrixschatting een afname van het aantal iteraties zien. Een snellere uitvoering wordt daadwerkelijk bereikt als het aantal 6-punts samples veel kleiner is dan het aantal 7-punts samples, oftewel voor hogere

outlierpercentages. Dit is tevens het geval als de grootte van de subset voor de accumulatie klein is in vergelijking met het aantal correspondenties.

Het aanpassen van de trekkingskansen kan in het algemeen op verschillende manieren gebeuren, bijvoorbeeld door vergelijking met quadrics of met een gewogen trekkingsmethode. Een alternatief is het gebruiken van bepaalde eigenschappen van de afbeelding, zoals lokale gelijkenissen die worden gebruikt voor segmentatie. Dit zou het groeperen van correspondenties mogelijk maken en een verdergaande identificatie van outliers.

In vergelijking met het aanpassen van de trekkingskansen, is het gebruik van kleinere samples breder toepasbaar. Het zoeken kan dan niet gehinderd worden door mogelijke andere epipolaire geometrieën. Dit is een belangrijke eigenschap die het schatten van de epipolaire geometrie in complexe scènes ook efficiënter kan maken.

Samenvatting van het proefschrift: “*Random Trekkingsmethoden voor Geometrieschatting van Beeldparen*”.

R.J.M. den Hollander, Koudekerk aan den Rijn, Maart 2007.

Acknowledgements

It is a great pleasure to mention the people who have made the creation of this thesis possible.

First of all, I would like to thank my supervisor Alan Hanjalić for guiding me in the world of science. I learned a lot from our discussions and enjoyed the freedom you let me in doing research. As a roommate, I also thank you for your willingness to answer all my questions.

Thanks to my professor Inald Lagendijk for supporting me over the years. Your advice regarding my research ideas was always highly valuable.

I really enjoyed the time that I was working in the Information and Communication Theory group. The atmosphere was great and there were plenty of opportunities to learn about other things than just your own research. In particular, I would like to thank Marcel Reinders and Emile Hendriks for organizing all the paper discussions which led me to a better understanding of computer vision. I am also grateful for the assistance of the supporting staff members Ben, Hans, Robbert, Anja, Annett and Sanne, who were of great help in keeping my computer running and handling all sorts of formalities. Also many thanks to all Ph.D. students and postdocs whose combined efforts have resulted in many amusing social events.

Finally, I would like to thank my family for supporting me all those years. Your encouragement has helped me to complete this journey.

Curriculum Vitae

Richard Jacobus Maria den Hollander was born in Leiderdorp on March 8, 1978. He obtained his Gymnasium diploma at the Groene Hart Lyceum in Alphen aan den Rijn in 1996. The same year he started studying electrical engineering at Delft University of Technology, where he received his M.Sc. degree in 2001. His graduation project on digital watermarking was carried out at the Information and Communication Theory group, and resulted in a thesis entitled “Information hiding for sources with memory: a capacity calculation.”

In 2002 he started as a Ph.D. student in the Information and Communication Theory group at Delft University of Technology. Initially he worked on object recognition techniques, but later his research shifted towards two-view geometry estimation, of which this thesis is the result.

Since 2006 he works at TNO Defence, Security and Safety in The Hague, where he contributes to the development of image processing algorithms for camera systems.

