EÖTVÖS LORÁND UNIVERSITY

DOCTORAL THESIS

Affine Correspondences and their Applications for Model Estimation

Author:
Dániel BARÁTH

Supervisor:
Dr. Levente HAJDER

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Department of Algorithms And Their Applications

July 4, 2019

Eötvös Loránd University

Abstract

Department of Algorithms And Their Applications

Doctor of Philosophy

Affine Correspondences and their Applications for Model Estimation

by Dániel BARÁTH

This work aims to solve sub-problems of two major fields in computer vision: minimal problems in two- or multi-view geometric model estimation and robust model fitting. Minimal solvers, i.e. algorithms solving estimation problems from a minimal sample of data points, are involved in most of the vision pipelines as an the engine of the applied robust method, e.g. RANSAC [1] and its recent variants. Vision pipelines, including calibration, structure-from-motion, image matching and retrieval, benefits from efficient minimal solvers which improve their performance upon. Given a minimal point correspondence set in two views, state-of-the-art solvers, with a few exceptions, use them solely through their coordinates. Nevertheless, as it will be demonstrated in this thesis, exploiting affine correspondences which encode higher-order geometric information leads to methods superior to the state-of-the-art in terms of stability and the number of data points required. Methods will be proposed for surface normal, homography, epipolar geometry, and focal length estimation.

The second major part of this work focuses on robust model fitting which is also a significant part of vision tasks. The base problem is to fit a single or more model instances, e.g. planes to a 3D point cloud or fundamental matrix to point correspondences, interpreting the input whilst it is contaminated by noise and contains outliers. We consider outliers as points not belonging to any desired model instance. First, a method is proposed to distinguish inliers and outliers in a set of correspondences without necessarily assuming an underlying model. Then a new local optimization step is proposed for locally optimized RANSAC (LO-RANSAC) outperforming its state-of-the-art variants. Finally, we focus on multi-homography, then general multi-class multi-instance, fitting – the problem of interpreting the input data as a mixture of noisy observations originating from multiple instances of multiple classes. The methods proposed in this work were validated both on synthesized and publicly available real world datasets.

Contents

List of Figures

List of Tables

List of Symbols

$\mathbb{R}$	the set of the real numbers
$\mathbb{P}^n$	the projective space representing vectors of $\mathbb{R}^n$ and the ideal points
$a, b, \cdots, \alpha, \beta, \cdots$	a scalar number (real-valued if not denoted otherwise)
$\mathcal{A}, \mathcal{B}, \cdots$	calligraphic letters denote sets
$\mathbf{x}, \mathbf{y}, \cdots$	Euclidean vector from $\mathbb{R}^n$
$\mathbf{B}, \mathbf{C}, \cdots$	matrix from $\mathbb{R}^{n \times m}$
$[\mathbf{x}]_\times$	3×3 cross-product matrix of vector $\mathbf{x}$
$\mathbf{x} \times \mathbf{y}$	cross-product of vectors $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{a}^\mathrm{T} [\mathbf{n}]_\times \mathbf{b}$	scalar triple product
$\mathbf{p} = [x \quad y \quad 1]^\mathrm{T}$	homogeneous image point
$\mathbf{P}$	3×4 perspective projection matrix
$\mathbf{K}$	3×3 intrinsic camera matrix
$\mathbf{F}$	3×3 fundamental matrix
$\mathbf{E}$	3×3 essential matrix
$\mathbf{H}$	3×3 homography matrix
$\mathbf{A}$	2×2 local affine transformation
$\mathbf{R}^\alpha$	2D rotation matrix with α angle
$\det(\mathbf{M})$	determinant of square matrix $\mathbf{M}$
$\mathrm{tr}(\mathbf{M})$	trace of square matrix $\mathbf{M}$
$\mathrm{diag}(v_1, \cdots, v_n)$	$n \times n$ diagonal matrix with values values $v_1, \cdots, v_n$
$\mathbf{B}^\dagger = (\mathbf{B}^\mathrm{T}\mathbf{B})^{-1}\mathbf{B}^\mathrm{T}$	the Moore-Penrose pseudo-inverse of matrix $\mathbf{B}$
$[\![.]\!]$	the Iverson-bracket which is 1 if the condition inside holds, otherwise, 0
$\lvert\mathcal{A}\rvert$	the cardinality of a set
$\lvert\mathbf{x}\rvert$	the L_2-norm of vector $\mathbf{x}$
$\lvert\mathbf{M}\rvert$	the Frobenius-norm of matrix $\mathbf{M}$
$\mathbf{M}_{[i:k,j:l]}$	the $(k-i+1) \times (l-j+1)$-sized sub-matrix of matrix $\mathbf{M}$ $(0 < i < k, 0 < j < l)$
$\mathbf{v}_{[i:k]}$	the vector consisting of the elements of vector $\mathbf{v}$ from ith to kth $(i < k)$
L	a labeling
$\sim$	equality up to an arbitrary scale
$\triangle$	the standard symmetric difference operator of sets
$\forall$	for all
$\exists$	there exist(s)
$\div$	integer division operator
$\%$	modulo operator

Chapter 1

Introduction

During the last few decades good-quality cameras and sensors had become publicly available. This progress established the need for a research field describing the mathematical relationships of the real world and its projections into images. This field became the so-called *computer vision*.

Understanding the surrounding environment or the camera motion are fundamental problems in computer vision. This information is usually characterized by mathematical models, for instance the relative motion of the cameras by a 3×3 essential matrix; or the mapping between the projections of a 3D plane in two images by a 3×3 homography matrix. The general approach for estimating these kinds of models consists of two major steps: (i) establishing point correspondences between image pairs, (ii) then applying a robust estimator. In this thesis, we investigate the *robust* estimation of these geometric models exploiting a non-traditional input data: affine correspondences. Even though the relationships of geometric vision are considered as already solved problems advanced in the early 80's, we show that several problems remained unsolved. The thesis aims to solve a few of them.

An affine correspondence is basically a point correspondence in two views together with a 2×2 local affine transformation. This affine mapping approximately transforms the regions around the observed points in the images. Nowadays, several affine covariant feature detectors are available, such as Affine SIFT [2] or Hessian-Affine [3]. However, the commonly used detectors like SIFT [4] also provide a part of the related affine transformation, e.g. rotation or scales. With a few exceptions, *this additional information is ignored* in most of the geometric estimation tasks and solely the point coordinates are exploited.

Of course, several estimation problems had been successfully approached by using affine correspondences. For instance, a homography matrix can be estimated using two correspondences [5]. There is a *one-to-one* relationship between a surface normal and a local affinity if calibrated cameras are given [5], and the fundamental matrix can be approximated [6], [7] or estimated indirectly using three [8] or two [9] of them. However, several problems remained unsolved including the direct relationship of local affinities and epipolar geometry or multi-view surface normal estimation. In this thesis, we show that many of the computer vision problems are solvable using local affinities. The proposed estimators always require smaller samples, i.e. less data points, than the state-of-the-art for obtaining a model. Moreover, in many cases, the proposed methods are superior to the state-of-the-art in terms of geometric accuracy as well. We also show problems where the direct relationship with affinities had not been explored before.

Having geometrically accurate estimators is a justifiable goal, but to see the impact of requiring small samples, we need to understand the field where minimal

solvers are most frequently used. State-of-the-art *hypothesize-and-verify* robust estimators like locally optimized RANSAC [10] (LO-RANSAC) or USAC [11] are randomized algorithms combined with a minimal solver as an engine. To achieve probabilistic guaranties of finding the best desired model instance with a predefined confidence, the size of a sample highly affects the number of samples, i.e. iteration, that have to be drawn. Thus the processing time is a function of the sample size. Benefiting from affine correspondences which encode higher-order geometric information, the size of the required samples is significantly decreased and, thus, the estimation process is speeded up.

In the second part of the thesis, we switch from geometric estimation and focus on *robust model estimation*. We partition these kind of problems into three groups as follows: (i) single-class single-instance, (ii) single-class multi-instance and (iii) multi-class multi-instance fitting. Solving the *single-class single-instance case*, hypothesize-and-verify approaches like RANSAC [1] and its recent variants, had become a part of the most successful algorithms in computer vision. They have thousands of citations and dozens of modifications published year-by-year. In general, these methods aim to find a single model instance, e.g. an essential matrix interpreting the relative motion of a camera, by drawing and validating random samples.

Generalizing the problem, multi-model fitting has been studied since the early sixties, the Hough-transform [12] being the first popular method for extracting multiple instances of a single class. Most recent approaches [13], [14] focus on the single class case: finding multiple instances of the same model class. A popular group of methods [13], [15] adopts a two step process: initialization by RANSAC-like instance generation followed by a point-to-instance assignment optimization by *energy minimization* using graph labeling techniques [16]. Another group of methods uses *preference analysis*, introduced by RHA [17], which is based on the distribution of residuals of individual data points with respect to the instances. In this thesis, we approached a special case: multi-homography fitting which is the problem of finding a set of homographies in two images.

The *multiple instance multiple class case* considers fitting of instances that are not necessarily of the same class. This generalization has received much less attention than the single-class case. To the best of our knowledge, the last significant contribution is that of Stricker and Leonardis [18] who search for multiple parametric models simultaneously by minimizing description length using Tabu-search. In the last part of the thesis, we propose a general formulation for the multi-class case and show that it leads to results superior to the state-of-the-art single-class approaches for various problems.

1.1 Main Contributions

The contributions of this thesis can be separated into two distinct groups: (a) *minimal solvers* exploiting affine correspondences for various problems and (b) *robust methods* for estimating geometric models.

1.1.1 Published Papers

Impacted journal articles.

[19] Barath, D.; Efficient energy-based topological outlier rejection, Computer Vision and Image Understanding, 2018
[20] Barath, D.; Hajder, L.; Efficient Recovery of Essential Matrix From Two Affine Correspondences, Transactions on Image Processing, 2018
[21] Barath, D.; Hajder, L.; A Theory of Point-wise Homography Estimation, Pattern Recognition Letters, 2017

Conference papers.

[22] Barath, D.; Matas, J.; Multi-Class Model Fitting by Energy Minimization and Mode-Seeking, In Proceedings of the European Conference on Computer Vision, 2018
[23] Barath, D.; Matas, J.; Graph-Cut RANSAC, In Proceedings of the Conference on Computer Vision and Pattern Recognition, 2018
[24] Barath, D.; Five-point Fundamental Matrix Estimation for Uncalibrated Cameras, In Proceedings of the Conference on Computer Vision and Pattern Recognition, 2018
[25] Barath, D.; Toth, T.; Hajder, L.; A Minimal Solution for Two-view Focal-length Estimation using Two Affine Correspondences, In Proceedings of the Conference on Computer Vision and Pattern Recognition, 2017
[26] Barath, D.; P-HAF: Homography Estimation Using Partial Local Affine Frames, 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2017
[27] Barath, D.; Matas, J.; Hajder, L., Multi-H: Efficient Recovery of Tangent Planes in Stereo Images, 27th British Machine Vision Conference, 19-22 September, York, England,28, 32, 2016
[28] Barath, D.; Hajder, L.; Matas, J.; Accurate Closed-form Estimation of Local Affine Transformations Consistent with the Epipolar Geometry, 27th British Machine Vision Conference, 2016
[29] Barath, D.; Hajder, L.; Energy-based Topological Outlier Filtering, 23rd International Conference on Pattern Recognition,2016
[30] Barath, D.; Hajder, L., Novel Ways to Estimate Homography from Local Affine Transformations, In Proceedings of the 11th International Conference on Computer Vision Theory and Application, 3, 432-443, 2016
[31] Barath, D.; Eichhardt, I., A Novel Technique for Point-wise Surface Normal Estimation, In Proceedings of the 11th International Conference on Computer Vision Theory and Applications, 3, 686-693, 2016
[32] Barath, D.; Molnar, J.; Hajder, L., Novel methods for estimating surface normals from affine transformations, In Proceedings of the 10th International Joint Conference on Computer Vision, Imaging and Computer Graphics, 316-337, 2015
[33] Molnar, J.; Csetverikov, D.; Kato, Z.; Barath, D.; A Theory of Camera-Independent Correspondence, 2015
[34] Barath, D.; Molnar, J.; Hajder, L.; Optimal surface normal from affine transformation, 2015, SciTePress

Papers on ArXiV.

[35] Barath, D.; Matas, J.; MAGSAC: marginalizing sample consensus, 2018

Conference papers published in Hungarian.

[36] Barath, D.; Matas, J.; Hajder, L.; Epipoláris Geometriával Konzisztens, Legközelebbi Affin Transzformáció Optimális Becslése, 2017, Képfeldolgozók és Alakfelismerők Társaságának 11. konferenciája
[37] Barath, D.; Matas, J.; Hajder, L., Multi-H: érintősíkok Hatékony Kinyerése Képpárokból, 2017, Képfeldolgozók és Alakfelismerők Társaságának 11. konferenciája
[38] Eichhardt, I.; Barath, D., Felületi normális becslése egyetlen pontmegfeleltetés alapján, 2017, Képfeldolgozók és Alakfelismerők Társaságának 11. konferenciája
[39] Barath, D.; Hajder, L.; Homográfia becslése részlegesen ismert affin transzformációból, 2016, GRAFGEO, Neumann János Számítógép-tudományi Társaság
[40] Hajder, L.; Barath, D.; Molnar, J.; Normálvektorok optimális becslése affin transzformációkból, 2015, GRAFGEO, Neumann János Számítógéptudományi Társaság
[41] Barath, D.; Homográfiabecslés affin transzformációból, 2015, Képfeldolgozók és Alakfelismerők Társaságának 9. konferenciája

1.1.2 Contribution

Minimal solvers. An optimal method, in the least squares sense, is proposed to estimate surface normals in both stereo [34], [42] and multi-view cases[1]. The proposed algorithm exploits exclusively photometric information via affine correspondences and estimates the normal for each correspondence independently. The normal is obtained as a root of a quartic polynomial, therefore the processing time is negligible. The method has been validated on both synthetic and publicly available real world datasets. It is superior to the state-of-the-art in terms of accuracy and processing time.

We propose a method, called HAF, to estimate planar homography from an affine correspondence satisfying the epipolar constraint in an image pair [21], [32]. As a minimal solver, it estimates the homography *from a single correspondence*, however, it is generalized for the over-determined case as well. As a side-effect of the tests, the state-of-the-art affine-covariant detectors are compared to each other w.r.t. the accuracy of the estimated point-wise homographies. We then generalized HAF, making it applicable if only partial affine correspondences are given. [26]

We then show the direct relationship of epipolar geometry and affine correspondences. Two novel, linear constraints are derived between the essential or fundamental matrices and a local affine transformation. Even though perspective cameras are assumed, the constraints can straightforwardly be generalized to arbitrary camera models since they describe the direct relationship between local affinities and epipolar lines (or curves).

Exploiting this relationship, for a pair of images satisfying the epipolar constraint, a method for accurate estimation of affine correspondences is proposed. The method returns the local affine transformation consistent with the epipolar geometry that is closest in the least squares sense to the initial estimate provided by an affine-covariant detector. The minimized L_2 norm of the affine matrix elements is found in closed-form [28]. The method, with negligible computational requirements, is

[1] A paper, entitled *Optimal Multi-View Surface Normal Estimation using Affine Correspondences*, was submitted to Transactions on Image Processing

validated on publicly available benchmarking datasets and on synthetic data. The accuracy of the local affine transformations is improved for all detectors and all image pairs. Implicitly, precision of the tested feature detectors was compared.

It is shown that the essential matrix is estimable from two affine correspondences for a pair of calibrated perspective cameras. The proposed method is also applicable to the over-determined case. We extend the normalization technique of Hartley to local affinities and show how the intrinsic camera matrices modifies them. Benefiting from the low number of exploited points, it can be used in robust estimators, e.g. RANSAC, as an engine, thus leading to significantly less iterations.

A minimal solution using two affine correspondences is presented [25] to estimate the common focal length and the fundamental matrix between two semi-calibrated cameras – known intrinsic parameters except a common focal length. The obtained multivariate polynomial system is efficiently solved by the hidden-variable technique. Observing the geometry of local affinities, we introduce novel conditions eliminating invalid roots. To select the best one out of the remaining candidates, a root selection technique is proposed outperforming the recent ones especially in case of high-level noise.

Robust methods. An approach is proposed [19], [29] for outlier rejection from a set of 2D point correspondences which does not require any underlying models, e.g. fundamental matrix. The solution is obtained minimizing an energy originated from the neighborhood-graphs in both images using a grab-cut-like algorithm: iterated graph-cut and re-fitting. The method is validated on publicly available datasets, it is real time for most of the problems and achieves more accurate results than RANSAC and its state-of-the-art variants in term of outlier rejection ratio. It is applicable to scenes where a single fundamental matrix is not estimable, e.g. non-rigid or degenerate ones.

A novel method, called Graph Cut RANSAC [23], GC-RANSAC in short, is presented. To separate inliers and outliers, it runs the graph cut algorithm in the local optimization (LO) step which is applied after a *so-far-the-best* model is found. The proposed LO step is conceptually simple, easy to implement, globally optimal and efficient. Experiments show that GC-RANSAC outperforms LO-RANSAC and its state-of-the-art variants in terms of both accuracy and the required number of iterations for line, homography and fundamental matrix estimation on public datasets.

Considering the problem of fitting multiple homographies in two views, we proposed an efficient method [27] for the recovery of the tangent planes of a set of point correspondences satisfying the epipolar constraint. The problem is formulated as a search for a labeling minimizing an energy that includes a data and spatial regularization terms. Experiments on the fountain-P11 3D dataset show that Multi-H provides highly accurate tangent plane estimates. It also outperforms all state-of-the-art techniques for multi-homography estimation on the publicly available AdelaideRMF dataset. Since the method achieves nearly error-free performance, we introduce a more challenging dataset for multi-plane fitting evaluation.

In the end of the thesis, we propose a general formulation, called Multi-X [22], for multi-class multi-instance model fitting – the problem of interpreting the input data as a mixture of noisy observations originating from multiple instances of multiple classes. Solving the problem, we augment the commonly used α-expansion-based technique with a new move in the label space. The move replaces a set of labels with the corresponding mode in the model parameter domain, thus achieving faster and more robust minimization. Key optimization parameters like the band-width of the mode-seeking are set automatically within the algorithm. Considering that a group

of outliers may form spatially coherent structures in the data, we propose a cross-validation-based technique removing statistically insignificant instances. Multi-X outperforms significantly the state-of-the-art on publicly available datasets for diverse problems: multiple plane and rigid motion detection; motion segmentation; simultaneous plane and cylinder fitting; circle and line fitting.

Chapter 2

Theoretical Background

In this section, we discuss the topics closely related to this thesis but do not belong to the *fundamental knowledge* of computer vision. We consider projective and epipolar geometry, homography, least-squares estimation, robust estimation, etc. as parts of this knowledge. Thus they are not not discussed deeply in the thesis. For the broad understanding of the topic, we suggest to read the book of Hartley and Zisserman [43] first.

2.1 Affine Correspondences

In this paper, we consider an affine correspondence (AC) as a triplet: $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{A})$, where $\mathbf{p}_1$ and $\mathbf{p}_2$ are a corresponding point pair in the two images, and

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$$

is a 2×2 linear transformation which we call in the latter sections *local affine transformation* (see Fig. 2.1). To define a local affine transformation, we use the definition provided in [44] as it is given as the first-order Taylor-approximation, w.r.t. the image directions, of the 3D $\rightarrow$ 2D projection functions. This is shown in depth in Appendix B. For perspective cameras, $\mathbf{A}$ is the first-order approximation of the related *homography* matrix.

Although, in the literature, one can find affine correspondences referred as *affine frames*, we differentiate them. An affine frame is a triplet of point correspondences $(\mathbf{p}_1^1, \mathbf{p}_2^1)$, $(\mathbf{p}_1^2, \mathbf{p}_2^2)$, $(\mathbf{p}_1^3, \mathbf{p}_2^3)$ providing only an approximation of the related affine correspondence. Without proving the difference, it can easily be seen that $\mathbf{A}$, as the first-order approximation of the projection functions, is valid only infinitesimally close to the observed correspondences. This "infinitesimally closeness" can only be *approximated* by a triplet of correspondences, to the best of our knowledge.

In the rest of the thesis, affine correspondences are considered as input provided by an affine-covariant feature detector. These detectors, including Affine SIFT [2] (ASIFT), Hessian-Affine [3], MSER [45], obtain point coordinates and local affine transformations simultaneously. We will distinguish two types of them: (i) ones providing ACs calculated from three correspondences (from affine frames) like MSER [45]. (ii) Other types of detectors, like ASIFT [2] and MODS [46], obtain matrix $\mathbf{A}$ directly by sampling the affine space; or by applying an optimization of a photometric cost function, e.g. Hessian-Affine, Harris-Affine. To obtain accurate ACs, we chose Hessian-Affine combined with the view-synthesizer of ASIFT.

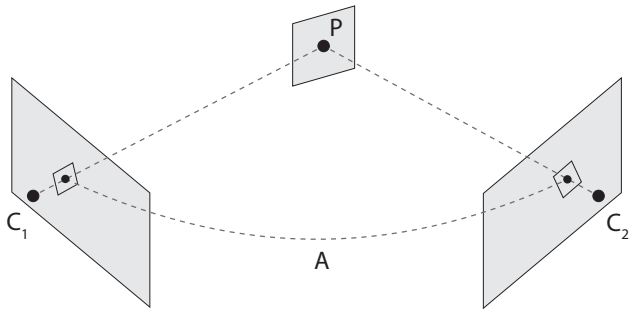


FIGURE 2.1: Cameras $\mathbf{C}_1$ and $\mathbf{C}_2$ observing a point $\mathbf{P}$ lying on a continuous surface, e.g. plane. The neighboring pixels of the projected points between the 1st and 2nd views are related by a local affine transformation $\mathbf{A}$.

2.2 Data Normalization

In this section, we show the data normalization technique which we use for fundamental matrix and homography estimation. We just write here the final formulas since all the proofs are available in [43].

Given a set $\{(\mathbf{p}_1^j, \mathbf{p}_2^j)\}_{j=1}^n$ of $n \in \mathbb{N}$ point correspondences in their homogeneous form. Normalizing transformation $\mathbf{T}_i$ in the ith image ($i \in \{1, 2\}$) is as follows:

$$\mathbf{T}_i = \begin{bmatrix} \sqrt{2}/d_i & 0 & 0 \\ 0 & \sqrt{2}/d_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\bar{x}_i \\ 0 & 1 & -\bar{y}_i \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

where $\bar{\mathbf{p}}_i = [\bar{x}_i, \bar{y}_i, 1]^\mathrm{T}$ is the mean of the point set in the ith image and

$$d_i = \frac{1}{n} \sum_{j=1}^{n} \sqrt{(\mathbf{p}_i - \bar{\mathbf{p}})^\mathrm{T} (\mathbf{p}_i - \bar{\mathbf{p}})} \tag{2.2}$$

is the average distance of the points from $\bar{\mathbf{p}}_i$. The normalized correspondence set is as follows: $\{(\hat{\mathbf{p}}_1^j, \hat{\mathbf{p}}_2^j)\}_{j=1}^n = \{(\mathbf{T}_1 \mathbf{p}_1^j, \mathbf{T}_2 \mathbf{p}_2^j)\}_{j=1}^n$.

For fundamental matrix estimation, using the normalized correspondences leads to significantly more accurate results [47]. After the estimation, $\mathbf{F}$ is calculated from the normalized fundamental matrix $\hat{\mathbf{F}}$ as follows: $\mathbf{F} = \mathbf{T}_2^{-\mathrm{T}} \hat{\mathbf{F}} \mathbf{T}_1^{-1}$.

Homography estimation from the normalized correspondences also leads to results superior. Homography $\mathbf{H}$ is recovered from the normalized one as follows: $\mathbf{H} = \mathbf{T}_2^{-1} \hat{\mathbf{H}} \mathbf{T}_1$.

2.3 Iteration number of Random Sample Consensus

In this section, we discuss the required iteration number of RANSAC. Suppose that $n \in \mathbb{N}$ data points are given and $l \in \mathbb{N}$ ($l \leq n$) of them are inliers. The inlier ratio, i.e. the probability of selecting an inlier if uniform distribution is considered, is $p = \frac{l}{n}$ ($\in [0, 1]$). Selecting a sample which consists of $m \in \mathbb{N}$ inliers leads to probability

TABLE 2.1: Theoretical iteration numbers for RANSAC. $q \in [0, 1]$ is the desired probability, $p \in [0, 1]$ the inlier ratio, $n \in \mathbb{N}$ the point number used for the estimation.

$q \rightarrow$	0.95			0.99		
m/p	0.75	0.50	0.25	0.75	0.50	0.25
1	2	4	10	3	7	16
2	4	10	46	6	16	71
3	5	22	190	8	34	292
4	8	46	765	12	71	1 177
5	11	94	3 066	17	145	4 713
6	15	190	12 269	23	292	18 860
7	21	382	49 081	32	587	75 449
8	28	765	196 327	44	1 177	301 802

p^m. The number of iterations required $k \in \mathbb{N}$ which decreases $(1 - p^m)^k$ below a user-defined confidence value $q \in [0, 1]$ is as follows:

$$k \geq \frac{\log(1 - q)}{\log(1 - p^m)}. \tag{2.3}$$

Example values are reported in Table 2.1 for confidence values 0.95 and 0.99 (1st row). The iteration numbers of different minimal sample sizes (1st column) are shown for varying inlier percentage (2nd row).

2.4 Minimal Solvers

In this thesis, we describe *minimal solvers* as methods which estimate geometric models from *minimal samples* exploiting a predefined constraint set. A minimal sample is a set consisting of as few data points as required for the estimation.

As an example, for estimating fundamental matrix $\mathbf{F}$ in two images, more than one minimal solvers exist. The eight-point algorithm [47] considers no constraints but the scale-ambiguity of $\mathbf{F}$ and the well-known relationship of point correspondences and fundamental matrices: $\mathbf{p}_2^\mathrm{T}\mathbf{F}\mathbf{p}_1 = 0$, where $\mathbf{p}_1$ and $\mathbf{p}_2$ are the points in the images. To reduce the size of the required sample, the seven-point algorithm [43] enforce the so-called *determinant constraint* stating that $\det(\mathbf{F}) = 0$. Even though the seven-point method is "more minimal" we consider both methods as minimal solvers since w.r.t. a predefined constraint set, they provide an estimation from a minimal sample.

Justifying the need for minimal solvers, state-of-the-art *hypothesize-and-verify* robust estimators, e.g. RANSAC [1] are randomized algorithms selecting a minimal sample as a first step, generating a hypothesis using a minimal solver, then verifying it w.r.t. data points. Combining these methods with a solver which exploits less data leads to more stable results and earlier termination due to the combinatorics of the problem (see Table 2.1). From theoretical point of view, each minimal solver interprets the solved problem more efficiently and effectively than the ones before and thus, leads to deeper understanding.

Chapter 3

Estimating Planes and their Projections

In this chapter, we discuss the exploitation of affine correspondences for estimating (i) surface normals and (ii) homographies. Each of the proposed methods uses fully or partially known affine correspondences and aims to solve a minimal problem.

3.1 Optimal Multi-View Surface Normal Estimation

Even though computer vision has been an intensively researched area in computer sciences for decades, several unsolved problems exist in the field. The one, we aim at in this section, is the analytic estimation of surface normals in a multi-view system exploiting exclusively photometric information, i.e. affine correspondences. The spatial relationship of the points is not considered thus achieving point-wise estimation without requiring dense clouds.

Several tasks, including surface reconstruction and segmentation, or object detection, require accurate surface normals. Benefiting from the higher-order information which they encode, the accuracy of surface reconstruction improves upon. For instance, the widely-used Poisson-reconstruction technique [48], [49] is based on both the point coordinates and surface normal. Having an oriented point cloud makes geometric primitive fitting, e.g. that of planes or cylinders, significantly easier due to the fact that less points are enough for the model-hypothesis generation. This number highly influences state-of-the-art multi-model fitting algorithms like PEARL [13] in terms of accuracy and processing time. As an example, plane fitting needs at least one oriented or three non-oriented points.

One of the first algorithms solving the surface normal estimation problem was the photometric stereo (PS) method [50]. Requiring totally controlled light conditions, the applicability of PS is limited into the laboratory. The original PS assumes Lambertian surface, thus not dealing with shiny materials, and estimates the normal using the so-called "Bidirectional Reflectance Distribution Function" [51] with known light-source parameters. However, several modifications, e.g. [52], [53], have been proposed since then, making it more accurate and applicable to various materials.

Between two calibrated views, the normal estimation problem is usually approached by decomposing the homographies of corresponding image patches [54], [55]. For calibrated views, a homography can be interpreted as the tangent plane of the surface at the observed 3D location, and the normal can accurately be computed. However, the decomposition itself is ambiguous as it was shown by several studies, e.g. in [56], and homography estimation cannot be done for each point correspondence independently. Thus, in general, these methods are applied to superpixel

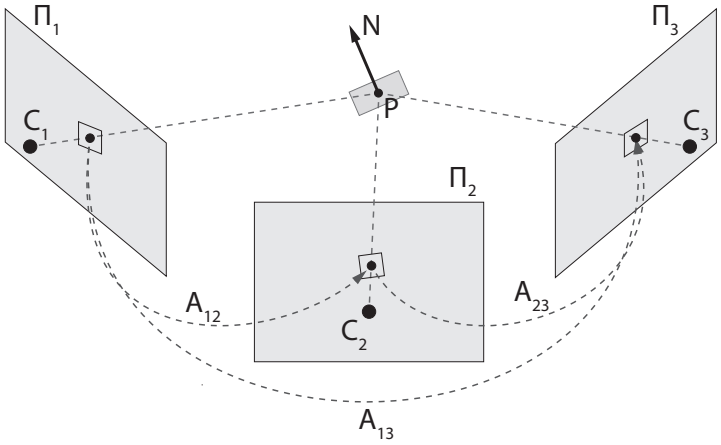


FIGURE 3.1: Three cameras observing a point **P** on a plane with normal **N**. The neighboring pixels of the projected points between the ith and jth views are related by a local affine transformation $\mathbf{A}_{ij}$.

correspondences, i.e. corresponding image regions, supposing that the underlying surface patch is planar.

In 2009, Kevin Köser [5] proposed a technique exploiting local affine transformations. In brief, a local affinity can be interpreted as the partial derivative, w.r.t. the image directions, of the underlying homography at the observed location. Therefore, it encodes higher-order geometric information, i.e. the surface normal. To the best of our knowledge, the method in [5] was the first which made the analytical point-wise normal estimation achievable between two views since local affinities can be measured by affine-covariant feature detectors, e.g. Hessian-Affine [57], Affine-SIFT [2] or MODS [46], for each point correspondence independently. Benefiting from this approach, the ambiguity, to which the homography decomposition leads, disappeared.

Considering multiple views, an objective of several structure-from-motion (SfM) pipelines is to estimate the surface normals accurately since they contain fundamental information for the further surface reconstruction. The well-known algorithm, called Patch-based Multi-View Stereo (PMVS) proposed by Furukawa et al. [58], [59], solves the problem as an optimization numerically refining the plane parameters to minimize a joint photometric cost function. The cost is based on zero-mean cross-correlation applied to patches, each transformed by the homography which the plane induces. [60] approaches the problem similarly to PMVS, assuming that the surfaces can be represented by local planar patches. It proposes a unified cost function considering both geometric and photometric terms. These methods obtain accurate surface normals, nevertheless, they are sensitive to the size of the patch for which the photometric cost is computed, i.e. the window size. Being solved numerically, they are relatively slow and do not guarantee global optimum.

The contributions of the section are: (i) we propose an analytic multi-view normal estimation technique which is optimal in the least squares sense and exploits local affine transformations (see Fig. 3.1). First, we show the relationship of local affinities and surface normals considering two views, then this approach is extended. To the best of our knowledge, this is the *first analytic solution* applicable to the multiple view case. The equations are not linearized, therefore, the *globally optimal* solution is carried out efficiently as a root of a fourth-order polynomial thus achieving *fast calculation*. (ii) Reflecting the fact that the estimation of local affinities is sensitive to

the view angle, thus a measured set of affinities might contain outliers, we propose a robust estimation technique. It is reported both on synthesized and real world tests, that the proposed method outperforms the state-of-the-art in terms of accuracy and processing time. (iii) Besides, we demonstrate the applicability of the method on two problems: replacing the seed-point generation step of PMVS with the proposed approach leads to more accurate reconstruction; and multi-plane fitting becomes more robust applied to the resulting oriented point cloud.

3.1.1 Relationship of Affine Correspondences and Surface Normals

In this section, we discuss the relationship of local affine transformations and surface normals considering perspective camera model.

Assume that a surface point $[x \quad y \quad z]^{\mathrm{T}}$ is observed by two cameras. The camera model can be arbitrary. The projected image points $\mathbf{p}_1 = [u_1 \quad v_1]^{\mathrm{T}}$ and $\mathbf{p}_2 = [u_2 \quad v_2]^{\mathrm{T}}$ are calculated using the 3D $\rightarrow$ 2D projection function $\mathbf{\Pi}_i$ as $[u_i \quad v_i]^{\mathrm{T}} = \mathbf{\Pi_i}(x, y, z)$, where $i \in \{1, 2\}$ denotes the image number. Affine transformation $\mathbf{A}$, mapping the infinitesimally close neighborhood of $\mathbf{p}_1$ to that of $\mathbf{p}_2$, is defined by the Jacobian of the surface projections through $\mathbf{P}_1$ and $\mathbf{P}_2$ as follows:

$$\mathbf{A} = \mathbf{J}_2\mathbf{J}_1^{-1} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}, \tag{3.1}$$

if the surface is written in parametric form. For details, see Appendix B that shows in depth how affine transformation $\mathbf{A}$ can be determined if projective functions $\mathbf{\Pi}_i$ are given.

For perspective cameras, the projection is written as

$$\begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^{\mathrm{T}} = \frac{1}{s_i}\mathbf{P}_i \begin{bmatrix} x & y & z & 1 \end{bmatrix}^{\mathrm{T}},$$

where

$$\mathbf{P}_i = \begin{bmatrix} p_{i,11} & p_{i,12} & p_{i,13} & p_{i,14} \\ p_{i,21} & p_{i,22} & p_{i,23} & p_{i,24} \\ p_{i,31} & p_{i,32} & p_{i,33} & p_{i,34} \end{bmatrix} \quad i \in \{1, 2\}$$

is the projection matrix, $s_i = p_{i,31}x + p_{i,32}y + p_{i,33}z + p_{i,34}$ is the projective depth, u_i and v_i are the projected coordinates in the ith image, and $[x \quad y \quad z \quad 1]^{\mathrm{T}}$ is the homogeneous 3D point. The gradients of the projection formulas w.r.t. to the spatial directions are as follows:

$$\begin{array}{llll} \frac{\partial u_i}{\partial x} = \frac{1}{s_i}(p_{i,11} + u_i p_{i,31}), & \frac{\partial u_i}{\partial y} = \frac{1}{s_i}(p_{i,12} + u_i p_{i,32}), \\ \frac{\partial u_i}{\partial z} = \frac{1}{s_i}(p_{i,13} + u_i p_{i,33}), & \frac{\partial v_i}{\partial x} = \frac{1}{s_i}(p_{i,21} + v_i p_{i,31}), \\ \frac{\partial v_i}{\partial y} = \frac{1}{s_i}(p_{i,22} + v_i p_{i,32}), & \frac{\partial v_i}{\partial z} = \frac{1}{s_i}(p_{i,23} + v_i p_{i,33}). \end{array}$$

Therefore, the gradient vectors are written as

$$\nabla\mathbf{\Pi}_{i,u} = \frac{1}{s_i}\begin{bmatrix} p_{i,11} + u_i p_{i,31} \\ p_{i,12} + u_i p_{i,32} \\ p_{i,13} + u_i p_{i,33} \end{bmatrix}, \quad \nabla\mathbf{\Pi}_{i,v} = \frac{1}{s_i}\begin{bmatrix} p_{i,21} + v_i p_{i,31} \\ p_{i,22} + v_i p_{i,32} \\ p_{i,23} + v_i p_{i,33} \end{bmatrix}.$$

By building the Jacobians using gradient vectors $\nabla\mathbf{\Pi}_{i,u}$ and $\nabla\mathbf{\Pi}_{i,v}$ and multiplying them, local affine transformation $\mathbf{A}$ becomes

$$\mathbf{A} = \mathbf{J}_2\mathbf{J}_1^{-1} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} = \frac{1}{\alpha \mathbf{n}^{\mathrm{T}}\mathbf{w}_5} \begin{bmatrix} \mathbf{n}^{\mathrm{T}}\mathbf{w}_1 & \mathbf{n}^{\mathrm{T}}\mathbf{w}_2 \\ \mathbf{n}^{\mathrm{T}}\mathbf{w}_3 & \mathbf{n}^{\mathrm{T}}\mathbf{w}_4 \end{bmatrix}, \tag{3.2}$$

where $\alpha = s_1/s_2$ is the ratio of the projective depths in the two images and

$$\begin{aligned} \mathbf{w}_1 &= s_1 s_2 \nabla\mathbf{\Pi}_{1,v} \times \nabla\mathbf{\Pi}_{2,u}, \quad \mathbf{w}_2 = s_1 s_2 \nabla\mathbf{\Pi}_{2,u} \times \nabla\mathbf{\Pi}_{1,u}, \\ \mathbf{w}_3 &= s_1 s_2 \nabla\mathbf{\Pi}_{1,v} \times \nabla\mathbf{\Pi}_{2,v}, \quad \mathbf{w}_4 = s_1 s_2 \nabla\mathbf{\Pi}_{2,v} \times \nabla\mathbf{\Pi}_{1,u}, \\ \mathbf{w}_5 &= s_1 s_1 \nabla\mathbf{\Pi}_{1,v} \times \nabla\mathbf{\Pi}_{1,u}. \end{aligned} \tag{3.3}$$

Eq. 3.2 determines the relationship of surface normals and local affine transformations for the perspective camera model. We will use this relationship to define the optimal solvers for both the two- and multi-view cases.

Note that if the projective depth s_i is unknown, but the upper left 3×3 submatrices of the projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ are known, the gradient vectors can be calculated up to an unknown scale – this scale is the multiplicative inverse of the projective depth s_i. Also note that vectors $\mathbf{w}_1, ..., \mathbf{w}_4$ are scaled by $s_1 s_2$ whilst $\mathbf{w}_5$ by $s_1 s_1$. Therefore, the surface normal is independent of the translation between the two cameras since the last columns of the projection matrices are the product of the intrinsic parameters and the translation.

3.1.2 Multi-View Optimal Surface Normals

Optimal solvers are proposed for the stereo and multi-view cases in this subsection. Then we propose a robust algorithm minimizing the effect of the outliers.

Stereo Case. In this subsection, we show that a surface normal $\mathbf{n} = [n_x \quad n_y \quad n_z]^{\mathrm{T}}$ can optimally be estimated, in the least squares sense, exploiting a local affinity. Suppose that an affine correspondence $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{A})$ obtained by e.g. an affine-covariant feature detector is given in two images. The optimization problem is written by reformulating Eq. 3.2 as follows:

$$\arg\min_{\mathbf{n}} \sum_{k=1}^{4} \left(\frac{\mathbf{n}^{\mathrm{T}}\mathbf{w}_k}{\mathbf{n}^{\mathrm{T}}\mathbf{w}_5} - a_k \right)^2, \tag{3.4}$$

where the only unknowns are the coordinates of $\mathbf{n}$. Note that the four equations can be linearized multiplying each by $\mathbf{n}^{\mathrm{T}}\mathbf{w}_5$, however, the linearization distorts the original signal-noise ratio leading to noise-sensitive estimates.

Such kind of optimization problems are usually solved by Lagrange multipliers, however, in the current case the derivatives would be difficult to solve. Therefore, we exploit that the length of the surface normal can be arbitrary and consider constraint

$$n_x + n_y + n_z = 1$$

which leads to $\mathbf{n} = [n_x \quad n_y \quad 1 - n_x - n_y]^{\mathrm{T}}$. Applying this constraint, Eg. 3.4 becomes

$$\arg\min_{\mathbf{m}} \sum_{k=1}^{4} \left(\frac{\mathbf{m}^{\mathrm{T}}\mathbf{q}_k + w_{k,z}}{\mathbf{m}^{\mathrm{T}}\mathbf{q}_5 + w_{5,z}} - a_k \right)^2, \tag{3.5}$$

where $\mathbf{m} = [n_x \quad n_y]^{\mathrm{T}}$, $\mathbf{q}_i = [w_{i,x} - w_{i,z} \quad w_{i,y} - w_{i,z}]^{\mathrm{T}}$ and $w_{i,x}$, $w_{i,y}$, $w_{i,z}$ are the x, y, z coordinates of $\mathbf{w}_i$, respectively. Note that setting a coordinate to be equal to a predefined value, e.g. $n_x = 1$, is not preferred since the case when $n_x \approx 0$ would be degenerate and should be handled.

The optimal solution, in the least squares sense, is where the derivative w.r.t. $\mathbf{m}$ equals to zero:

$$\sum_{k=1}^{4} \beta_k \mathbf{r}_k = 0,$$

where

$$\begin{aligned} \beta_k &= \frac{\mathbf{m}^{\mathrm{T}}\mathbf{q}_k + w_{k,z}}{\mathbf{m}^{\mathrm{T}}\mathbf{q}_5 + w_{5,z}} - a_k, \\ \mathbf{r}_k &= \frac{(\mathbf{m}^{\mathrm{T}}\mathbf{q}_5 + w_{5,z})\mathbf{q}_k - (\mathbf{m}^{\mathrm{T}}\mathbf{q}_k + w_{k,z})\mathbf{q}_5}{(\mathbf{m}^{\mathrm{T}}\mathbf{q}_5 + w_{5,z})^2}. \end{aligned}$$

Note that $\mathbf{r}_k$ is a two-dimensional vector consisting of the expressions regarding to both coordinates of vector $\mathbf{m}$. After elementary modifications, including the multiplication by the denominator, the following formula is obtained:

$$\sum_{k=1}^{4} \mathbf{s}_k \begin{bmatrix} \mathbf{m}^{\mathrm{T}}(\mathbf{q}_5 q_{k,x} - \mathbf{q}_i q_{5,x}) + w_{5,z} q_{k,x} - w_{k,z} q_{5,x} \\ \mathbf{m}^{\mathrm{T}}(\mathbf{q}_5 q_{k,y} - \mathbf{q}_i q_{5,y}) + w_{5,z} q_{k,y} - w_{k,z} q_{5,y} \end{bmatrix} = 0,$$

where $\mathbf{s} = \mathbf{m}^{\mathrm{T}}(\mathbf{q}_k - \mathbf{q}_k\mathbf{q}_5) + w_{k,z} - a_k q_{5,z}$. Replacing $\mathbf{m}$ with its coordinates, the equation becomes

$$\sum_{k=1}^{4} (\Omega_k n_x + \Psi_k n_y + \Gamma_k) \begin{bmatrix} \Omega_k^1 n_x + \Psi_k^1 n_y + \Gamma_k^1 \\ \Omega_k^2 n_x + \Psi_k^2 n_y + \Gamma_k^2 \end{bmatrix} = 0,$$

where

$$\begin{aligned} \Omega_k &= q_{k,x} - q_{5,x} a_k, & \Psi_k &= q_{k,y} - q_{5,y} a_k, \\ \Gamma_k &= w_{k,z} - a_k w_{5,z}, & \Omega_{k,1} &= 0, \\ \Psi_{k,1} &= q_{5,y} q_{k,x} - q_{k,y} q_{5,x}, & \Gamma_{k,1} &= w_{5,z} q_{k,z} - w_{k,z} g_{5,x}, \\ \Omega_{k,2} &= q_{5,x} q_{k,y} - q_{k,x} q_{5,y}, & \Psi_{k,2} &= 0, \\ \Gamma_{k,2} &= w_{5,z} q_{k,y} - w_{k,z} q_{5,y}. \end{aligned}$$

The rows of the vector equation yield two quadratic curves written in implicit form as

$$\begin{aligned} &\sum_{k=1}^{4} A_{k,1} n_x^2 + B_{k,1} n_y^2 + C_{k,1} n_x n_y + D_{k,1} n_x + E_{k,1} n_y + F_{k,1} = 0, \\ &\sum_{k=1}^{4} A_{k,2} n_x^2 + B_{k,2} n_y^2 + C_{k,2} n_x n_y + D_{k,2} n_x + E_{k,2} n_y + F_{k,2} = 0, \end{aligned}$$

where

$$\begin{aligned} A_{k,l} &= \Omega_k \Omega_{k,l}, & B_{k,l} &= \Psi_k \Psi_{k,l}, \\ C_{k,l} &= \Omega_{k,l} \Psi_k + \Psi_{k,l} \Omega_k, & D_{k,l} &= \Omega_{k,l} \Gamma_k + \Gamma_{k,l} \Omega_{k,}, \\ E_{k,l} &= \Psi_{k,l} \Gamma_k + \Psi_{k,l} \Omega_k, & F_{k,l} &= \Gamma_k \Gamma_{k,l}, \end{aligned}$$

for $l \in \{1, 2\}$. Since $\Omega_{k,1} = \Psi_{k,2} = 0$ coefficients $A_{k,1} = B_{k,2} = 0$.

The summation can be eliminated from the equation by adding up the coefficients separately, e.g. $\hat{B}_1 = \sum_{k=1}^{k} B_{k,1}$. Thus the resulting curves are as follows:

$$\hat{B}_1 n_y^2 + \hat{C}_1 n_x n_y + \hat{D}_1 n_x + \hat{E}_1 n_y + \hat{F}_1 = 0, \tag{3.6}$$

$$\hat{A}_2 n_x^2 + \hat{C}_2 n_x n_y + \hat{D}_2 n_x + \hat{E}_2 n_y + \hat{F}_2 = 0. \tag{3.7}$$

This polynomial system is straightforward to solve, thus applying a sophisticated polynomial solver, e.g. Groebner-basis [61], would be an overshot. Instead, we express parameter n_y from Eq. 3.7 as

$$n_y = -\frac{\hat{A}_2 n_x^2 + \hat{D}_2 n_x + \hat{F}_2}{\hat{C}_2 n_x + \hat{E}_2}. \tag{3.8}$$

Substituting Eq. 3.8 into Eq. 3.6 and multiplying by the denominator lead to

$$\begin{aligned}\hat{B}_1(\hat{A}_2 n_x^2 + \hat{D}_2 n_x + \hat{F}_2)^2 - \hat{C}_1(\hat{A}_2 n_x^2 + \hat{D}_2 n_x + \hat{F}_2)(\hat{C}_2 n_x + \hat{E}_2) + \\ \hat{D}_1 x(\hat{C}_2 n_x + \hat{E}_2)^2 - \hat{E}_1(\hat{A}_2 n_x^2 + \hat{D}_2 n_x + \hat{F}_2)(\hat{C}_2 n_x + \hat{E}_2) + \\ \hat{F}_1(\hat{C}_2 n_x + \hat{E}_2)^2 = 0.\end{aligned}$$

The coefficients regarding each monomial (n_x^4, n_x^3, n_x^2, n_x^1, and n_x^0) are as follows:

$$\begin{aligned}
n_x^4 &: \ \hat{B}_1\hat{A}_2^2 - \hat{C}_1\hat{A}_2\hat{C}_2, \\
n_x^3 &: \ 2\hat{B}_1\hat{A}_2\hat{D}_2 - \hat{C}_1\hat{A}_2\hat{E}_2 - \hat{C}_1\hat{D}_2\hat{C}_2 + \hat{D}_1\hat{C}_2^2 - \hat{E}_1\hat{A}_2\hat{C}_2, \\
n_x^2 &: \ \hat{B}_1\hat{D}_2^2 + 2\hat{B}_1\hat{A}_2\hat{F}_2 - \hat{C}_1\hat{D}_2\hat{E}_2 - \hat{C}_1\hat{F}_2\hat{C}_2 + \\
&\quad \ 2\hat{D}_1\hat{C}_2\hat{E}_2 - \hat{E}_1\hat{A}_2\hat{E}_2 - \hat{E}_1\hat{D}_2\hat{C}_2 + \hat{F}_1\hat{C}_1^2, \\
n_x^1 &: \ 2\hat{B}_1\hat{D}_2\hat{F}_2 - \hat{C}_1\hat{F}_2\hat{E}_2 + \hat{D}_1\hat{E}_2^2 - \hat{E}_1\hat{D}_2\hat{E}_2 - \\
&\quad \ \hat{E}_1\hat{F}_2\hat{C}_2 + 2\hat{F}_1\hat{C}_2\hat{E}_2, \\
n_x^0 &: \ \hat{B}_1\hat{F}_2^2 - \hat{E}_1\hat{F}_2\hat{E}_2 + \hat{F}_1\hat{E}_2^2.
\end{aligned}$$

This fourth-order polynomial equation can be solved by any polynomial solver toolbox, e.g. Matlab *roots* or OpenCV *solvePoly* methods. Coordinate n_y is then obtained using Eq. 3.8 and finally, $n_z = 1 - n_x - n_y$. To select the best out of the candidate real roots, we choose the one minimizing Eq. 3.4.

Summarizing this subsection, the coordinates of the surface normal can optimally be estimated in closed-form as the roots of a fourth-order polynomial without linearizing the original equations.

Multi-View Case. Given a sequence of points in $N \geq 2$ images with local affinities between every pair – this is a realistic assumption since affine-covariant feature detectors estimate Jacobian $\mathbf{J}$ for each image independently, thus affinity $\mathbf{A}_{ij}$ mapping from the ith to jth images is calculated as $\mathbf{J}_j\mathbf{J}_i^{-1}$. Extending Eq. 3.4 to more image pairs, the optimization problem becomes

$$\arg\min_{\mathbf{n}} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{k=1}^{4} \left(\frac{\mathbf{n}^{\mathrm{T}}\mathbf{w}_{ij,k}}{\mathbf{n}^{\mathrm{T}}\mathbf{w}_{ij,5}} - a_{ij,k} \right)^2, \tag{3.9}$$

where each vector $\mathbf{w}_{ij}$ is calculated similarly to Eq. 3.3 using the coordinates in the ith and jth images. It can be seen that the inner summation leads to two quadratic curves (Eqs. 3.6, 3.7), and the outer two is basically the summation of these curves

over the possible view pairs:

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \hat{B}_{ij,1}n_y^2 + \hat{C}_{ij,1}n_x n_y + \hat{D}_{ij,1}n_x + \hat{E}_{ij,1}n_y + \hat{F}_{ij,1} = 0, \tag{3.10}$$

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \hat{A}_{ij,2}n_x^2 + \hat{C}_{ij,2}n_x n_y + \hat{D}_{ij,2}n_x + \hat{E}_{ij,2}n_y + \hat{F}_{ij,2} = 0. \tag{3.11}$$

These two equations can be formulated as

$$\widehat{B}_1 y^2 + \widehat{C}_1 n_x n_y + \widehat{D}_1 n_x + \widehat{E}_1 n_y + \widehat{F}_1 \quad = 0, \tag{3.12}$$

$$\widehat{B}_2 y^2 + \widehat{C}_2 n_x n_y + \widehat{D}_2 n_x + \widehat{E}_2 n_y + \widehat{F}_2 \quad = 0, \tag{3.13}$$

where

$$\widehat{S}_k = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \hat{S}_{ij,k} \quad k \in \{1,2\}, \quad S \in \{B,C,D,E,F\}.$$

Thus the solution is given as the intersection of the summed curves (Eqs. 3.12, 3.13) in a fairly similar manner to that of the two-view case. Note that the normalization of the coefficients is necessary to avoid numerical instability. Another note that the missing data problem, i.e. when information is not given for every image pair, can be resolved by introducing weight q_{ij} into Eq. 3.9. Weight q_{ij} is zero if there is no correspondence between the ith and jth views and one otherwise.

Robust Estimation. Reflecting the fact that the local affinities might be contaminated by noise and contain outliers, we propose a robust estimation process here as an iteratively re-weighted least squares algorithm [62]. First, all weights are set to 1.0 and the indicated normal is computed applying the multi-view algorithm. Then, in each step of the alternation, the weights for the view pairs are re-calculated on the basis of the error of the estimated normal.

Each weight q_{ij} regarding the ith and jth views affects the indicated quadratic curves (the inner part of Eqs. 3.10, 3.11) by multiplying them as follows:

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} q_{ij}(\hat{B}_{ij,1}n_y^2 + \hat{C}_{ij,1}n_x n_y + \hat{D}_{ij,1}n_x + \hat{E}_{ij,1}n_y + \hat{F}_{ij,1}) = 0, \tag{3.14}$$

$$\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} q_{ij}(\hat{A}_{ij,2}n_x^2 + \hat{C}_{ij,2}n_x n_y + \hat{D}_{ij,2}n_x + \hat{E}_{ij,2}n_y + \hat{F}_{ij,2}) = 0. \tag{3.15}$$

3.1.3 Experimental Results

In this subsection, the performance of the proposed method is evaluated both on synthesized and real world tests.

Synthesized Tests. In order to test the proposed method in a fully controlled environment, N cameras were generated by their projection matrices looking towards the origin, each located in a random surface point of a 5-radius sphere. Then a random 3D oriented point, at most one unit far from the origin and with random normal, was projected onto the cameras. See the right plot of Fig. 3.2. The local

affine transformation was calculated from the ground truth surface normal using Eq. 3.2. Finally, zero-mean Gaussian noise with σ standard deviation is added to both the point locations and affine parameters. The reported results are computed as the mean of 500 runs for each test case.

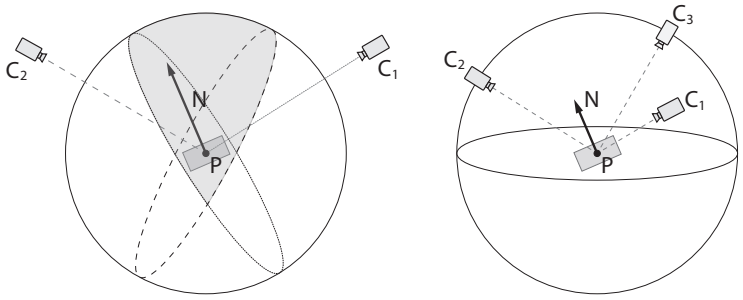


FIGURE 3.2: **(Left)** The proposed geometric constraint demonstrated by two views. A hemisphere is selected by each camera (denoted by different dashed lines) around the observed point. The surface normal must be in the intersubsection of these hemispheres. **(Right)** The set up for the synthesized tests. The cameras are put in a random point of a sphere.

The competitor algorithms are the two-view optimal method proposed in this section, the techniques of Baráth et al. [34] and Kevin Köser [5]. Since they are 2-view methods, the multi-view results are computed as the mean of the normals calculated for every possible view pair.

Figs. 3.3(a), 3.3(b), 3.3(c), and 3.3(d) plot the angular error (in degrees) as a function of the noise σ for 3, 5, 10 and 25 views, respectively. It can be seen that the proposed method outperforms the competitor algorithms.

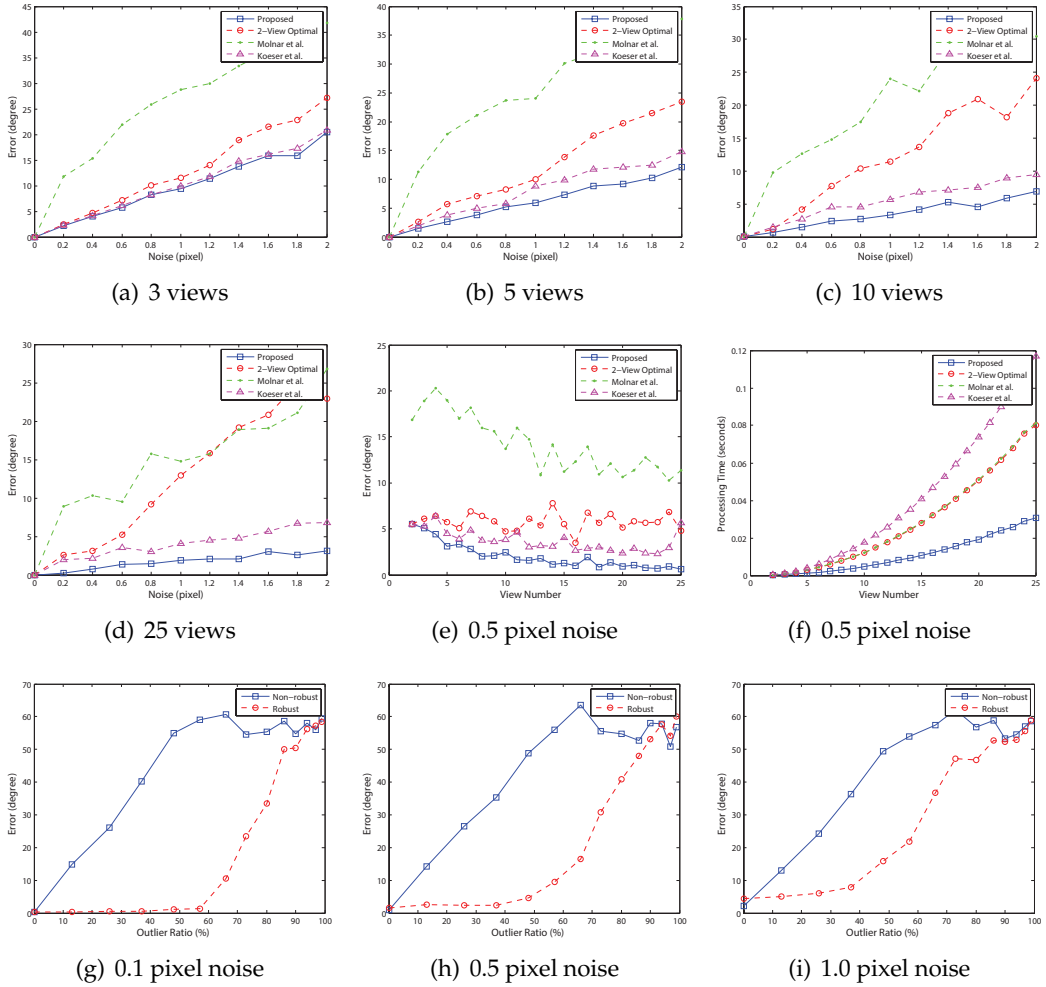
Fig. 3.3(e) shows the angular error as a function of the view number with fixed $\sigma = 0.5$ pixel noise. It can be seen that the proposed method is consistent - the more samples are given, the lower error is achieved -, and converges to the ground truth normal faster than the other methods.

Figs. 3.3(g), 3.3(h) and 3.3(i) compare the robust version of the proposed algorithm to the original one with σ set to 0.1, 0.5 and 1.0 pixels, respectively. For these tests $I \in [2, 15]$ views were generated, and $15 - I$ outliers (random point correspondences and affinities) were added. For instance, if $I = 10$, i.e. 10 inlier and 5 outlier views are given, the outlier percentage is calculated as

$$1 - \frac{\binom{10}{2}}{\binom{15}{2}} \approx 0.57. \tag{3.16}$$

In the figures, the horizontal axis reports the outlier ratio and the vertical one shows the mean angular error of the results. It can be seen that the robust version of the proposed algorithm is able to fully overcome at most $50 - 60\%$ outlier ratio, and significantly reduces the error even for higher noise level.

The mean processing times of the methods are reported in Fig. 3.3(f) plotted as the function of the view number. Due to the pair-wise parameter calculation, the time demands of all methods show a quadratic trend, however, the proposed one is significantly faster for more views than the competitors, e.g. processing 25 views lasts ≈ 0.03 seconds in Matlab.



(a) 3 views (b) 5 views (c) 10 views

(d) 25 views (e) 0.5 pixel noise (f) 0.5 pixel noise

(g) 0.1 pixel noise (h) 0.5 pixel noise (i) 1.0 pixel noise

FIGURE 3.3: Synthesized tests comparing normal estimators. (a-d) report the angular error plotted as the function of noise σ with different number of views; (e) and (f) are the error and the processing time w.r.t. increasing view number; (g-i) show the accuracy of the non-robust and robust algorithms w.r.t. increasing noise σ on different outlier levels.

Real World Tests. To test the proposed method on real world data we used the publicly available benchmarking datasets of Stretcha et al. [63], Pusztai et al. [64] and ETH3D [65]. The dataset[1] of [63] consists of several images of size 3072×2048 of buildings. Both the intrinsic and extrinsic parameters are given for all images, the dense point cloud for each scene is obtained using a LiDAR sensor. The images of [64] are captured by a turn-table equipment, the cameras are calibrated and the ground truth point clouds are estimated using a structured light scanner. ETH3D[2] contains image sequences captured by both HD and mobile cameras and 3D point clouds obtained by laser scanner. For all datasets, the ground truth surface normals are estimated using the dense point clouds by fitting a paraboloid to the neighborhood of each point.

The competitor algorithms are FNE [34], the method of Kevin Köser [5], the two-view optimal method (2-Opt), the proposed multi-view algorithm (MV-Opt) and its robust variant (Robust MV-Opt). Table 3.1 reports the results of the methods on each test scene (rows). Every block, consisting of three columns, shows the average

[1] Available at http://cvlabwww.epfl.ch/data/multiview/denseMVS.html
[2] Available at https://www.eth3d.net/datasets#high-res-multi-view

TABLE 3.1: Surface normal estimation. For each method, the mean (AVG) angular error in degrees, the standard deviation, (σ) and the processing time (T) given in milliseconds are reported. Tests (rows): (1) fountain-P11, (2) Herz-Jesus-P8, (3) Herz-Jesus-P25 are from [63], (4) books1, (5) books2, (6) bag are from [64] and, finally, (7) courtyard (8) delivery area (9) pipes (10) playground, (11) relief and (12) terrace are from ETH3D [65].

	FNE			Köser			2-Opt			MV-Opt			**Robust MV-Opt**		
	AVG	σ	T	AVG	σ	T	AVG	σ	T	AVG	σ	T	AVG	σ	T
(1)	13.2	15.3	0.08	13.3	20.2	0.21	13.3	15.2	0.04	13.1	15.1	**0.01**	**5.7**	**5.4**	0.44
(2)	42.1	24.6	0.40	24.4	18.9	1.10	24.4	18.8	0.02	24.2	18.8	**0.01**	**3.4**	**0.6**	0.05
(3)	22.4	18.9	0.30	22.3	18.6	0.80	22.3	18.6	0.10	22.3	18.5	**0.03**	**9.6**	**12.2**	0.10
(4)	10.6	13.6	0.10	10.6	13.2	0.30	10.6	13.2	0.06	10.4	13.2	**0.02**	**5.9**	**7.8**	0.05
(5)	15.4	20.9	0.10	15.6	20.8	0.20	15.5	20.8	0.05	15.2	20.8	**0.01**	**11.4**	**19.7**	0.04
(6)	25.1	16.1	0.05	24.6	16.0	0.10	24.6	15.9	0.03	24.3	15.7	**0.01**	**18.9**	**12.1**	0.07
(7)	24.3	19.8	0.06	24.4	19.8	0.15	24.4	19.8	0.03	24.2	19.4	**0.01**	**12.3**	**11.0**	0.22
(8)	36.1	25.0	0.04	35.6	25.2	0.10	35.6	25.2	0.02	35.8	25.2	**0.01**	**18.3**	**19.0**	0.72
(9)	39.9	24.6	0.02	40.5	24.7	0.05	40.5	24.7	0.01	40.1	24.6	**0.01**	**20.3**	**21.8**	0.62
(10)	49.4	24.1	0.02	48.6	24.2	0.05	48.6	24.2	0.01	48.3	24.2	**0.01**	**36.0**	**24.7**	0.71
(11)	35.4	20.4	0.05	35.1	20.3	0.14	35.1	20.3	0.03	35.1	20.2	**0.01**	**29.4**	**16.8**	0.45
(12)	52.6	23.9	0.02	55.3	24.0	0.05	55.3	24.0	0.01	54.3	23.3	**0.01**	**39.2**	**22.9**	0.68
AVG	30.5	20.6	0.10	29.2	20.5	0.27	29.2	20.1	0.03	28.9	19.9	**0.01**	**17.5**	**14.5**	0.35
MED	30.3	20.7	0.06	24.5	20.3	0.15	24.5	20.1	0.03	24.3	19.8	**0.01**	**15.3**	**14.5**	0.33

(AVG) angular errors, their standard deviation (σ), and the mean processing time of the point-wise computation in milliseconds. The mean and median results on all scenes are reported in the last two rows.

It can be seen that the optimal method without robust estimation (MV-Opt) is more accurate except two cases and, on average, one order of magnitude faster than the competitor algorithms. Even though its errors are the lowest, the difference is not significant, approx. 0.3 degrees. Since the synthesized tests reported larger difference, this means that the outlier percentage is high. Overcoming this problem, the robust algorithm (Robust MV-Opt) obtains twice as accurate surface normals with similar speed as the competitor methods. In Fig. 3.5, each row shows the result on a test sequence. The first column is an image from the sequence. The second and third ones show the reconstructed oriented point cloud rendered from different viewpoints. In practice, the robust algorithm rejects $\approx 60\%$ of the detected points. For the kept ones, the ratio of the view-pairs considered as inlier is $\approx 70\%$ on average.

Application: Improving PMVS2. In this subsection, we show that combining the proposed normal estimation technique with the state-of-the-art PMVS2 [66] structure-from-motion algorithm is beneficial and leads to superior results. PMVS2 has an initial seed point generation step applied before the dense reconstruction. During this step, it detects feature points and estimates surface normals applying an iterative strategy which minimizes a photo-consistency-based cost function. To demonstrate the accuracy of the proposed method, we replaced this normal estimation step with the proposed one.

Each row of Table 3.2 is a test sequence. The first block, consisting of four columns, shows the error of the original PMVS2 w.r.t. the ground truth point cloud obtained by a laser scanner. The second block reports the results of PMVS2 combined with the proposed approach. The reported properties are: the mean error of the point cloud ($\mathcal{E}_{\mathbf{p}}$, Eucledian distance), its standard deviation ($\sigma_{\mathbf{p}}$), the angular error of the normals ($\mathcal{E}_{\mathbf{n}}$, in degrees) and, finally, its standard deviation ($\sigma_{\mathbf{n}}$). It can be seen that combining the proposed estimation technique with PMVS2 leads to more

TABLE 3.2: The accuracy of the oriented point clouds obtained by applying the original PMVS2 and the one combined with the proposed normal estimation. $\mathcal{E}_\mathbf{p}$ is the mean distance of the reconstructed and the ground truth points and $\sigma_\mathbf{p}$ is the standard deviation. $\mathcal{E}_\mathbf{n}$ is the mean angular error (in degrees) of the obtained normals w.r.t. the ground truth ones, $\sigma_\mathbf{n}$ is the standard deviation of the errors. Tests (rows): (1) fountain-P11, (2) Herz-Jesus-P8, (3) Herz-Jesus-P25 are from [63], (4) books1, (5) books2, (6) bag are from [64].

	PMVS2				**PMVS2 + Robust MV-Opt**			
	$\mathcal{E}_\mathbf{p}$	σ_p	$\mathcal{E}_\mathbf{n}$	$\sigma_\mathbf{n}$	$\mathcal{E}_\mathbf{p}$	σ_p	$\mathcal{E}_\mathbf{n}$	$\sigma_\mathbf{n}$
(1)	0.013	0.015	25.6	19.1	**0.008**	**0.011**	**23.1**	**18.1**
(2)	0.077	0.052	33.2	22.7	**0.013**	**0.018**	**24.4**	**18.7**
(3)	0.023	0.028	27.6	19.8	**0.016**	**0.022**	**23.7**	**17.6**
(4)	**0.031**	**0.048**	**27.8**	**19.7**	0.032	0.051	28.1	**19.7**
(5)	0.057	0.063	32.0	20.6	**0.053**	**0.060**	**31.3**	**20.1**
(6)	0.050	**0.050**	31.8	18.5	**0.049**	**0.050**	**31.5**	**18.3**
AVG	0.042	0.043	29.7	20.1	**0.029**	**0.035**	**27.0**	**18.8**
MED	0.041	0.049	29.8	19.8	**0.024**	**0.036**	**26.2**	**18.5**

TABLE 3.3: Multiple plane fitting to oriented (1PT) and non-oriented (3PT) point clouds using PEARL [13] algorithm. The mean misclassification error (ME) in percentage is reported for each test case (columns; corresponds to Fig. 3.4). The properties of each scene are in Table 3.4.

		(a)	(b)	(c)	(d)	(e)	AVG	MED
1PT	ME (%)	**12.0**	**23.0**	**37.0**	39.8	**10.9**	**24.5**	**23.0**
3PT		16.4	31.6	40.2	**36.5**	11.8	27.3	31.6

accurate reconstructions both in terms of the quality of the dense point cloud and that of the surface normals.

Application: Plane fitting. In this subsection, we demonstrate an application as the fitting of planes to an oriented point cloud obtained by the proposed technique. We took several photos of buildings having large flat walls, then points are detected by ASIFT and the whole system is calibrated using OpenMVG [67] with a priori intrinsic camera parameters. Points are assigned manually to planes or the outlier class, i.e. points not belonging to any dominant planes, to have a ground truth clustering. The properties of each scene are written in Table 3.4. We chose PEARL [13] for multi-model fitting since it has publicly available source code and can be considered as a state-of-the-art technique.

Table 3.3 reports the clustering results of each column in Fig. 3.4. The first row of the table denotes the test case. The second and third rows show the results of PEARL generating the initial model-hypotheses exploiting the surface normals (1PT) or not

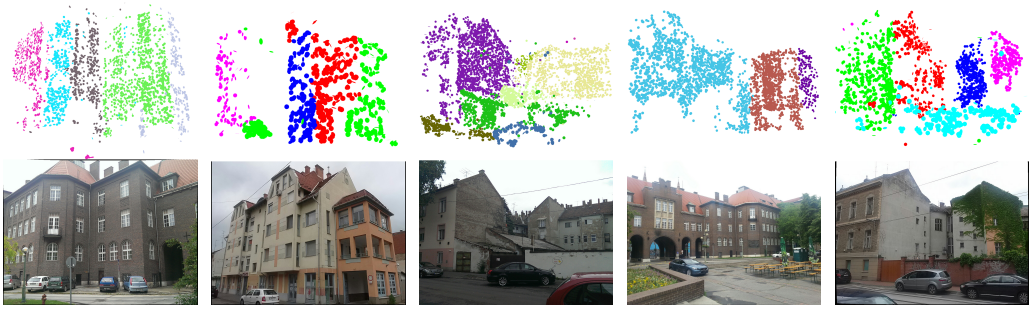
TABLE 3.4: The properties of multi-plane fitting scenes. The point number (1st row), plane number (2nd row) and outlier percentage (3rd row) are reported for each test case (columns, corresponds to Fig. 3.4). The clustering results are in Table 3.3.

	(a)	(b)	(c)	(d)	(e)
Point #	3 257	2 105	4 391	2 758	1 749
Plane #	6	6	8	6	5
Outlier %	16%	15%	31%	11%	21%

(3PT), respectively. The error is the misclassification error (ME), i.e. the ratio of the misclassified points:

$$\text{ME} = \frac{\#\text{Misclassified Points}}{\#\text{Points}}$$

It can be seen that applying PEARL to oriented point clouds leads to the most accurate results in all but one case.



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

FIGURE 3.4: Multi-plane fitting results. First row shows obtained 3D point cloud. Colors denote planes. Second row consists of an image of each sequence.

3.1.4 Summary

In this section, we propose an optimal method for two-view surface normal estimation, then it is extended to multiple views. The method estimates a normal for each affine correspondence individually, and its robust version is able to deal with approx. 60-70% outlier ratio. It is superior to the state-of-the-art both in synthesized tests and on publicly available real datasets. Comparing with other components of a structure-from-motion pipeline, the technique has negligible time demand despite the pair-wise term since the coefficient computation is efficient and only the obtained polynomial equation has to be solved. Usually limited number of views are given, at most $10-20$, where a point can be tracked. Therefore, it is very rare to have problems for which the computation lasts even for a few milliseconds. In our C++ implementation the processing time of 100 views is ≈ 7 milliseconds. However, aiming at real time capability for thousands of point sequences, both the coefficient calculation for each view and the processing of each point sequence can be parallelized and implemented on GPU straightforwardly. Exploiting the obtained oriented point cloud in PMVS or multi-plane fitting applications is beneficial and leads to significant improvement in accuracy as it is demonstrated experimentally.

3.2 Point-wise Homography Estimation

Understanding the surrounding environment is an important goal of computer vision. This problem can be approached from several directions: as the urban scenes and most of the man-made objects usually consist of planes or planar-like surfaces, one of the most popular ways is to segment the observed scene into planar regions. There is a high number of applications exploiting this information such as 3D reconstruction [68], [69], camera calibration [70]–[72], augmented reality [73], robot vision [74], indoor navigation [75]. A good example for such an indoor environment is the office where the dominant objects are tables, chairs, partition walls, or an ordinary

home also contains several planar surfaces. As a part of complex pipelines, homography estimation is frequently used for detecting scenes which are degenerate for fundamental matrix estimation.

In stereo vision, a plane correspondence between two images is described by the so-called homography matrix which is a $P^2 \rightarrow P^2$ perspective transformation. It can be estimated in several ways as it is discussed by [76] in deep. The most popular algorithms are based on point correspondences such as the well-known normalized 4-point algorithm [43] or three correspondences are also enough if the fundamental matrix is estimated beforehand [43]. However, more complex input data can also be exploited, e.g. line [43], region [77], contour [78], conic [79], [80], or affine correspondences [5].

This section addresses the problem of point-wise homography estimation from a set of point correspondences satisfying the epipolar constraint and the related local affine transformations. Nowadays, the acquirement of local affinities are not a real challenge. Beside the well-known affine-covariant detectors such as MSER, Hessian-Affine or Harris-Affine [3], some approaches based on view-synthesizing was recently proposed. Such detectors are ASIFT [2] and MODS [46]. These methods warp the original images by an affine transformation creating a synthetic view and apply a feature detector to the transformed images. The local affinity related to a point pair is given as the multiplication of two transformations: the affinity regarding to the synthetic-view and the one which the applied detector obtains. It is shown in this section that the usage of these affine-covariant detectors creates a natural way to point-wise homography estimation. As a side-effect, we compare them and select the most suitable one for homography estimation.

The contributions are: (i) A novel approach is proposed to estimate a planar homography from a single affine correspondence. It is applicable to correspondences satisfying the epipolar constraint. (ii) Affine-covariant feature detectors are compared w.r.t. the accuracy of the estimated homographies and the most precise one is proposed for further usage. (iii) It is shown that the proposed method makes multi-homography estimation superior to the state-of-the-art in terms of accuracy.

Applications. The main benefit of point-wise homography estimation is that robust estimators based on random sampling are significantly faster if the homographies are estimated using fewer points. Table 2.1 shows the required iteration number for RANSAC [1]. Parameters q, p, and n denote the desired probability, the inlier ratio, and the number of correspondences used for the estimation, respectively. For point-wise estimation ($n = 1$) significantly less iterations are required.

The proposed point-wise estimation can be applied for multi-homography estimation as it is shown by [27]. Another possible application is surface normals estimation: as point-wise homography estimation determines the tangent plane of the observed surface, thus its normal can be computed.

3.2.1 Towards Point-wise Homography Estimation

The relationship among a homography, a local affine transformation and the epipolar geometry is shown in this section. Then these are exploited to derive constraints making the homography estimable from only one affine correspondence.

The setup of the problem is visualized in Fig. 3.6. A plane is given with its two projections in an image pair. Denote the locations of the projections in their homogeneous form in the first and second images by $\mathbf{p}_1 = [x_1 \quad y_1 \quad 1]^{\mathrm{T}}$ and $\mathbf{p}_2 =$

$[x_2 \quad y_2 \quad 1]^T$, respectively. It is well-known in projective geometry [43] that the transformation between the images is a $P^2 \rightarrow P^2$ perspective homography. A homography is approximated locally around a point correspondence by an affine transformation which transforms the vicinity of the point in the first image to the neighborhood of the point in the second one.

The input of the proposed method: point locations in stereo images (two coordinates per image), and the affine transformation (four parameters).

Homography Estimation without Fundamental Matrix (HA). We show here that homography $\mathbf{H}$ depends on both the point locations and the related affine transformation considering the fundamental matrix to be unknown.

Direct Linear Transformation. The relationship of points $\mathbf{p}_1$ and $\mathbf{p}_2$ is written as $\mathbf{H}[\mathbf{p}_1 \quad 1]^T \sim [\mathbf{p}_2 \quad 1]^T$ which leads to well-known equations [43]

$$\begin{aligned} x_1 h_{11} + y_1 h_{12} + h_{13} - x_1 x_2 h_{31} - y_1 x_2 h_{32} - x_2 h_{33} = 0, \\ x_1 h_{21} + y_1 h_{22} + h_{23} - x_1 y_2 h_{31} - y_1 y_2 h_{32} - y_2 h_{33} = 0, \end{aligned} \tag{3.17}$$

where h_{ij} is the element of $\mathbf{H}$ in the ith row and jth column. This widely-used technique [43] is called the Direct Linear Transformation (DLT).

Estimation using Affine Transformations. Local affine transformation $\mathbf{A}$ describes the mapping between the infinitely small area around points $\mathbf{p}_1$ and $\mathbf{p}_2$. Suppose that a translation vector $[\Delta x_1 \quad \Delta y_1]^T$ is added to $\mathbf{p}_1$. The translation $[\Delta x_2 \quad \Delta y_2]^T$ which it implies in the second image can be approximated using $\mathbf{A}$ as

$$\begin{bmatrix} \Delta x_2 \\ \Delta y_2 \end{bmatrix} \approx \mathbf{A} \begin{bmatrix} \Delta x_1 \\ \Delta y_1 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}.$$

Consequently, shorter the translation, better the approximation.

The affine parameters can be calculated by the partial derivatives of $\mathbf{H}$ [32] as it is described in detail in Appendix C. The relationship is written as

$$a_1 = \frac{h_{11} - h_{31} x_2}{s}, \; a_2 = \frac{h_{21} - h_{31} y_2}{s}, \; a_3 = \frac{h_{12} - h_{32} x_2}{s}, \; a_4 = \frac{h_{22} - h_{32} y_2}{s}, \tag{3.18}$$

where $j \in \{1, 2\}$, $s = \mathbf{h}_3^T [x_1 \quad y_1 \quad 1]^T$ and $\mathbf{h}_3^T$ is the last row of $\mathbf{H}$. By reformulating Eqs. 3.18, the following homogeneous, linear system of equations is obtained:

$$\begin{aligned} h_{11} - (x_2 + a_1 x_1) h_{31} - a_1 y_1 h_{32} - a_1 h_{33} = 0, \\ h_{12} - (x_2 + a_2 y_1) h_{32} - a_2 x_1 h_{31} - a_2 h_{33} = 0, \\ h_{21} - (y_2 + a_3 x_1) h_{31} - a_3 y_1 h_{32} - a_3 h_{33} = 0, \\ h_{22} - (y_2 + a_4 y_1) h_{32} - a_4 x_1 h_{31} - a_4 h_{33} = 0. \end{aligned} \tag{3.19}$$

Eqs. 3.19 put constraints to all elements of $\mathbf{H}$ but h_{31} and h_{32}. The original DLT equations defined in Eq. 3.17 can be combined with these equations obtaining a homogeneous linear system for estimating all the elements of $\mathbf{H}$. This estimation is called HA (Homography from Affine transformation) here. The optimal solution in the least squares sense is obtained as the eigenvector corresponding to the smallest eigenvalue of matrix $\mathbf{B}^T\mathbf{B}$, where $\mathbf{B}$ is the coefficient matrix of the equation system. Since *each affine correspondence yields* 6 *equations, at least two correspondences are required for HA method*.

Note that Kevin Köser [5] also solved this problem using two correspondences. The advantage of the proposed formulation is its simplicity compared to the method of Köser.

Homography Estimation using the Fundamental Matrix. The relationship of a homography and fundamental matrix is shown in this subsection. It is known in epipolar geometry [43] that

$$[\mathbf{e}_2]_\times \mathbf{H} = \lambda \mathbf{F}, \tag{3.20}$$

where $\mathbf{e}_2 = [e_{x,1} \quad e_{y,1} \quad 1]^\mathrm{T}$, $\mathbf{H}$, $\mathbf{F}$, and λ are the epipole on the second image in its homogeneous form, the homography, the fundamental matrix, and an arbitrary scale, respectively. This formula represents how the fundamental matrix decreases the DoF of the homography estimation. Note that operator $[\mathbf{v}]_\times$ denotes the skew-symmetric cross product matrix of vector $\mathbf{v}$.

Usually, this relationship is exploited to make the homography estimable using three correspondences [43]. In contrast to the common solutions we derive it in a different way since this formulation is easier to use. The last row is linearly dependent as the rank of $[\mathbf{e}_2]_\times$ is two. Thereby, it is removed. The remaining formula is

$$\begin{bmatrix} 0 & -1 & e_{y,1} \\ 1 & 0 & -e_{x,1} \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \lambda \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \end{bmatrix}.$$

As a consequence, the DoF is reduced to three since the elements in the first two rows of $\mathbf{H}$ can be expressed by those in the third one (h_{31}, h_{32}, and h_{33}):

$$\begin{aligned} h_{11} &= e_{x,1}h_{31} + \lambda f_{21}, \quad h_{12} = e_{x,1}h_{32} + \lambda f_{22}, \quad h_{13} = e_{x,1}h_{33} + \lambda f_{23}, \\ h_{21} &= e_{y,1}h_{31} - \lambda f_{11}, \quad h_{22} = e_{y,1}h_{32} - \lambda f_{12}, \quad h_{23} = e_{y,1}h_{33} - \lambda f_{13}. \end{aligned} \tag{3.21}$$

Both the fundamental matrix and homography are determined up to an arbitrary scale. In our algorithms, $\lambda = 1$.

Estimation using Point Correspondences. An inhomogeneous, linear system of equations is formed by substituting Eqs. 3.21 into the basic formula $\mathbf{H}[\mathbf{p}_1 \quad 1]^\mathrm{T} \sim [\mathbf{p}_2 \quad 1]^\mathrm{T}$ applied for DLT algorithm as follows:

$$\begin{aligned} (x_1 e_{x,1} - x_1 x_2)h_{31} + (y_1 e_{x,1} - y_1 x_2)h_{32} + (e_{x,1} - x_2)h_{33} = \\ -x_1 f_{21} - y_1 f_{22} - f_{23}, \end{aligned} \tag{3.22}$$

$$\begin{aligned} (x_1 e_{y,1} - x_1 y_2)h_{31} + (y_1 e_{y,1} - y_1 y_2)h_{32} + (e_{y,1} - y_2)h_{33} = \\ x_1 f_{11} + y_1 f_{12} + f_{13}, \end{aligned} \tag{3.23}$$

Due to the epipolar geometry, the point pair has to lie on the corresponding epipolar lines, thus only one of these equations contain additional information. Even so, exploiting both of them is preferred to minimize the effect of the noise. Using these equations, a DLT-like method can be formed, and this is called 3PT in the rest of the section since *it can be solved if at least three point correspondences are given*.

Estimation using Affine Transformations (HAF). The information provided by fundamental matrix $\mathbf{F}$ can be exploited to decrease the DoF of the affine transformation by substituting Eqs. 3.21 into Eqs. 3.19. To exploit the translation, Eqs. 3.22 and 3.23 are also added to the system. Thus a linear, inhomogeneous system of equations is formed as $\mathbf{C}\mathbf{y} = \mathbf{d}$, where $\mathbf{y} = [h_{31} \quad h_{32} \quad h_{33}]^\mathrm{T}$, $\mathbf{d} = [f_{21} \quad f_{22} \quad -f_{11} \quad -f_{12} \quad -x_1 f_{21} - y_1 f_{22} - f_{23} \quad x_1 f_{11} + y_1 f_{22} - f_{13}]^\mathrm{T}$, and $\mathbf{C}$ are the vector of the unknown

parameters, the inhomogeneous part, and the coefficient matrix, respectively. $\mathbf{C}$ is as follows:

$$\mathbf{C} = \begin{bmatrix} a_1x_1 + x_2 - e_{x,1} & a_1y_1 & a_1 \\ a_2x_1 & a_2y_1 + x_2 - e_{x,1} & a_2 \\ a_3x_1 + y_2 - e_{y,1} & a_3y_1 & a_3 \\ a_4x_1 & a_4y_1 + y_2 - e_{y,1} & a_4 \\ x_1e_{x,1} - x_1x_2 & y_1e_{x,1} - y_1x_2 & e_{x,1} - x_2 \\ x_1e_{y,1} - x_1y_2 & y_1e_{y,1} - y_1y_2 & e_{y,1} - y_2 \end{bmatrix}. \tag{3.24}$$

The optimal solution in the least squares sense is obtained as $\mathbf{y} = \mathbf{C}^\dagger\mathbf{d}$, where $\mathbf{C}^\dagger$ is the Moore-Penrose pseudo-inverse of matrix $\mathbf{C}$. Since the obtained vector $\mathbf{y}$ is the last row of $\mathbf{H}$, the full homography is calculated using the formulas written in Eq. 3.21.

The four equations of the affine transformation (Eqs. 3.19) are linearly dependent, the epipolar geometry determines the rotation and the scale perpendicular to the epipolar line as it is proven in [28]. Theoretically, only two of those have to be kept. The two equations of 3PT (Eqs. 3.22, 3.23) are reduced to one since the point pair has to lie on the related epipolar line [43]. Thus we have three linearly independent equations, two from $\mathbf{A}$ and another one from point correspondence, to estimate the three unknowns. However, in order to minimize the effect of noise, all equations are considered in the proposed method. As a consequence, *the homography can be calculated from a single affine correspondence*.

Note that for HAF algorithm, surface planarity is not required since the method is capable to estimate the tangent plane related to individual spatial surface points.

Generalization to Arbitrary Point Number (HAF). Every additional affine correspondence yields six constraints (rows) in matrix $\mathbf{C}$ and vector $\mathbf{d}$. The optimal solution can be carried out in the same way: $\mathbf{y} = \mathbf{C}^\dagger\mathbf{d}$. To increase the accuracy, data normalization and numerical optimization are required. These are discussed later.

Theoretical Contribution. Since an affine correspondence, consisting of a local affinity and a point correspondence as $(\mathbf{A}, \mathbf{p}_1, \mathbf{p}_2)$, is calculable from the related homography $\mathbf{H}$, and $\mathbf{H}$ is calculable from $(\mathbf{A}, \mathbf{p}_1, \mathbf{p}_2)$ the theoretical contribution is:

Theorem 1 (Equivalence of Affine and Perspective-invariances)**.** *Given an image pair and a set of point correspondences satisfying the epipolar constraint. Affine-invariance is equivalent to perspective-invariance, where the latter one denotes invariance to 2D perspective transformations.*

Degenerate Cases. It is well-known that collinearity is a degenerate case [43] for point-based homography estimation. Even so, this method is based on the full local affine transformation, therefore, the collinearity of the points are not a degenerate case. However, if the plane on which the point(s) lies contains the optical axis of any camera, the homography cannot be calculated. In practice, this is not a real situation since a point which plane contains the optical axis of one of the cameras cannot be observed in both views.

3.2.2 Improvements

The aim of the section is to show possible improvements of the proposed method in order to achieve high accuracy.

Normalization. Data normalization is a crucial point of homography estimation because of numerical instability. The normalizing equations for points and fundamental matrix are written in Section 2.2.

Normalization of Affine Transformation. Obtaining the normalized affinity $\hat{\mathbf{A}}$ is not trivial since it depends on the unknown normalized homography $\hat{\mathbf{H}}$.

The normalizing transformations modify the basic equations written in Eqs. 3.18 as follows:

$$\begin{aligned}
(h_{31}x_1 + h_{32}y_1 + h_{33})\,\hat{a}_{11} &= \tfrac{l_x^2}{l_x^1}h_{11} - \tfrac{l_x^2}{l_x^1}x_2h_{31} \\
(h_{31}x_1 + h_{32}y_1 + h_{33})\,\hat{a}_{2} &= \tfrac{l_x^2}{l_x^1}h_{11} - \tfrac{l_x^2}{l_x^1}x_2h_{31}, \\
(h_{31}x_1 + h_{32}y_1 + h_{33})\,\hat{a}_{3} &= \tfrac{l_x^2}{l_x^1}h_{11} - \tfrac{l_x^2}{l_x^1}x_2h_{31}, \\
(h_{31}x_1 + h_{32}y_1 + h_{33})\,\hat{a}_{4} &= \tfrac{l_x^2}{l_x^1}h_{11} - \tfrac{l_x^2}{l_x^1}x_2h_{31},
\end{aligned}$$

where $l_x^k = \mathbf{T}_{k,11}$, $l_y^k = \mathbf{T}_{k,22}$ $(k \in \{1,2\})$ are the horizontal and vertical scales of the kth normalizing transformation. The left sides of Eqs. 3.25 are the multiplications of the projective depth and affine parameters in the normalized coordinate system. After elementary modification, the normalized affine parameters become:

$$\hat{a}_1 = \frac{l_x^2}{l_x^1}a_1, \quad \hat{a}_2 = \frac{l_x^2}{l_y^1}a_2, \quad \hat{a}_3 = \frac{l_y^2}{l_x^1}a_3, \quad \hat{a}_4 = \frac{l_y^2}{l_y^1}a_4.$$

The resulting homography $\mathbf{H}$ is calculated from the normalized one by denormalizing: $\mathbf{H} = \mathbf{T}_2^{-1}\hat{\mathbf{H}}\mathbf{T}_1$.

Robustification. One of the main advantages of the proposed HAF method is that the *minimum point number for the estimation is one*. Therefore, the stochastic model creation stage of the applied robust methods such as RANSAC [1] can be removed. Considering the RANSAC strategy which maximizes the inlier number, the globally optimal homography can be found by a simple linear search among the homographies as follows:

$$\mathbf{H}_{\text{opt}} = \arg\max_{\mathbf{H}\in\mathcal{H}} \sum_{i=1}^{n} I(\mathbf{H}, \epsilon), \tag{3.25}$$

where $\mathcal{H}$ and I are the set of the homographies and the function counting the inliers w.r.t. threshold ϵ, respectively.

Remark that this analogy can straightforwardly be applied to other robust methods such as MLESAC [81], LMeDS [82], J-Linkage [83], or T-Linkage [84].

Numerical Optimization. Numerical optimization is necessary since Eqs. 3.24 are given as the multiplication of the original equations by their denominators. This operation distorts the original signal-noise ratio, thereby numerical refinement of the obtained results is required using the original formulas. For that purpose, Levenberg-Marquardt [85] optimization technique is used to minimize the following affine error:

$$\begin{aligned}
\mathcal{E}(\mathbf{H}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{A}) &= \mathcal{E}_{\text{A}}(\mathbf{H}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{A}) + \mathcal{E}_{\text{L}}(\mathbf{H}, \mathbf{p}_1, \mathbf{p}_2) \\
\mathcal{E}_{\text{A}}(\mathbf{H}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{A}) &= \tfrac{1}{s}\left|\left|\begin{bmatrix} h_{11} - h_{31}u^2 & h_{12} - h_{32}u^2 \\ h_{21} - h_{31}v^2 & h_{22} - h_{32}v^2 \end{bmatrix} - \mathbf{A}\right|\right|_F, \\
\mathcal{E}_{\text{L}}(\mathbf{H}, \mathbf{p}_1, \mathbf{p}_2) &= \left|\left|\tfrac{\mathbf{H}\mathbf{p}_1}{s} - \mathbf{p}_2\right|\right|_2^2,
\end{aligned} \tag{3.26}$$

where $\mathcal{E}$ (overall error) is the sum of errors $\mathcal{E}_\mathrm{A}$ and $\mathcal{E}_\mathrm{L}$ and $s = \mathbf{h}_3^\mathrm{T}\mathbf{p}_1$ is the projective depth. Error E_A is the one yielded by the affine transformations. It is the Frobenius-norm of the difference matrix of the measured affine transformation and the one which homography $\mathbf{H}$ yields at points $\mathbf{p}_1$ and $\mathbf{p}_2$ (Eqs. 3.18). Function E_L is the L_2 norm of the re-projection error. The application of the Frobenious-norm is justified in Appendix D

Algorithmic Details. The normalized HAF algorithm is written in Alg. 1. Its input are two sets $\mathcal{P}_1$ and $\mathcal{P}_2$ of point correspondences, a set $\mathcal{A}$ of local affine transformations for each point pair and the fundamental matrix $\mathbf{F}$ ($|\mathcal{P}_1| = |\mathcal{P}_2| = |\mathcal{A}| \geq 1$). Note that $\mathbf{F}$ can be estimated by e.g. RANSAC combined with the 7-point algorithm [43] as an engine. The output is the estimated homography $\mathbf{H}$. Due to nature of data normalization, it is undefined for less than two correspondences – this condition is written in the first line of Alg. 1. For robust estimation, this algorithm is inserted into a robust method, e.g. RANSAC.

Algorithm 1 Normalized HAF

Input: $\mathcal{P}_1$, $\mathcal{P}_2$ – points in the first and second images, $\mathcal{A}$ – local affine transformations, $\mathbf{F}$ – fundamental matrix
Output: $\mathbf{H}$ – homography

1: **if** $|\mathcal{P}_1| > 1$ **then** ▷ Normalization is undefined for one correspondence
2: $\quad\mathcal{P}_1, \mathcal{P}_2, \mathcal{A}, \mathbf{F}$:= NormalizeData($\mathcal{P}_1, \mathcal{P}_2, \mathcal{A}, \mathbf{F}$)
3: $\mathbf{C}, d$:= BuildCoefficientMatrix($\mathcal{P}_1, \mathcal{P}_2, \mathcal{A}, \mathbf{F}$); ▷ Eq. 3.24
4: $x := \mathbf{C}^\dagger d$ ▷ † is the Moore-Penrose pseudo-inverse
5: $\mathbf{H}$:= HomographyFromFundamentalMat(x, $\mathbf{F}$) ▷ Eq. 3.21

3.2.3 Experimental Results

It is presented in this section that the proposed method is applicable to both synthetic and real world data. A potential way is proposed to provide input data using real images, and it is shown that HAF makes the multi-homography estimation less ambiguous.

Synthesized Tests. Generating a synthetic scene, two perspective cameras were generated by their projection matrices $\mathbf{p}_1$ and $\mathbf{p}_2$. The fundamental matrix was computed using the two projection matrices [43]. Their positions were restricted to plane $z = 60$ (see Fig. 3.7(a)). The common focal length and the principal point were set to $f_x = f_y = 600$ and $p_0 = [300 \quad 300]^\mathrm{T}$. Then 50 points were sampled from a random plane passes over the origin and projected onto the cameras. Zero-mean Gaussian noise was added to the point coordinates. The affine transformations were calculated from the noisy point locations and the ground truth homography using Eqs. 3.18. Methods were applied to these correspondences. Tests were repeated 5000 times on each noise level.

The three competitor methods were the normalized DLT[3], normalized 3PT, and normalized HA methods[4]. Note that 3PT (Eqs. 3.22, 3.23) is constructed similarly to

[3] It is implemented in OpenCV 3.0.
[4] http://web.eee.sztaki.hu/~dbarath/

the original three-point algorithm [43], and HA (Eqs. 3.19) is similar to the proposed method by [5]. The point-based algorithms (DLT and 3PT) were followed by a numerical refinement stage minimizing the re-projection error (based on $\mathbf{H}\mathbf{p}_1 \sim \mathbf{p}_2$) by Levenberg-Marquardt [85] optimization technique. HA and the proposed HAF methods also use Levenberg-Marquardt optimization to minimize the affine error given by Eq. 3.26.

Fig. 3.8 shows the mean and median errors (vertical axis) of each method w.r.t. increasing noise level (horizontal axis). HAF method outperforms the competitor ones in both aspects. The mean diagram shows that normalized DLT becomes very unstable in several cases, while the other point-based method (3PT) is stable due to the fundamental matrix.

The top-left (mean error) and top-right (median error) charts of Fig. 3.9 show the effect of the proposed normalization for increasing noise. The normalized version of HAF is significantly more stable then the original one. The bottom-left chart shows the average processing time (in milliseconds) of each method plotted as the function of the point number. Even though HAF is the slowest one, its time demand is approx. 0.004 sec for one thousand points. Therefore, it is still applicable to real time tasks. The bottom-right figure visualizes the effect of different view-angles. For that test only one correspondence was considered, therefore, only HAF method was applicable. The plane on which the observed point lies was the XZ plane. The cameras were placed on the surface of a 60-unit sphere around the origin and looked at the observed point (see Fig. 3.7(b)). The horizontal axis of the chart represents the view-angle. If it is $0°$, both cameras are in the same position over the observed point (at the top of the sphere). As the view-angle getting higher, the cameras are getting lower on the sphere. It can be seen that the method is sensitive to high view-angle, where the observed plane is nearly perpendicular to the view-plane. Even so, this sensitivity is significant only over $60 - 70°$ which is a challenging case for feature detectors as well.

Affine-covariant Feature Detectors. There are many affine-covariant feature detectors [57] available in the field such as MODS [46][5], ASIFT [2], ASURF, AAKAZE, ABRISK, AORB, AHessian-Affine[6], Harris-Affine, Hessian-Affine [3], MSER [86][7], etc. They obtain an affine transformation $\mathbf{A}_i^k$ for the kth point in the ith image ($i \in \{1, 2\}$). Then the affinity which transforms $\mathbf{A}_1^k$ in the first image to $\mathbf{A}_2^k$ in the second one as $\mathbf{A}^k\mathbf{A}_1^k = \mathbf{A}_2^k$ is as follows: $\mathbf{A}_i^k = \mathbf{A}_2^k(\mathbf{A}_1^k)^{-1}$.

In order to test the quality of the detectors w.r.t. the estimated homography, the publicly available AdelaideRMF dataset[8] was used. It consists of point correspondences and a label for each which denotes the plane to which the point is assigned. All affine-covariant detectors were applied to every image pair. The related homographies were estimated for all of the obtained correspondences using the proposed HAF method. The closest annotated homography is assigned to each point pair. If the projection error from the closest homography is higher than 1.0, the correspondence is discarded from the evaluation.

[5] http://cmp.felk.cvut.cz/wbs/

[6] http://www.ipol.im/pub/art/2011/my-asift/.
Each detector – AKAZE, BRISK, ORB, SIFT, SURF, Hessian-Affine – replaces SIFT in the view-synthesizer.

[7] MSER, Harris-Affine, and Hessian-Affine are downloaded from http://www.robots.ox.ac.uk/~vgg/research

[8] http://cs.adelaide.edu.au/~hwong/doku.php?id=data

TABLE 3.5: Mean re-projection errors of $\mathbf{H}_{\text{opt}}$ homographies per annotated plane on test pairs (a – i) from AdelaideRMF dataset. Columns N, $Cvg.$ and T are the average number of points, the percentage of the coverage, and the processing time (in seconds) of each method, respectively. Coverage is the number of planes for which the detector obtains at least one point correspondence divided by the ground truth plane number. Test pairs: (a) hartley, (b) johnsonnb, (c) neem, (d) sene, (e) oldclassicswing (f) ladysymon, (g) napierb, (h) bonhall, (i) unihouse, (j) elderhalla.

	N	Cvg.	T	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	avg	med
AAKAZE	258	94%	82	3.2	28.6	4.2	10.3	4.6	3.4	3.6	4.3	3.1	13.3	7.7	4.3
ABRISK	338	96%	81	2.3	10.3	3.7	2.9	3.6	3.7	7.5	2.0	1.5	4.7	4.2	3.7
AHES-AFF	1553	100%	89	1.9	4.5	2.4	2.1	1.4	1.5	3.3	1.7	1.4	3.3	2.3	2.0
AORB	117	83%	86	3.8	16.0	9.4	3.4	3.4	7.1	12.3	3.5	4.2	14.4	7.5	6.3
ASIFT	2468	99%	81	**1.6**	**2.5**	**1.8**	**1.8**	1.1	3.0	**2.8**	**1.1**	**1.3**	3.2	**2.0**	**1.8**
ASURF	1183	100%	84	2.1	4.3	3.6	2.5	1.2	**1.5**	2.9	1.5	2.4	**3.2**	2.5	2.5
HAR-AFF	79	97%	4	2.8	4.2	2.6	2.7	3.1	5.6	3.3	3.5	2.5	5.6	3.5	3.2
HES-AFF	66	81%	3	2.8	5.4	2.2	3.4	1.7	2.6	3.5	6.9	2.8	3.5	3.4	2.9
MODS	846	99%	53	6.0	4.9	2.6	3.2	**0.9**	1.5	4.9	1.9	2.2	3.9	3.2	3.0

Fig. 3.10 shows the mean projection error of each method. The blue bars are the average projection errors of all obtained homographies. The red ones are the average projection errors of homographies $\mathbf{H}_{\text{opt}}$ (Eq.3.25), each is associated to a plane. It is interesting that the best method is different w.r.t. these two aspects. AHessian-Affine and Hessian-Affine yield the most accurate homographies on average. Even so, the lowest error of each $\mathbf{H}_{\text{opt}}$ is achieved by ASIFT [2]. This is caused by the number of detected correspondences since ASIFT yields the most (Table 3.5). Therefore, the probability of measuring a very precise estimate is even higher.

Table 3.5 shows the errors of the obtained $\mathbf{H}_{\text{opt}}$ matrices of each affine-covariant detector. The error value is the mean of the projection errors computed from the point correspondences on the annotated planes with the related $\mathbf{H}_{\text{opt}}$. As it is also presented in Fig. 3.10, the most accurate $\mathbf{H}_{\text{opt}}$ homographies are carried out by ASIFT feature detector.

Multiple Homography Estimation. Even though this main section does not address accurate multi-homography estimation, we show that the usage of these point-wise homographies reduces the ambiguity of this problem.

Most of the multi-model estimation algorithms are based on stochastic model creation [13], [83], [84]. They differ in the way how they use these randomly selected models. Stochastic sampling is essential to create hypotheses since models are often estimated using many data points. By the usage of the proposed method this step can be omitted in the case of multi-homography estimation because a homography is given for every single correspondence. Therefore, *the resulting method is not based on random sampling, it is deterministic.*

In order to demonstrate the advantage of this property, we replaced the model creation stage of T-Linkage [84][9] with the obtained models of the proposed method. T-Linkage can be considered as one of the state-of-the-art multi-model fitting techniques. The top row of Fig. 3.11 shows the obtained planes by including the well-known 4-point algorithm [43] into T-Linkage. The bottom row visualizes the resulting segmentation by HAF method. Each column consists of the first image of an image pair from AdelaideRMF dataset. Even though that neither segmentations are perfect, the one uses HAF method to model estimation obtains significantly more

[9] http://www.diegm.uniud.it/fusiello/demo/jlk/

accurate results. The same parameter setup is used for both cases. Points which are assigned to no plane are not visualized.

3.2.4 Summary

A method is presented in this section to estimate a planar homography from a single affine correspondence. The method is extended to the overdetermined case. It is shown that it outperforms the competitor methods on synthesized data. In order to apply it to real world image pairs, the available state-of-the-art affine-covariant feature detectors are compared to each other w.r.t. the accuracy of estimable homographies. On average, AHessian-Affine and Hessian-Affine obtain the most accurate point-wise homographies. However, ASIFT is the most robust method due to the large number of detected points. Finally, it is presented that the usage of these point-wise homographies makes multi-homography estimation process less ambiguous.

3.3 Homographies using Partial Affine Correspondences

Estimating planar correspondences is a crucial part of several vision tasks e.g. robot vision [74], [87], camera calibration [70]–[72], 3D reconstruction [68], [69] and augmented reality applications [73]. Even though the most popular estimation techniques are based on point correspondences [88], a homography is estimable from line [88], region [89], contour [78], or affine correspondences [5], [30], [90]. Most of these algorithms include data normalization [88], and numerical optimization to minimize the effect of the noise. In this section, we assume that not only the point locations but several affine components and the fundamental matrix are known.[10]

Local affine transformations have become more popular in the last decade. Matas et al. [91] presented that local affinities can support stereo matching. The 3D camera pose can also be estimated using a corresponding point pair and the related affinity as it is proposed by Köser and Koch [92]. These transformations can facilitate the recovery of spatial point coordinates [5]. Current 3D reconstruction pipelines exploit point correspondences as well as patches [9], [66], [93] to compute realistic 3D models of real-world objects. Bentolila et al. [94] proved that affine transformations put constraints on the epipoles in stereo images. Barath et al. [32] showed that a one-to-one relationship exists between the surface normal and the local affinity.

Even though local affine transformations are useful and can significantly improve the quality of the estimation, it is time consuming to recover them – e.g. by affine covariant detectors which cannot be applied in real time. Even so, most of the detectors obtain some part of these local affinities, such as SIFT [95] or SURF [96] recovering the rotational and scale components. Therefore, using solely the translation part (the point location) causes information loss. The motivation of this research is to formulate a general theory about the usage of the affine components obtained by partially affine covariant feature detectors. The main contributions are as follows: (i) A general theory to exploit the affine components obtained by partially affine covariant detectors which is real time capable. (ii) The proposed method estimates the homography from two SIFT correspondences if the fundamental matrix is known. To our knowledge, the minimum number of required correspondences was three before this work.

[10]The pre-estimation of the fundamental matrix for rigid scenes using point correspondences is a usual step in computer vision pipelines. The proposed theory is straightforward to generalize for multiple rigid motions.

3.3.1 Homographies and Partial Affine Transformations

In this section, we show that the homography estimation problem becomes much simpler if the epipolar geometry and local affine transformations are known.

Homography from Affinities. As it is shown in the Homography from Affine transformation and Fundamental matrix (HAF) method [30] the estimation of a homography can be written in an inhomogeneous, linear form if at least one local affine transformation and the epipolar geometry is known. The coefficient matrix $\mathbf{C}$ is as follows:

$$\mathbf{C} = \begin{bmatrix} a_{11}^i x_1^i + x_2^i - e_x & a_{11}^i y_1^i & a_{11}^i \\ a_{12}^i y_1^i + x_2^i - e_x & a_{12}^i x_1^i & a_{12}^i \\ a_{21}^i x_1^i + y_2^i - e_y & a_{21}^i y_1^i & a_{21}^i \\ a_{22}^i y_1^i + y_2^i - e_y & a_{22}^i x_1^i & a_{22}^i \end{bmatrix}. \tag{3.27}$$

The equation system can be formed as $\mathbf{Ck} = \mathbf{d}$, where vector $\mathbf{d} = [f_{21} \quad f_{22} \quad -f_{11} \quad -f_{12}]^{\mathrm{T}}$ is the inhomogeneous part while $\mathbf{k} = [h_{31} \quad h_{32} \quad h_{33}]^{\mathrm{T}}$ is the vector of the unknown parameters. The optimal solution in the least squares sense is given by $\mathbf{k} = \mathbf{C}^{\dagger}\mathbf{d}$, where $\mathbf{C}^{\dagger}$ is the Moore-Penrose pseudo-inverse of matrix $\mathbf{C}$. Note that augmenting this system with the formulas regarding to the point locations (Eqs. 3.22, 3.23) leads to more robust estimation.

Affine Transformation Model. Let us denote the affine transformation related to the ith ($i \in [1, N]$) point pair without the translation part as follows:

$$\mathbf{A}^i = \begin{bmatrix} a_{11}^i & a_{12}^i \\ a_{21}^i & a_{22}^i \end{bmatrix} = \begin{bmatrix} \cos(\alpha^i) & -\sin(\alpha^i) \\ \sin(\alpha^i) & \cos(\alpha^i) \end{bmatrix} \begin{bmatrix} s_x^i & w^i \\ 0 & s_y^i \end{bmatrix} = \begin{bmatrix} s_x^i \cos(\alpha^i) & w^i \cos(\alpha^i) - s_y^i \sin(\alpha^i) \\ s_x^i \sin(\alpha^i) & w^i \sin(\alpha^i) + s_y^i \cos(\alpha^i) \end{bmatrix} \tag{3.28}$$

Variables α^i, s_x^i, s_y^i, and w^i are the rotational angle, scales along x and y axes, and the shear parameters, respectively.

Homography from Partially Known Affine Transformation. It is shown here that not the full local affinity is necessary for homography estimation, but their parts – obtained by e.g. SIFT or other partially affine covariant detector – can also be exploited. *In the rest of this section, the proposed method is called P-HAF as the abbreviation of Partial HAF.* Let us substitute Eqs. 3.28 into Eqs. 3.27 as

$$h_{31}\left(s_x \cos(\alpha^i) x_1^i + x_2^i - e_x\right) + h_{32} s_x^i \cos(\alpha^i) y_1^i + h_{33} s_x^i \cos(\alpha^i) = f_{21}, \tag{3.29}$$

$$h_{32}\left((w^i \cos(\alpha^i) - s_y^i \sin(\alpha^i)) y_1^i + x_2^i - e_x\right) + h_{31}(w^i \cos(\alpha^i) - s_y^i \sin(\alpha^i)) x_1^i + h_{33}(w^i \cos(\alpha^i) - s_y^i \sin(\alpha^i)) = f_{22}, \tag{3.30}$$

$$h_{31}\left(s_x^i \sin(\alpha^i) x_1^i + y_2^i - e_y\right) + h_{32} s_x^i \sin(\alpha^i) y_1^i + h_{33} s_x^i \sin(\alpha^i) = -f_{11}, \tag{3.31}$$

$$h_{32}\left((w^i \sin(\alpha^i) + s_y^i \cos(\alpha^i)) y_1^i + v_i^2 - e_y\right) + h_{31}(w^i \sin(\alpha^i) + s_y^i \cos(\alpha^i)) x_1^i + h_{33}(w^i \sin(\alpha^i) + s_y^i \cos(\alpha^i)) = -f_{12}. \tag{3.32}$$

These four equations contain the affine transformation in an easy-to-handle form. For a given part of the affinity, e.g. rotation and scale, the appropriate equations can

be selected and used. After the selection, the given system is linear, inhomogeneous and can straightforwardly be solved.

Specialization to SIFT Features. The popular SIFT [95] detector obtains rotation and scale covariant features, therefore, the proposed theory can be specialized to use SIFT. Beside the point locations, the rotation and the scale is given for each feature point. After the matching process, the related parts of the local affine transformation are as follows:

$$s = \frac{s_2}{s_1}, \quad \alpha = \alpha_2 - \alpha_1,$$

where s_1, s_2, α_1, and α_2 are the scales and angles in the two images, respectively. Here, we assume s as horizontal scale, thus only Eqs. 3.29 and 3.31 have to be kept. Even though one SIFT correspondence yields three equations – one from the locations and two from the affinity –, the two regarding to the affine parts are linearly dependent. *As a consequence, two SIFT correspondences are enough for homography estimation* – and the system has been already overdetermined.

For $n \geq 2$ point pairs, an overdetermined, inhomogeneous, linear system is formed.

Normalization. As it is well-known, normalization of the input data is a usual and important part of homography estimation [88] due to the numerical instability. Let us denote the normalization transformations by $\mathbf{T}_1$ and $\mathbf{T}_2$ where the normalized homography is calculated as $\hat{\mathbf{H}} = \mathbf{T}_2\mathbf{H}\mathbf{T}_1^{-1}$. The transformation matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ are special affine transformations: they consist of translation and scale. The horizontal and vertical scales of the two transformations are denoted by l_x^k and l_y^k ($k \in \{1, 2\}$), respectively. The normalization of the fundamental matrix and the point coordinates is written in the previous section. We discuss here, how the affine components have to be normalized.

As it is shown in the previous section (see Eq. 3.25), the affine components are modified as follows:

$$\hat{a}_{11}^i = \left(l_x^2/l_x^1\right) a_{11}^i, \quad \hat{a}_{12}^i = \left(l_x^2/l_y^1\right) a_{12}^i, \quad \hat{a}_{21}^i = \left(l_y^2/l_x^1\right) a_{21}^i, \quad \hat{a}_{22}^i = \left(l_y^2/l_y^1\right) a_{22}^i.$$

The normalized affine transformation modify Eqs. 3.29–3.32 as

$$\begin{aligned} h_{31}\left(s_x^i\left(l_x^2/l_x^1\right)\cos(\alpha_i)x_1^i + x_2^i - e_x\right) + \\ \left(l_x^2/l_x^1\right)\left(h_{32}s_x^i\cos(\alpha_i)y_1^i + h_{33}s_x^i\cos(\alpha_i)\right) = f_{21}, \end{aligned} \tag{3.33}$$

$$\begin{aligned} h_{32}\left((w_i\cos(\alpha_i) - s_y^i\sin(\alpha_i))\left(l_x^2/l_y\right)y_1^i + x_2^i - e_x\right) + \\ \left(l_x^2/l_y^1\right)h_{31}(w_i\cos(\alpha_i) - s_y^i\sin(\alpha_i))x_1^i + \\ \left(l_x^2/l_y^1\right)h_{33}(w_i\cos(\alpha_i) - s_y^i\sin(\alpha_i)) = f_{22}, \end{aligned} \tag{3.34}$$

$$\begin{aligned} \left(l_y^2/l_x^1\right)h_{31}\left(s_x^i\sin(\alpha_i)x_1^i + y_2^i - e_y\right) + \\ \left(l_y^2/l_x^1\right)\left(h_{32}s_x^i\sin(\alpha_i)y_1^i + h_{33}s_x^i\sin(\alpha_i)\right) = -f_{11}, \end{aligned} \tag{3.35}$$

$$\begin{aligned} \left(l_y^2/l_y^1\right)h_{32}\left((w_i\sin(\alpha_i) + s_y^i\cos(\alpha_i))y_1^i + y_2^i - e_y\right) + \\ \left(l_y^2/l_y^1\right)h_{31}(w_i\sin(\alpha_i) + s_y^i\cos(\alpha_i))x_1^i + \\ \left(l_y^2/l_y^1\right)h_{33}(w_i\sin(\alpha_i) + s_y^i\cos(\alpha_i)) = -f_{12}. \end{aligned} \tag{3.36}$$

TABLE 3.6: The processing time (in milliseconds) of normalized P-HAF – including normalization – implemented in Matlab and C++. The first row shows the time of P-HAF applied to a minimal subset – two correspondences. The second one reports the mean time on all pairs of the AdelaideRMF and Multi-H datasets. On average, P-HAF is applied to 27 SIFT point pairs as an overdetermined system.

	Matlab (ms)	C++ (ms)
2 points	0.336	0.005
N points	1.106	0.012

If the system is combined with equations of the 3-point algorithm (Eqs. 3.22, 3.23), an inhomogeneous, linear system of equations is obtained. Note that the normalized correspondences and $\hat{\mathbf{F}}$ are used in Eqs. 3.33– 3.36.

Algorithmic Details. Alg. 2 shows the P-HAF algorithm specialized to SIFT features. The required input is a set of point correspondences P and the related components rotation R and scale S, for each. The output is the homography.

Algorithm 2 P-HAF for SIFT points

Input: $\mathcal{P}$ – points on the first and second images
$\mathcal{R}$, $\mathcal{S}$ – rotation and scale for each point pair
$\mathbf{F}$ – fundamental matrix
Output: $\mathbf{H}$ – homography

1: $\mathcal{P}, \mathcal{R}, \mathcal{S}$:= Normalization(. . .);
2: $\mathbf{C}, \mathbf{d}$:= BuildCoefficientMatrix(. . .); ▷ Eqs. 3.29, 3.31
3: $\mathbf{k} := \mathbf{C}^{\dagger}\mathbf{d}$; ▷ † is the Moore-Penrose pseudo-inverse
4: $\mathbf{H}$:= HomographyFromFundMat(x, $\mathbf{F}$);

Processing Time. The processing time of the proposed algorithm depends on the solution of the inhomogeneous, linear system which can be carried out via its Moore-Penrose pseudo-inverse. On a serial processor its time complexity is $\mathcal{O}(m^3) + \mathcal{O}(r^3)$ where m and r are the row number of the coefficient matrix $\mathbf{A}$ and its rank, respectively. Remark that it is reduced to $\mathcal{O}(m) + \mathcal{O}(r^3)$ in parallel computing [97]. Therefore, P-HAF is computable in a few milliseconds (see Table 3.6).

Time demand of RANSAC. Augmenting RANSAC [1] or other robust methods with P-HAF significantly reduces the iteration number, thus higher processing speed is achieved. Table 2.1 reports the required iteration number [88] of RANSAC to converge using different minimal methods as engine.
It can be seen that using two points leads to significantly less iterations, thus speeding up the process, especially for high outlier ratio.

3.3.2 Experimental Results

The aim of this section is to show that the proposed theory works both on synthetic and real world data. All algorithms ended with a numerical refinement stage using Levenberg-Marquardt optimization technique [85] to minimize the re-projection error. The competitor methods are the Direct Linear Transformation (DLT) and Three Point Method (3PT) applied to normalized data.

TABLE 3.7: The mean re-projection error (in pixels) of the methods applied to the AdelaideRMF and Multi-H datasets. Each row represents an image pair and each column consists of the re-projection errors of a method. Homographies are estimated using the 25% of the correspondences, re-projection error is computed w.r.t. all of them. Test pairs: (1) `barrsmith`, (2) `bonhall`, (3) `bonython`, (4) `boxesandbooks`, (5) `elderhallb`, (6) `glasscasea`, (7) `glasscaseb`, (8) `graffiti`, (9) `johnssona`, (10) `johnssonb`, (11) `library`, (12) `napiera`, (13) `napierb`, (14) `neem`, (15) `nese`, (16) `sene`, (17) `unihouse`, (18) `unionhouse`.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	avg	med
P-HAF	**27.0**	1.0	**1.3**	**2.1**	**4.7**	**7.9**	9.6	**0.9**	10.8	**5.3**	**4.8**	15.0	17.7	**4.3**	**4.4**	**4.1**	8.8	**7.0**	**7.6**	**5.1**
DLT	36.0	**0.8**	1.4	8.5	5.3	26.8	21.3	1.0	**10.4**	6.3	6.0	**14.5**	30.6	5.4	6.9	7.9	**5.4**	7.5	11.2	7.2
3PT	27.2	1.0	1.4	**2.1**	5.2	9.6	**8.0**	1.0	11.3	5.7	5.0	17.8	**17.3**	5.6	4.7	4.7	5.6	**7.0**	7.8	5.6

Synthesized Tests. For synthesized testing, two perspective cameras are generated by their projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$. Their positions are randomized – using uniform distribution – on a plane represented by function $S_c(u, v) = [u \quad v \quad 60]^{\mathrm{T}}$, $(u, v \in [-20, 20])$. Both cameras point towards the origin. Their common focal length and principal point are 600 and $[300 \quad 300]^{\mathrm{T}}$, respectively. Fundamental matrix $\mathbf{F}$ is computed from projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ [88].

A plane passing through the origin is generated with random orientation and sampled in 50 different locations – these points are projected onto cameras $\mathbf{P}_1$ and $\mathbf{P}_2$. Zero-mean Gaussian-noise is added to the point coordinates. The local affinity related to each point pair is calculated from the plane parameters [30] and the noisy point locations, then decomposed into the form

$$\mathbf{A} = \begin{bmatrix} s_x \cos(\alpha_i) & w \cos(\alpha) - s_{i,y} \sin(\alpha) \\ s_x \sin(\alpha_i) & w \sin(\alpha) + s_{i,y} \cos(\alpha) \end{bmatrix},$$

and angle α, scale s_x are kept. Tests are repeated 500 times on every noise level.

Fig. 3.12(a) and Fig. 3.12(b) visualize the mean and median errors of the normalized DLT, 3PT and P-HAF methods plotted as the function of the σ value of the zero-mean Gaussian-noise. P-HAF achieves the lowest mean and median errors. Fig. 3.12(c) shows the effect of the normalization. Even though the difference is not significant, the normalized algorithm is the most accurate estimator.

Homography Estimation. In order to test P-HAF on real world images AdelaideRMF [98] and Multi-H [27] datasets are used. They consist of images of different sizes and point correspondences assigned to planes. Figure 3.13 shows four example images, the first one from each stereo pair, from the datasets. The left column is from Multi-H, pairs `boxesandbooks` and `glasscasea`, and the right one from AdelaideRMF – pairs `elderhalla` and `bonhall`. Points are painted by circles and each is assigned to a plane by color.

Annotations contain no information about the rotational or scale components, therefore, SIFT detector is applied to each image pair. Then the closest detected feature is paired to every annotated one. If the distance is greater than 5 pixels the point pair is omitted from the evaluation. The fundamental matrix $\mathbf{F}$ is estimated by the RANSAC eight-point technique [88] with threshold value set to 1.0 followed by a Levenberg-Marquardt optimization minimizing symmetric epipolar distance. Every homography is estimated using the 25% of the correspondences, however, the reported re-projection errors are computed using all of them.

In Table 3.7, the mean re-projection errors (in pixels) are reported. Columns represent different test pairs from the AdelaideRMF and Multi-H datasets, rows show

the related errors. It can be seen that the mean errors of P-HAF and 3PT are quite similar, even so, P-HAF is slightly better. The median error of P-HAF is significantly better than that of DLT and 3PT. This is expected since DLT and 3PT use a smaller part of the underlying affine transformation – the translation – while P-HAF exploits all the available information.

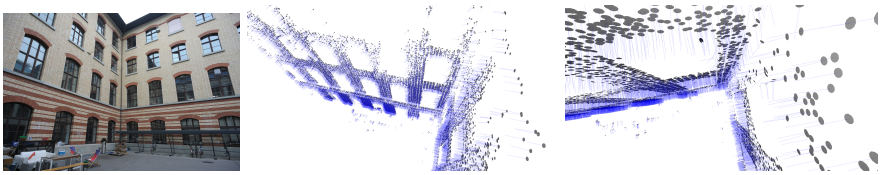
Multiple Homography Fitting. One of the main advantage of P-HAF is the required minimal point number as it is able to estimate a homography from only two SIFT correspondences. DLT needs four and 3PT three of those. Most of the robust model fitting techniques, e.g. RANSAC, are based on minimum subsets consisting of the minimum number of data to estimate a given model. Using as few data as possible makes the estimation faster, less ambiguous, and possibly more accurate.

In this section, a multi-model fitting technique, PEARL [13], is augmented with different model initialization methods: normalized DLT and P-HAF. We used the same datasets as in the previous experiments, AdelaideRMF and Multi-H. AdelaideRMF mainly consists of buildings while Multi-H smaller planar objects.

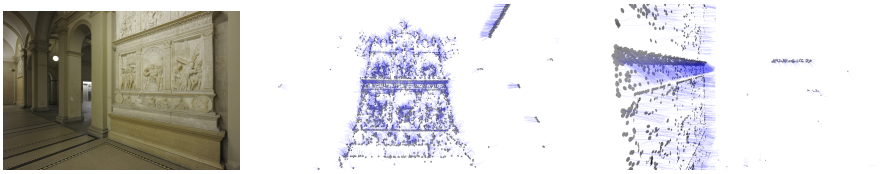
Fig. 3.14 shows the results of multi-homography fitting. Each row consists of the first image of a selected test pair. The left column shows the original image and the other ones report the obtained planar labellings obtained by PEARL with different hypothesis generation techniques: normalized DLT (middle) or P-HAF (right). The same parameters are used for all the tests and the same amount of hypothesizes are generated. The reported misclassification error (ME) is the ratio of the points assigned to wrong plane in percentage. It can be seen that *PEARL augmented with P-HAF is significantly more accurate then the one using normalized DLT*.

3.3.3 Summary

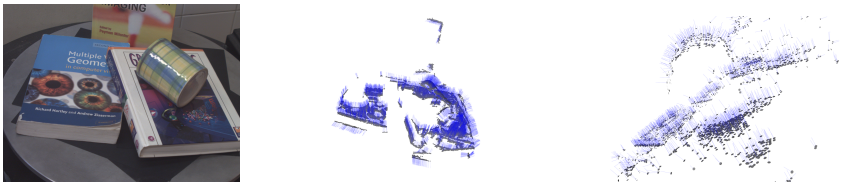
A novel minimal method is presented in this section to improve the general point-based homography estimation by exploiting the information yielded by the commonly used feature detectors. The proposed P-HAF method is able to estimate the homography using at least two SIFT correspondences and applicable in real time. The main message of this section is that usually there are more information about the underlying homography than only the point coordinates – e.g. SIFT, SURF obtain the rotational component and the scale as well. Neglecting this information yields information loss. *We see no reasons to use the four-point algorithm instead of P-HAF* for rigid scenes if SIFT or SURF features are given.



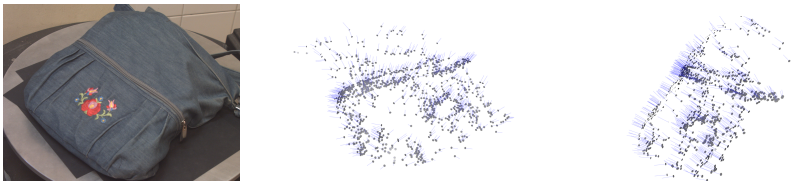
(a) `courtyard` (ETH3D)



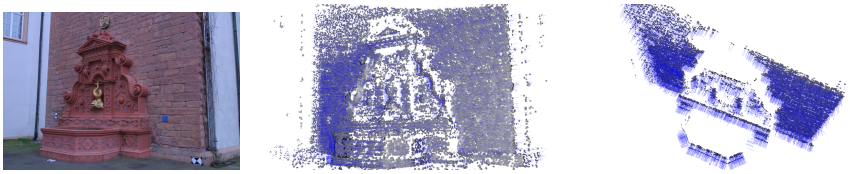
(b) `relief` (ETH3D)



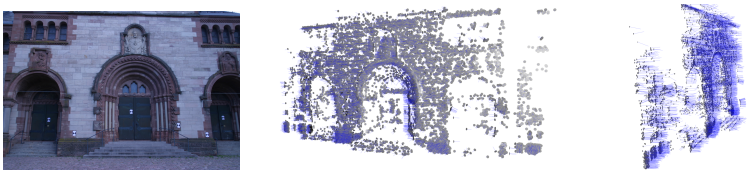
(c) `books2` (Pusztai)



(d) `bag` (Pusztai)



(e) `fountain-p11` (Strecha)



(f) `herz-jesus-p25` (Strecha)

FIGURE 3.5: Example results from each dataset. The first column is an image from the sequence, the remaining ones show the estimated normals (blue lines) and the triangulated points (gray patches) from different view-points.

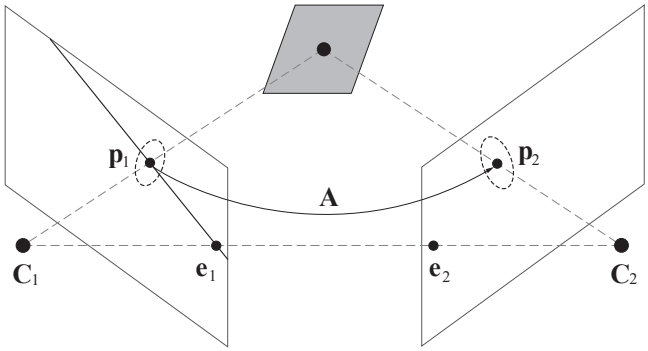
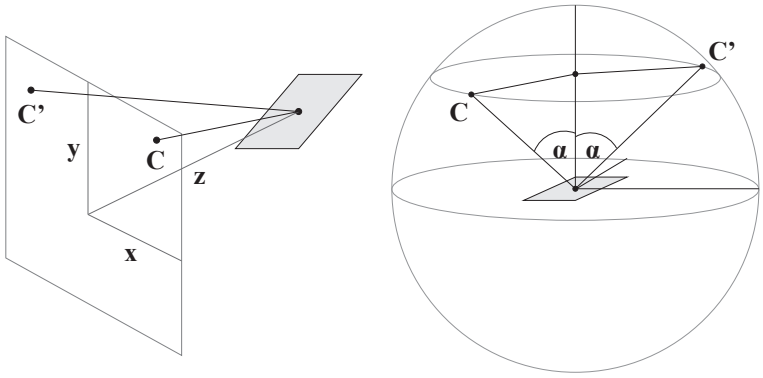


FIGURE 3.6: Two projections of a 3D point lying on the gray plane. Vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ denote the projections in cameras $\mathbf{K}_1$ and $\mathbf{K}_2$. Affine transformation $\mathbf{A}$ maps the infinitely small vicinity of point $\mathbf{p}_1$ to that of $\mathbf{p}_2$. The goal is to estimate the homography corresponding to the plane if the locations $\mathbf{p}_1$, $\mathbf{p}_1$ and affine transformation $\mathbf{A}$ are given.



(a) (b)

FIGURE 3.7: (a) The setup of the synthesized tests. The cameras $\mathbf{K}_1$ and $\mathbf{K}_2$ lie on plane $z = 60$. They observe a random plane passes over the origin. (b) The setup to test the sensitivity w.r.t. view-angle α. The cameras lie on the surface of a sphere around the observed patch.

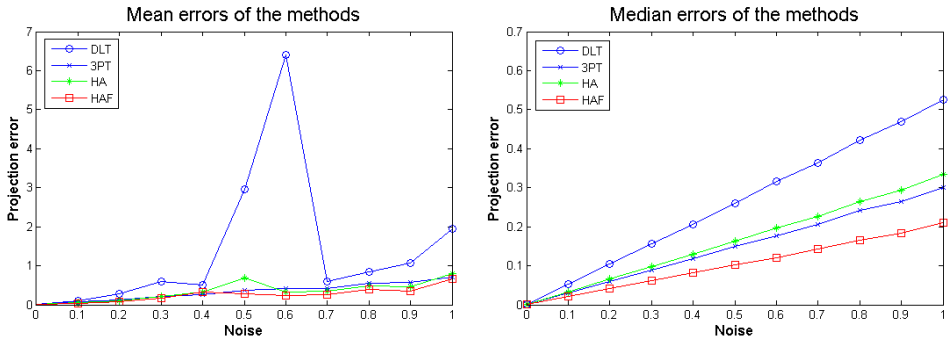


FIGURE 3.8: Mean (left) and median (right) errors plotted as the function of the zero-mean Gaussian noise in pixels (horizontal axis). Vertical axis shows the average of the mean re-projection errors of 5000 runs using 50 correspondences. All methods use normalized data and followed by a numerical refinement stage using Levenberg-Marquardt optimization.

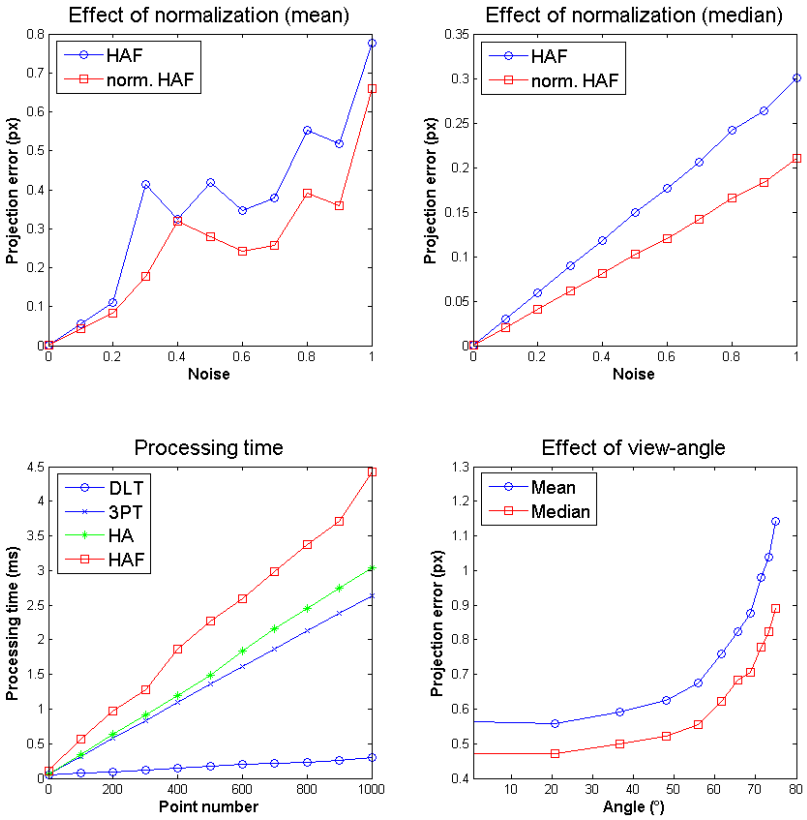


FIGURE 3.9: **(Top-left)** The effect of the normalization is shown w.r.t increasing noise level (horizontal axis). The vertical axis denotes the mean re-projection error in pixels. **(Top-right)** The median errors of the normalized and original HAF. **(Bottom-left)** The processing time in milliseconds of each method plotted as the function of the point number. **(Bottom-right)** The mean and median errors of HAF method w.r.t. increasing view-angle. σ is fixed to 0.5 px. Minimal case is considered.

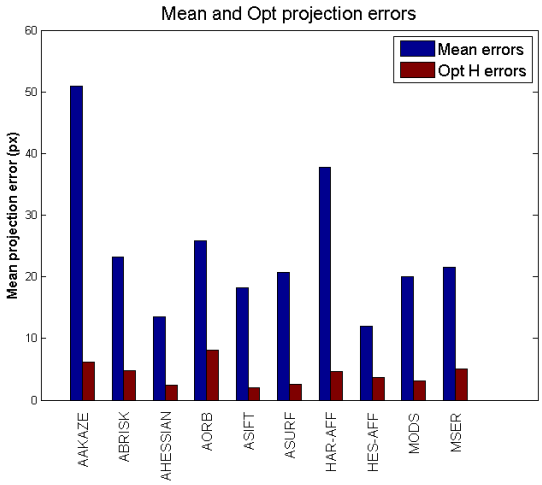
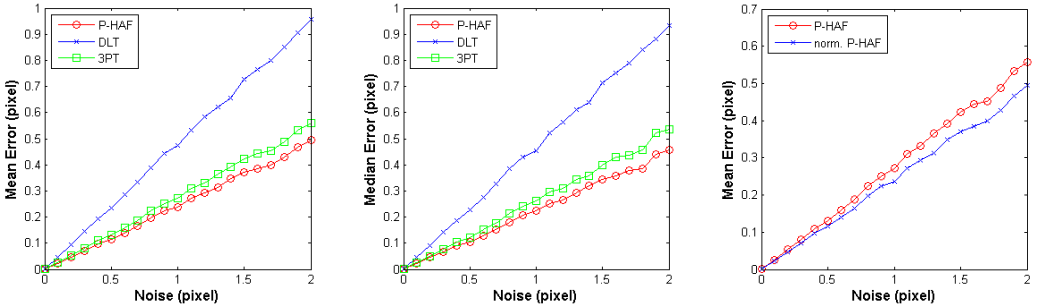


FIGURE 3.10: The mean projection error of all obtained homographies (blue) and that of the $\mathbf{H}_{\text{opt}}$ ones for each plane (red) are shown for every tested affine-covariant detector.

FIGURE 3.11: The obtained planar partitionings by T-Linkage using the 4-point (top) and the proposed HAF (bottom) methods. Each column represents a different test pair. The same parameter setup is used for both of them. Planes are denoted by different colors, points which are assigned to no plane are not visualized.

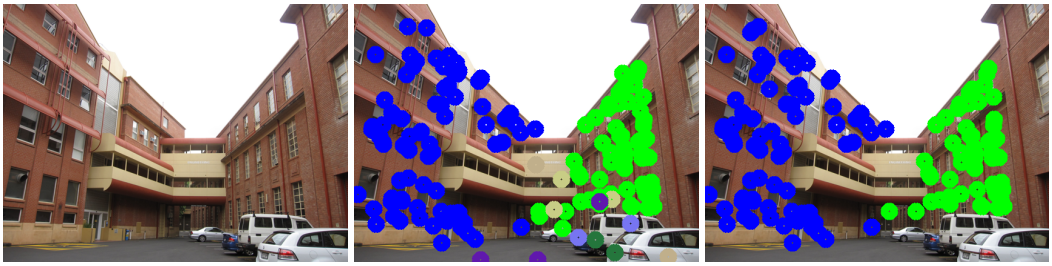


(a) Comparison of methods, mean error. (b) Comparison of methods, median error. (c) The effect of the normalization, mean error.

FIGURE 3.12: Re-projection error (vertical axis) calculated from 500 tests on each noise level. Parameter σ of the zero-mean Gaussian-noise added to the point coordinates is shown on the horizontal axis.

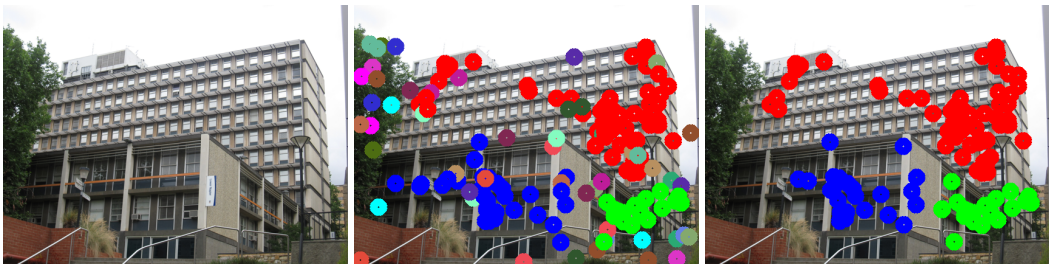
FIGURE 3.13: Example images from the image pairs of Multi-H (left column) and AdelaideRMF (right column) datasets. Points are marked by circles and planes by color.

(a) Test: `neem`. **1.** Original image, **2.** by DLT (ME = 29.46%), **3.** by P-HAF (ME = **10.63**%)



(b) Test: `nese`. **1.** Original image, **2.** by DLT (ME = 19.90%), **3.** by P-HAF (ME = **13.78**%)

(c) Test: `hartley`. **1.** Original image, **2.** by DLT (ME = 19.06%), **3.** by P-HAF (ME = **9.06**%)



(d) Test: `napierb`. **1.** Original image, **2.** by DLT (ME = 38.22%), **3.** by P-HAF (ME = **23.17**%)

FIGURE 3.14: The results of multiple homography fitting to point correspondences. Each row is the first image of a test pair from AdelaideRMF dataset and the results of PEARL. Columns reports the obtained planar labellings of PEARL method with different hypothesis generation techniques: normalized DLT or P-HAF. The same parameters are used for all the tests and the same amount of hypothesizes are generated. The reported misclassification error (ME) is the ratio of the points assigned to wrong plane in percentage. Points are painted by circles and planes marked by color.

Chapter 4

Epipolar Geometry and Affine Correspondences

4.1 Introduction

The objective of this chapter is to establish the direct relationship of epipolar geometry and affine correspondences. Then exploiting the proposed theory, we show how a local affine transformations can be corrected, w.r.t. the epipolar geometry, in closed-form minimizing a cost function which considers the L_2 distance of a measured affinity and the constraints of the epipolar geometry. In the second section, we show that two correspondences are enough to estimate the essential matrix. Then, in the third one, we deal with simultaneous fundamental matrix and focal length estimation assuming the semi-calibrated case, i.e. all intrinsic camera parameters are known except a common focal length.

4.2 Relationship of the Epipolar Geometry and Affine Correspondences

In this section, we show the direct relation of epipolar geometry and local affinities. Directness means in we do not derive the problem exploiting homographies but show the exact effect of affinities on epipolar lines.

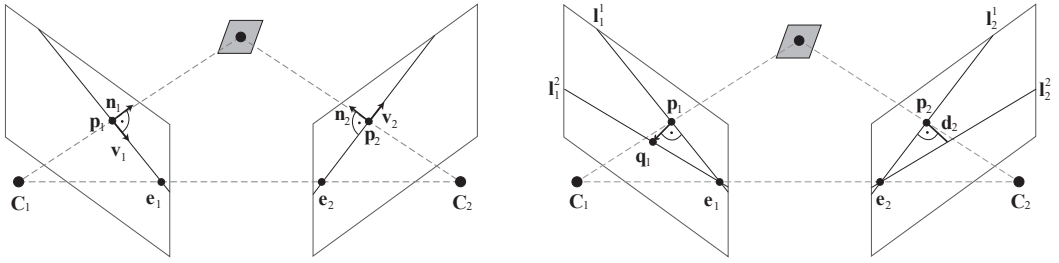
4.2.1 Normal of the Epipolar Line

To consider the direct connection, a possible way is to investigate the effect of local affinity $\mathbf{A}$ on the epipolar lines going through the related point pair $(\mathbf{p}_1, \mathbf{p}_2)$ as follows:

Lemma 1 (Constraints on the Normals of Epipolar Lines)**.** *Given a local affine transformation* $\mathbf{A}$ *transforming the infinitely close vicinities of the related point pair. The normals of the corresponding epipolar lines are* $\mathbf{n}_1$ *and* $\mathbf{n}_2$*. Matrix* $\mathbf{A}$ *is a valid local affine transformation if and only if* $\mathbf{A}^{-T}\mathbf{n}_1 = -\mathbf{n}_2$*.*

Proof. It is trivial that affinity $\mathbf{A}$ transforms the direction of the corresponding epipolar lines to each other as $\mathbf{A}\mathbf{v}_1 \parallel \mathbf{v}_2$, where $\mathbf{v}_1$ and $\mathbf{v}_2$ are the directions of the lines on the two images (see Fig. 4.1(a)). It is well-known from Computer Graphics [99] that this is equivalent to $\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1 = \beta\mathbf{n}_2$, where $\mathbf{n}_1 = (\mathbf{F}^{\mathrm{T}}\mathbf{p}_2)_{1:2}$ and $\mathbf{n}_2 = (\mathbf{F}\mathbf{p}_1)_{1:2}$ are the normals of the epipolar lines ($\beta \neq 0$). Note that lower index $(1:2)$ denotes the first two elements of a vector. We prove here that

$$\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1 = \beta\mathbf{n}_2, \quad \beta = -1. \tag{4.1}$$



(a) The compatibility constraint for orientation states that $\mathbf{A}\mathbf{v}_1 || \mathbf{v}_2$ which is equivalent to $\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1 || \mathbf{n}_2$.

(b) The compatibility constraint for scale states that the ratio of $||\mathbf{p}_1 - \mathbf{q}_1||_2$ and d_2 determines the scale of the related local affine transformation perpendicular to the epipolar line.

FIGURE 4.1: EG-Consistency compatibility constraints for orientation and scale. Matrix $\mathbf{A}$ is the affine transformation, vectors $\mathbf{v}_k$ and $\mathbf{n}_k$ are the direction and normal of epipolar line on which point $\mathbf{p}_k$ lie in the kth image ($k \in \{1, 2\}$).

We are given a corresponding point pair $\mathbf{p}_1 = [x_1 \quad y_1 \quad 1]^{\mathrm{T}}$ and $\mathbf{p}_2 = [x_2 \quad y_2 \quad 1]^{\mathrm{T}}$. Let $\mathbf{n}_1 = [n_{1,x} \quad n_{1,y}]^{\mathrm{T}}$ and $\mathbf{n}_2 = [n_{2,x} \quad n_{2,y}]^{\mathrm{T}}$ be the normal directions of epipolar lines $\mathbf{l}_1 = \mathbf{F}^{\mathrm{T}}\mathbf{p}_2 = [l_{1,a} \quad l_{1,b} \quad l_{1,c}]^{\mathrm{T}}$ and $\mathbf{l}_2 = \mathbf{F}\mathbf{p}_1 = [l_{2,a} \quad l_{2,b} \quad l_{2,c}]^{\mathrm{T}}$. It is well-known that $\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1 = \beta\mathbf{n}_2$ due to $\mathbf{A}\mathbf{v}_1 \parallel \mathbf{v}_2$, where $\beta \in \mathbb{R}$ is a scale factor.

First, the task is to determine how affinity $\mathbf{A}$ transforms the length of $\mathbf{n}_1$ if $|\mathbf{n}_1| = |\mathbf{n}_2| = 1$. Introduce point $\mathbf{q} = \mathbf{p} + \delta\mathbf{n}_1$, where $\delta \in \mathbb{R}$ is an arbitrary scalar value. This new point determines an epipolar line in the second image as $\mathbf{l}'_2 = \mathbf{F}\mathbf{q} = \mathbf{F}(\mathbf{p}_1 + \delta\mathbf{n}_1) = [l'_{2,a} \quad l'_{2,b} \quad l'_{2,c}]^{\mathrm{T}}$. Scale β is given by distance d_2 between line $\mathbf{l}'_2$ and point $\mathbf{p}_2$ (see Fig. 4.1(b)). The calculation of distance d_2 is written as follows:

$$d_2 = \frac{|s_{1,a}x_2 + s_{2,b}y_2 + s_{3,c}|}{\sqrt{s_{1,a}^2 + s_{2,b}^2}}, \quad s_{i,k} = l_{2,k} + \delta f_{i1}n_{1,x} + \delta f_{i2}n_{1,y}, \quad i \in \{1, 2, 3\}, \quad k \in \{a, b, c\}. \tag{4.2}$$

Point $\mathbf{p}_2$ lies on $\mathbf{l}_2$, which can be written as $l_{2,a}x_2 + l_{2,b}y_2 + l_{2,c} = 0$. This fact reduces Eq. 4.2 to

$$d_2 = \frac{|\hat{s}_1x_2 + \hat{s}_2y_2 + \hat{s}_3|}{\sqrt{s_1^2 + s_2^2}}, \tag{4.3}$$

where $\hat{s}_i = \delta f_{i1}n_{1,x} + \delta f_{i2}n_{1,y}, i \in \{1, 2, 3\}$. To determine β, the introduced point $\mathbf{q}$ has to be moved infinitely close to $\mathbf{p}$ ($\delta \to 0$). The square of β is then written as

$$\beta^2 = \lim_{\delta \to 0} \frac{\delta^2}{d_2^2} = \lim_{\delta \to 0} \frac{s_1^2 + s_2^2}{|\hat{s}_1x_2 + \hat{s}_2y_2 + \hat{s}_3|^2}.$$

After elementary modifications, the formula for scale β is

$$\beta = \frac{\sqrt{l'_{1,a}l'_{1,a} + l'_{1,b}l'_{1,b}}}{(|\widetilde{s}_1x_2 + \widetilde{s}_2y_2 + \widetilde{s}_3|)},$$

where $\widetilde{s}_i = f_{i1}n_{1,x} + f_{i2}n_{1,y}, \ i \in \{1, 2, 3\}$. Therefore, *we can calculate β for unit length normals*.

Consider the case when normals are kept in their original form and not normalized ($|\mathbf{n}_1| \neq |\mathbf{n}_2| \neq 1$). The normalization indicates the following formula

$$\mathbf{A}^{-\mathrm{T}} \frac{\mathbf{n}_1}{|\mathbf{n}_1|} = \beta \mathbf{n}_2. \tag{4.4}$$

The epipolar line corresponding to point $\mathbf{p}_2$ is parameterized as $[\mathbf{l}_{2,a} \quad \mathbf{l}_{2,b} \quad \mathbf{l}_{2,c}] = \mathbf{F}[x_1 \quad y_1 \quad 1]^{\mathrm{T}}$. Therefore, its normal is: $\mathbf{n}_2 = \begin{bmatrix}\mathbf{l}_{2,a} & \mathbf{l}_{2,b}\end{bmatrix}^{\mathrm{T}} = (\mathbf{F}\begin{bmatrix}x_1 & y_1 & 1\end{bmatrix}^{\mathrm{T}})_{(1:2)}$. Similarly, $\mathbf{n}_1 = (\mathbf{F}^{\mathrm{T}}\begin{bmatrix}x_2 & y_2 & 1\end{bmatrix}^{\mathrm{T}})_{(1:2)}$. The denominator in Eq. 4.4 for computing β is rewritten as $|\mathbf{n}| = \sqrt{l_{1,a}^2 + l_{1,b}^2}$. The numerator is as follows:

$$\begin{aligned} \widetilde{s_1}x_2 + \widetilde{s_2}y_2 + \widetilde{s_3} = \\ n_{1,x}(f_{11}x_2 + f_{21}y_2 + f_{31}) + n_{1,y}(f_{12}x_2 + f_{22}y_2 + f_{32}) = \\ n_{1,x}^2 + n_{1,y}^2 = |\mathbf{n}_1|^2. \end{aligned}$$

Thus $\beta = \pm|\mathbf{n}_1|/|\mathbf{n}_1|^2 = \pm 1/|\mathbf{n}_1|$. Therefore, Eq. 4.4 is modified to $\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1 = \pm\mathbf{n}_2$.

Since the direction of the epipolar lines on the two images must be the opposite of each other, the positive solution is omitted. The final formula is: $\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1 = -\mathbf{n}_2$. □

4.2.2 Linear Equations

The normals of the epipolar lines are expressed from $\mathbf{F}$ as the first two coordinates of the epipolar lines: $\mathbf{n}_1 = (\mathbf{l}_1)_{(1:2)} = (\mathbf{F}^{\mathrm{T}}\mathbf{p}_2)_{(1:2)}$ and $\mathbf{n}_2 = (\mathbf{l}_2)_{(1:2)} = (\mathbf{F}\mathbf{p}_1)_{(1:2)}$ [88], where the lower indices select a subvector. Therefore, Eq. 4.1 is written as

$$\mathbf{A}^{-\mathrm{T}}(\mathbf{F}^{\mathrm{T}}\mathbf{p}_2)_{(1:2)} = -(\mathbf{F}\mathbf{p}_1)_{(1:2)}$$

and forms a system of linear equations consisting of two equations as follows:

$$(x_2 + a_{11}x_1)f_1 + a_{11}y_1f_2 + a_{11}f_3 + (y_2 + a_{21}x_1)f_4 + a_{21}y_1f_5 + a_{21}f_6 + f_7 = 0, \tag{4.5}$$

$$a_{12}x_1f_1 + (x_2 + a_{12}y_1)f_2 + a_{12}f_3 + a_{22}x_1f_4 + (y_2 + a_{22}y_1)f_5 + a_{22}f_6 + f_8 = 0. \tag{4.6}$$

Thus each local affine transformation *reduces the degrees-of-freedom by two*.

4.3 Accurate Closed-form Estimation of Local Affine Transformations Consistent with the Epipolar Geometry

This section addresses the problem of precise estimation of local affine transformations in rigid 3D scenes[1]. Computer vision problems which exploit local features, e.g. structure-from-motion, commonly rely on point-to-point correspondences. Using the full local affine transformation has only become more popular in the last decade. Matas et al. [91] showed that local affine transformations facilitate two-view matching. Köser and Koch [92] proved that the 3D camera pose estimation is possible if the corresponding affinity and location of a single patch are given. Köser [5] showed that 3D points can be precisely triangulated from local affinities. Bentolila et al. [94] proved that affine transformations give constraints for estimating the epipoles in the images. Current 3D reconstruction pipelines use point correspondences as well as patches [9], [66], [93] in order to compute realistic 3D models of

[1]The generalization to multiple rigid motions each satisfying a different epipolar constraint is straightforward.

real-world objects. If the epipolar geometry is known, a homography can be estimated from a single local affinity [30]. Barath et al. [32] showed that there is a one-to-one relationship between the surface normal and the local affinity.

The main goal of this section is to show how to optimally correct local affine transformations between two frames, in the least squares sense, if the fundamental matrix $\mathbf{F}$ *is known.* The fundamental matrix can either be estimated from the local affine transformations [9], [94] to be refined or from point-to-point correspondences [43]. In calibrated set-ups, $\mathbf{F}$ is available.

The refinement of the translation part has been solved by Hartley and Sturm [100] who exploit the fact that point locations have to satisfy the epipolar geometry: if a point is given in the first image, its correspondence in the second frame must lie on its epipolar line [88]. The closest, in the least squares sense, locations are computed as the roots of a polynomial of degree 6. The method proposed in this section can be seen as an extension of the Hartley and Sturm method as we consider the full local affinity and present two additional constraints induced by the epipolar geometry.

Local affine transformations are commonly provided by three types of affine-covariant detectors. The first group, including MSER [45], estimates full local affine transformations directly. The second group optimizes the initial estimates – both Harris-Affine [3] and Hessian-Affine [101] perform the so-called Baumberg iteration [102] in order to obtain high-quality affinities. Finally, some methods generate synthesized views related by affine transformations and feature detectors are applied to these images. By combining the estimates of the detector with the transformation related to the current synthetic view, a local affinity is given for each point correspondence. The most frequently used combined view synthesizer and feature detector is the Affine SIFT (ASIFT) [2]. However, affine version of commonly used detectors like SURF [96], ORB [103], BRISK [104], etc. can easily been constructed using the synthesizer part of ASIFT. Matching On Demand with view Synthesis [46] (MODS) is a recently proposed method that obtains a mixture of MSER, ORB and Hessian-Affine points and does as little view-synthesizing as required to detect a predefined number of point pairs.

The contributions of this section are: an algorithm to estimate an EG-L_2-Optimal (EG-L_2-Opt) affine transformation in the least squares (LSQ) sense by enforcing the constraints proposed in the previous section. It is also proven that the LSQ optimization of the parameters has geometric and algebraic interpretations. We show experimentally that the EG-L_2-Opt procedure improves the accuracy of the output of all affine-covariant feature detector. As a side-effect, we determine the accuracy of affine-covariant feature detectors using ground truth data.

4.3.1 EG-L_2-Optimal Local Affine Transformation

First, we discuss how to estimate an affine transformation at each corresponding point pair. Finally, the computation of the EG-L_2-Opt transformation is discussed.

Local affine transformation. It is an open question how to get a good quality affine transformation related to each point pair in a real-world environment. We propose to use affine-covariant feature detectors [101] which obtain both the point locations and the affine transformations at the same time. Possibilities include ASIFT [2], MODS [46], Harris-Affine [57], Hessian-Affine [57], etc. These feature detectors provide an affine transformation for every ith point $\mathbf{p}_k^i = [x_k^i \quad y_k^i]^\mathrm{T}$ ($i \in [1, n]$) in the kth

image ($k \in 1, 2$) as $\mathbf{A}_k^i$. The transformation $\mathbf{A}^i$ mapping $\mathbf{A}_1^i$ into $\mathbf{A}_2^i$ is obtained as

$$\mathbf{A}^i = \mathbf{A}_2^i (\mathbf{A}_1^i)^{-1}. \tag{4.7}$$

Affine compatibility – Translation. The last column of matrix $\mathbf{A}$ is responsible for the translation between the related point pair. It is shown by Hartley and Sturm [100] that it can be refined in an optimal way in the least squares sense. Their method minimizes the Euclidean distance between the original and refined positions. Then the resulting point locations are fully consistent with the epipolar geometry.

Affine compatibility – Orientation and Scale. The constraints regarding to the orientation and scale are encoded in $\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1 = -\mathbf{n}_2$ (Eq. 4.1) proposed in the previous section.

The EG-L_2-Opt affinity. Suppose that an observed affine transformation $\mathbf{A}'$ is given. Then let us denote that by

$$\mathbf{A}' = \begin{bmatrix} a'_{11} & a'_{12} \\ a'_{21} & a'_{22} \end{bmatrix}. \tag{4.8}$$

The task is to find an $\mathbf{A}$ where

$$|\mathbf{A} - \mathbf{A}'|^2 \tag{4.9}$$

is minimal and $\mathbf{A}^{-\mathrm{T}}\mathbf{n}_1 = \beta\mathbf{n}_2$. In order to avoid inversion, it can be reformulated as $\mathbf{n}_1 = \beta\mathbf{A}^{\mathrm{T}}\mathbf{n}_2$. Note that the geometric interpretation of the L_2 norm is discussed in Appendix D. As it was proven in the previous section condition

$$\mathbf{n}_1 - \beta\mathbf{A}^{\mathrm{T}}\mathbf{n}_2 = 0 \tag{4.10}$$

holds for local affinities consistent with the epipolar geometry and it is linear in the parameters of $\mathbf{A}$. Note that $\beta = -1$.

$$n_{1,x} - \beta n_{2,x} a_{11} - \beta n_{2,y} a_{21} = 0, \quad n_{1,y} - \beta n_{2,x} a_{12} - \beta n_{2,y} a_{22} = 0. \tag{4.11}$$

Let us introduce a cost function J applying the constraints defined in Eqs. 4.9, 4.11. Using Lagrange multipliers, the cost function is as follows:

$$J(\mathbf{A}, \lambda_{11}, \lambda_{12}) = \frac{1}{2}\sum_{i=1}^{2}\sum_{j=1}^{2}(a_{ij} - a'_{ij})^2 + \\ \lambda_{11}(n_{1,x} - \beta n_{2,x} a_{11} - \beta n_{2,y} a_{21}) + \lambda_{12}(n_{1,y} - \beta n_{2,x} a_{12} - \beta n_{2,y} a_{22}), \tag{4.12}$$

where λ_{11} and λ_{12} are the Lagrange multipliers. Eq. 4.9 yields non-negative values. Therefore, the optimal solution is given by the partial derivatives of J:

$$\frac{\partial J}{\partial a_{11}} = a_{11} - a'_{11} - \beta n_{2,x}\lambda_{11} = 0, \quad \frac{\partial J}{\partial a_{12}} = a_{12} - a'_{12} - \beta n_{2,x}\lambda_{12} = 0,$$
$$\frac{\partial J}{\partial a_{21}} = a_{21} - a'_{21} - \beta n_{2,y}\lambda_{11} = 0, \quad \frac{\partial J}{\partial a_{22}} = a_{22} - a'_{22} - \beta n_{2,y}\lambda_{12} = 0,$$
$$\frac{\partial J}{\partial \lambda_{11}} = n_{1,x} - \beta n_{2,x} a_{11} - \beta n_{2,y} a_{21} = 0, \quad \frac{\partial J}{\partial \lambda_{12}} = n_{1,y} - \beta n_{2,x} a_{12} - \beta n_{2,y} a_{22} = 0.$$

This is an inhomogeneous, linear system of equations which can be written in form $\mathbf{C}\mathbf{x} = \mathbf{b}$, where $\mathbf{x} = [a_{11} \quad a_{12} \quad a_{21} \quad a_{22} \quad \lambda_{11} \quad \lambda_{12}]^{\mathrm{T}}$, $\mathbf{b} = [a'_{11} \quad a'_{12} \quad a'_{21} \quad a'_{22} \quad -$

$n_{1,x} \quad -n_{1,y}]^{\mathrm{T}}$, and $\mathbf{C}$ are the vector of the unknown parameters, inhomogeneous part, and coefficient matrix, respectively. $\mathbf{C}$ is as follows:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & -\beta n_{2,x} & 0 \\ 0 & 1 & 0 & 0 & 0 & -\beta n_{2,x} \\ 0 & 0 & 1 & 0 & -\beta n_{2,y} & 0 \\ 0 & 0 & 0 & 1 & 0 & -\beta n_{2,y} \\ -\beta n_{2,x} & 0 & -\beta n_{2,y} & 0 & 0 & 0 \\ 0 & -\beta n_{2,x} & 0 & -\beta n_{2,y} & 0 & 0 \end{bmatrix}.$$

The solution is $\mathbf{x} = \mathbf{C}^{-1}\mathbf{b}$. See Alg. 3 for the pseudo-code of the proposed algorithm.

Algorithm 3 EG-L_2-Optimal Affine Transformation

procedure CORRECTAFFINETRANSFORMATION
  Input:
  $\mathbf{F}$ – fundamental matrix.
  $\mathbf{p}_1$, $\mathbf{p}_2$ – corresponding point pair.
  $\mathbf{A}'$ – measured affine transformation.
  Output:
  $\mathbf{A}$ – optimally refined affine transformation.
  Algorithm:
  $\mathbf{l}_1 := \mathbf{F}^{\mathrm{T}}\mathbf{p}_2$; $\mathbf{l}_2 := \mathbf{F}\mathbf{p}_1$; $\mathbf{n}_1 := [l_1^a; l_1^b]/|[l_1^a; l_1^b]|_2$; $\mathbf{n}_2 := [l_2^a; l_2^b]/|[l_2^a; l_2^b]|_2$;
  $s_1 := f_{11}n_1^x + f_{12}n_1^y$; $s_2 := f_{21}n_1^x + f_{22}n_1^y$; $s_3 := f_{31}n_1^x + f_{32}n_1^y$;
  $\beta := (1/\left|s_1x_2 + s_2y_2 + s_3\right|)\sqrt{l_2^a l_2^a + l_2^b l_2^b}$;
  $\mathbf{C} := eye(6, 6)$; $\mathbf{C}_{55} := 0$; $\mathbf{C}_{66} := 0$;
  $\mathbf{C}_{15} := -\beta n_2^x$; $\mathbf{C}_{26} := -\beta n_2^x$; $\mathbf{C}_{35} := -\beta n_2^y$; $\mathbf{C}_{46} := -\beta n_2^y$;
  $\mathbf{C}_{51} := -\beta n_2^x$; $\mathbf{C}_{62} := -\beta n_2^x$; $\mathbf{C}_{53} := -\beta n_2^y$; $\mathbf{C}_{64} := -\beta n_2^y$;
  $\mathbf{b} := [a'_{11}; a'_{12}; a'_{21}; a'_{22}; -n_1^x; -n_1^y]$;
  $\mathbf{x} := \mathbf{C}^{-1}b$;
  $\mathbf{A} := [x_1, x_2; x_3, x_4]$;

4.3.2 Experimental Results

First, we show how to get ground truth affine transformations. Then we test the proposed theory on both synthesized and real-world data.

Synthesized tests. For synthesized testing, two perspective cameras are generated by their projection matrices $\mathbf{p}_1$ and $\mathbf{p}_2$. Their positions are randomized in the plane $z = 60$ which is parallel to plane XY. Both cameras point towards the origin. Their common focal length and principal point are 600 and $[300 \quad 300]^{\mathrm{T}}$, respectively. Then 50 spatial points are generated on a random plane that passes through the origin, and the points are projected onto the cameras. The ground truth affine transformation related to each point is calculated using the plane parameters. Tests are repeated 500 times at every noise level.

Fig. 4.2 shows the mean (left) and median (right) distances of the original noisy transformations and that of the optimal ones w.r.t. the ground truth data. Zero-mean Gaussian noise is added to the elements of the affine transformations and point locations. The error (vertical axis) is the mean of the L_2-norms of the difference matrices of the obtained and ground truth data. The horizontal axis shows the σ value of the noise.

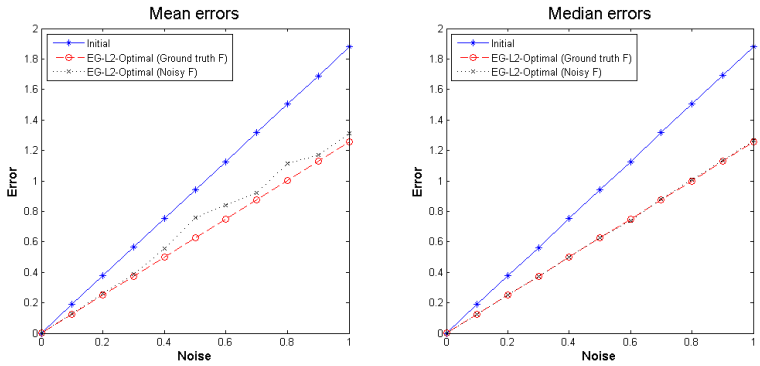


FIGURE 4.2: Error of the original and optimal affine transformations w.r.t. the noise level. The average L_2 distance from the ground truth transformation is plotted as a function of the σ value of the Gaussian noise (in pixels). The noise is added to the affine parameters and point locations. **(Red curve)** The ground truth **F** is used. **(Black curve) F** is estimated using the noisy point correspondences by the normalized 8-point algorithm followed by a Levenberg-Marquardt optimization minimizing the symmetric epipolar error. In the median figure, the black and red curves coincide.

The red curve shows the error if the ground truth fundamental matrix is used. For the black curve, the fundamental matrix is estimated using the noisy point locations by the normalized 8-point algorithm followed by Levenberg-Marquardt optimization minimizing the symmetric epipolar error. The refined transformations are closer to the ground truth matrices than the original ones. There is no significant difference between the median and mean plots and between results obtained on the ground truth and the estimated fundamental matrix.

The processing time of the proposed method is negligible since it consists of a few operations. It is calculated in C++ in around 0.04 milliseconds per point on a 2.3 GHz PC.

Tests on Real Data. The proposed theory is tested on the annotated AdelaideRMF dataset[2] and on image pairs `graffiti`[3], `stairs` and `glasscasea` (see Fig. 4.3). In the last three pairs, we manually marked point correspondences and assigned them to planes. The ground truth homographies are computed using the annotated point correspondences.

Several affine-covariant feature detectors are run on all image pairs. The following affine-covariant detectors are applied: AAKAZE, ABRISK, AORB, ASIFT, ASURF, AHessian-Affine[4], MODS[5], MSER, Harris-Affine and Hessian-Affine[6].

Correspondences of features points obtained by matching [95] are assigned to the closest annotated homography. The distance between a point pair and a homography is defined as the re-projection error ($\mathbf{Hp}_1 \sim \mathbf{p}_2$). If a correspondence is farther from its closest homography than 1.0 px, it is discarded from the evaluation since the ground truth affine transformation for such correspondence can not be calculated. For the remaining correspondences, ground truth affine transformations are

[2]Available at `http://cs.adelaide.edu.au/~hwong/doku.php?id=data`

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

[4]ASIFT is downloaded from `http://www.ipol.im/pub/art/2011/my-asift`. The "A-forms" of AKAZE, BRISK, ORB, SIFT, SURF, Hessian-Affine are obtained by replacing SIFT in the view-synthesizer.

[5]MODS is downloaded from `http://cmp.felk.cvut.cz/wbs`

[6]MSER, Har-Aff, and Hes-Aff downloaded from `http://www.robots.ox.ac.uk/~vgg/research/`

TABLE 4.1: Errors of the affine-covariant feature detectors "Observed" and their "EG-L_2-Opt" corrections. The error is the mean of the L_2-norms of the difference matrices of the obtained and ground truth affine transformations. Test pairs: (a) `hartley`, (b) `johnsonnb`, (c) `neem`, (d) `sene`, (e) `oldclassicswing`, (f) `ladysymon` (g) `graffiti` (h) `stairs` (i) `glasscasea`

Detector		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	avg	med
AAKAZE	Observed	0.26	0.30	0.17	0.30	0.26	0.18	0.25	0.62	0.38	0.30	0.26
	EG-L_2-Opt	0.21	0.22	0.12	0.19	0.19	0.14	0.16	0.54	0.26	0.23	0.19
ABRISK	Observed	0.28	0.33	0.27	0.38	0.28	0.30	0.28	1.31	0.31	0.42	0.30
	EG-L_2-Opt	0.21	0.25	0.19	0.24	0.22	0.18	0.18	0.50	0.20	0.24	0.21
AHES-AFF	Observed	0.19	0.23	0.18	0.20	0.14	0.17	0.21	0.24	0.22	0.20	0.20
	EG-L_2-Opt	**0.14**	0.17	0.11	**0.13**	**0.09**	0.11	0.13	**0.14**	**0.15**	**0.13**	**0.13**
AORB	Observed	0.34	0.34	0.15	0.45	0.23	0.24	0.27	-	0.28	0.29	0.28
	EG-L_2-Opt	0.27	0.28	0.10	0.29	0.17	0.18	0.18	-	0.20	0.20	0.19
ASIFT	Observed	0.27	0.28	0.27	0.26	0.21	0.22	0.27	0.23	0.29	0.26	0.27
	EG-L_2-Opt	0.20	0.21	0.15	0.17	0.14	0.17	0.16	0.17	0.18	0.17	0.17
ASURF	Observed	0.23	0.27	0.17	0.30	0.22	0.17	0.25	0.26	0.27	0.24	0.25
	EG-L_2-Opt	0.18	0.20	0.11	0.21	0.16	0.12	0.17	0.18	0.19	0.18	0.18
HAR-AFF	Observed	0.24	0.25	0.15	0.24	0.16	0.27	0.20	0.38	0.28	0.24	0.24
	EG-L_2-Opt	0.18	0.18	**0.09**	0.19	0.12	0.19	0.13	0.35	0.17	0.16	0.18
HES-AFF	Observed	0.24	0.22	0.20	0.22	0.13	0.20	0.19	-	0.24	0.21	0.21
	EG-L_2-Opt	0.17	**0.16**	0.10	0.17	**0.09**	**0.09**	**0.12**	-	**0.15**	**0.13**	0.14
MODS	Observed	0.29	0.40	0.23	0.31	0.26	0.25	0.61	0.24	0.47	0.34	0.29
	EG-L_2-Opt	0.20	0.25	0.13	0.22	0.19	0.17	0.42	0.19	0.32	0.23	0.20
MSER	Observed	0.42	0.69	0.46	0.34	0.29	0.31	0.42	0.51	0.34	0.42	0.42
	EG-L_2-Opt	0.24	0.32	0.23	0.25	0.20	0.22	0.25	0.31	0.21	0.25	0.24

TABLE 4.2: The average number of inliers – correspondences lying on an annotated homography – for different feature detectors. Processing times in seconds on an Intel Core4Quad 2.33 GHz PC with 4 GByte memory using only a single core.[7]

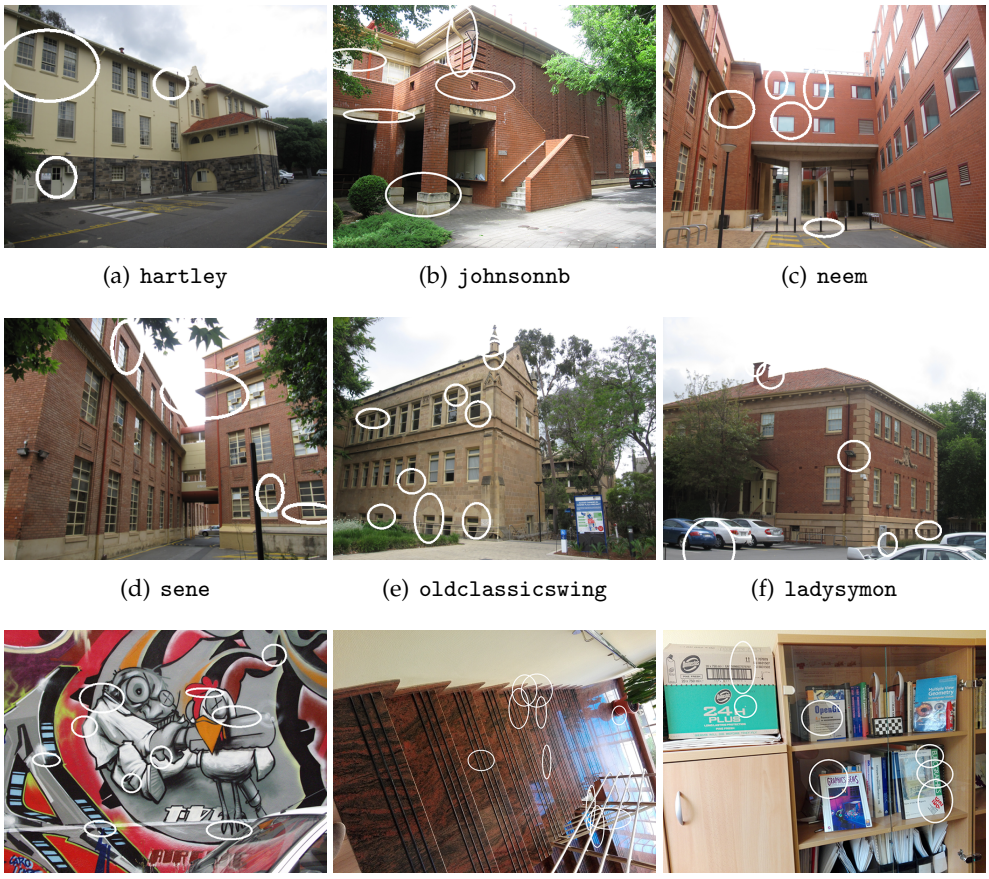
	AAKAZE	ABRISK	AHES-AFF	AORB	ASIFT	ASURF	HAR-AFF	HES-AFF	MODS	MSER
Inliers	239	110	1 420	145	2 082	837	64	73	941	78
Time	81.91	81.38	89.30	86.39	81.34	84.00	4.10	3.22	52.92	0.41

calculated using Eqs. 4.7. Fundamental matrices are computed by the normalized 8-point algorithm followed by a numerical refinement stage minimizing symmetric epipolar error by Levenberg-Marquardt optimization [85].

The errors are shown in Table 4.1. The error is the mean of the L_2-norms of the difference matrices of the obtained and ground truth data. Each column represents a test pair except the last two ones which show the mean and median errors. The corresponding odd and even rows visualize the mean error of the observed affine transformations given by each feature detector and that of the refined, EG-L_2-Opt ones. The error metric is the same as used for the synthesized tests. Every method is applied using their default parameterization. The median values show the same trend. The most important conclusion of these tests is that *the refined, EG-L_2-Opt affine transformations are always more accurate than the observed ones.*

Hessian-Affine augmented with the view-synthesizer of ASIFT (denoted by AHES-AFF) obtains the most accurate affinities (see Table 4.1) and provides many point correspondences as well (see Table 4.2). If the required number of correspondences needs not be high, Hessian-Affine without view-synthesizing might be the method of choice since it is significantly faster and its accuracy is nearly the same.

[7] Information in Table 4.2 is not assessing the precision of affine transformation, the main topic of the section. It complements Table 4.1 in providing broader characterization of detector performance.



(a) hartley (b) johnsonnb (c) neem

(d) sene (e) oldclassicswing (f) ladysymon

(g) graffiti (h) stairs (i) glasscasea

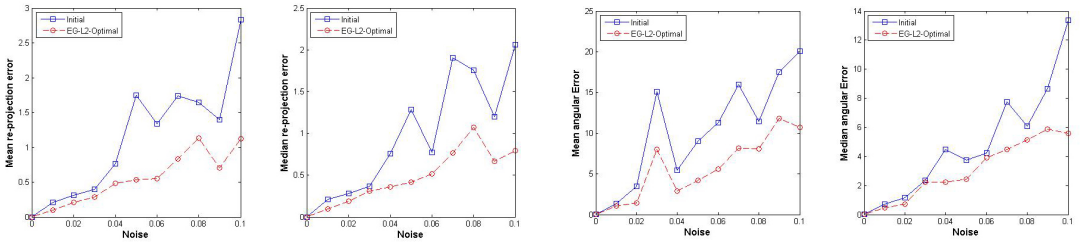
FIGURE 4.3: The first frames of the selected image pairs with a few local affinities each represented by an ellipse.

4.3.3 Improvements on Homography and Surface Normal Estimates

This section presents experiments showing that EG-L_2-Opt affinities lead to more accurate homography and surface normal estimates.

For homography estimation the same synthetic scene is constructed as for the previous synthesized tests: a random plane is generated and sampled at ten locations which are projected onto the cameras. The method proposed by Koeser [5] is applied to one of the ten correspondences and the related affinity. Tests are repeated 500 times for every noise level. Fig 4.4(a) shows that homographies calculated from the EG-L_2-Opt refined data are the most accurate ones. The error metric is the mean re-projection error (in pixels) computed for the point locations.

For surface normal estimation, the technique proposed recently by Barath et al. [32] is performed. In our tests, the same testing environment is used as proposed in [32] and FNE normal estimator is applied to both the initial and EG-L_2-Opt affinities. Fig. 4.4(b) confirms that the proposed technique makes the surface normals more accurate.



(a) (b)

FIGURE 4.4: Mean, (a) left, and median, (a) right, re-projection errors (in pixels) of the homography estimation [5] applied to the noisy and the EG-L_2-Opt refined affinities. Mean, (b) left, and median, (b) right, angular errors (in degrees) of the surface normals estimated from the initial and EG-L_2-Opt refined affinities. The errors are plotted as the function of the σ value of the isotropic 6D zero-mean Gaussian noise.

4.3.4 Summary

We showed how to improve the accuracy of a local affine transformation obtained by an affine-covariant feature detector by considering the epipolar constraint. The proposed algorithm is optimal in the least squares sense. Its computational cost is negligible. The proposed least squares minimization has an intuitive geometric interpretation.

The introduced EG-L_2-Opt procedure is validated on real-world image pairs. It improves the accuracy of all tested affine-covariant detectors. On average, the error of the refined affinities is reduced to about 65%. The EG-L_2-Opt affinities improve the accuracy of surface normal and homography estimates as well.

As a side-effect, the experiments quantitatively compared the precision of affine-covariant feature detectors. The Hessian-Affine detector combined with the view-synthesizer of ASIFT obtains the most accurate affinities.

4.4 Essential Matrix Estimation

The estimation of epipolar geometry between a pair of images is a key-problem for the recovery of relative camera motion and has been studied for decades. Luong and Fougeras showed that this relationship can be described by the so-called 3×3 fundamental matrix [105]. Since then, several approaches have been proposed to cope with this problem. The well-known seven and eight-point algorithms [43] need no a priori information about the camera parameters to estimate the fundamental matrix from point correspondences. However, exploiting the intrinsic camera parameters (focal length, principal point, etc.), the estimation can be done using six [106]–[109] or five correspondences [110]–[113].

In this section, we assume intrinsic parameters and two affine correspondences to be known between a pair of images to recover the essential matrix. An affine correspondence consists of a point pair and the related local affine transformation mapping the infinitesimally close vicinity of the point in the first image to that of the second one. Nowadays, several approaches are available for the estimation of local affine transformations. Beside the well-known affine-covariant feature detectors [57] such as MSER, Hessian-Affine, Harris-Affine, there are some modern ones based on view-synthesizing, e.g. ASIFT [2], ASURF or MODS [46]. They obtain accurate local affinities and many correspondences by transforming the original image with an affine transformation to create a synthetic view. Then a feature detector is applied to

the warped images. The final local affinity related to a point pair is estimated as the combination of the transformation regarding to the current synthetic view and the affine transformation which the applied detector obtains. MODS yields a mixture of ORB [103], MSER [45] and Hessian-Affine points and does as few view-synthesizing as necessary to obtain a predefined number of points.

Using local affinities for fundamental matrix estimation is not a novel idea. Perdoch et al. [114] and Chum et al. [7] proposed methods using two and three affine correspondences, respectively. Even so, they provide only rough estimations since they generate point correspondences exploiting local affinities and apply the six [106] and eight-point algorithms [43], respectively. Nevertheless, local affinities cannot generate point correspondences since they are defined as the partial derivative of the related homography. Thereby, they are valid only infinitesimally close to the observed point [32]. The obtained results of [114] and [7] are approximations – the error is not zero even for noise-free input. Bentolila et al. [8] showed that two affine transformations yields three explicit conic constraints on fundamental matrix estimation and three affine correspondences are enough. Recently, an approach is proposed by Raposo and Barreto [9] which is slightly similar to the base algorithm proposed in this section. Providing a derivation on the basis of homographies and applying the solver of the five-point algorithm [111], they estimate the epipolar geometry using two affine correspondences. Unlike them, we show that this relationship can be formalized directly, considering the way how a local affinity affects the epipolar lines. Through the proposed formulation it can straightforwardly be seen that the relationship holds for arbitrary camera models. Additionally, the solver we propose leads to results superior to [9] as it is demonstrated later.

The contributions of this section are as follows: **(1)** Using the constraints proposed previously, the essential matrix is estimable exploiting two affine correspondences. The method is generalized to solve the over-determined case as well and provides only one globally optimal essential matrix. It has been demonstrated both on synthesized and real world test that the algorithm is superior to the state-of-the-art in term of the accuracy of the estimated camera motion. **(2)** It is shown how the multiplication of the point locations by the camera matrices modifies the local affinities, thus making the method applicable to image pairs captured by different camera set ups. The normalization technique of Hartley [47] is extended to affine transformations to achieve numerically stable estimates in the over-determined case.

4.4.1 Preliminaries

The ith element of the essential $\mathbf{E}$ and fundamental matrices $\mathbf{F}$ in row-major order is denoted as e_i and f_i, respectively ($i \in [1, 9]$). The relationship of them is $\mathbf{F} = \mathbf{K}_2^{-\mathrm{T}}\mathbf{E}\mathbf{K}_1^{-1}$, where $\mathbf{K}_1$ and $\mathbf{K}_2$ are the intrinsic parameters of the two cameras. Fundamental matrix $\mathbf{F}$ ensures the epipolar constraint as $\mathbf{p}_2^{\mathrm{T}}\mathbf{F}\mathbf{p} = \mathbf{p}_2^{\mathrm{T}}\mathbf{K}_2^{-\mathrm{T}}\mathbf{E}\mathbf{K}_1^{-1}\mathbf{p}_1 = 0$. In the rest of the section, we assume that points $\mathbf{p}_1$ and $\mathbf{p}_2$ have been premultiplied by $\mathbf{K}_1$ and $\mathbf{K}_2$. This assumption simplifies the epipolar constraint to

$$\mathbf{q}_2^{\mathrm{T}}\mathbf{E}\mathbf{q}_1 = 0, \tag{4.13}$$

where $\mathbf{q}_1$ and $\mathbf{q}_2$ are the points multiplied by $\mathbf{K}_1$ and $\mathbf{K}_2$. Two additional constraints can be considered on the essential matrix $\mathbf{E}$. The first one is called trace constraint [43], it is as follows:

$$2\mathbf{E}\mathbf{E}^{\mathrm{T}}\mathbf{E} - tr(\mathbf{E}\mathbf{E}^{\mathrm{T}})\mathbf{E} = 0. \tag{4.14}$$

This matrix equation yields nine polynomial equations for the elements of $\mathbf{E}$. The second restriction ensures that the determinant of the essential matrix must be zero:

$$\det(\mathbf{E}) = 0. \tag{4.15}$$

These two properties will help us to recover the essential and fundamental matrices exploiting two affine correspondences.

4.4.2 Two-point Algorithm

First, we exploit the formulas, which the relationship of an affine transformation and the epipolar geometry provides (Eqs. 4.5, 4.6), to estimate the essential matrix using two affine correspondences. Then we show the effect of differing intrinsic camera parameters and that of the point normalization.

The Proposed Solver. Here, the proposed 2-point algorithm based on the introduced constraints is discussed. Suppose that two point pairs $(\mathbf{p}_1, \mathbf{p}_2)$ and $(\mathbf{p}'_1, \mathbf{p}'_2)$ and the related affinities $\mathbf{A}$ and $\mathbf{A}'$ are given. Fig. 4.5 shows how $\mathbf{A}$ and $\mathbf{A}'$ transform the infinitesimally close vicinities of the points from the first to the second images.

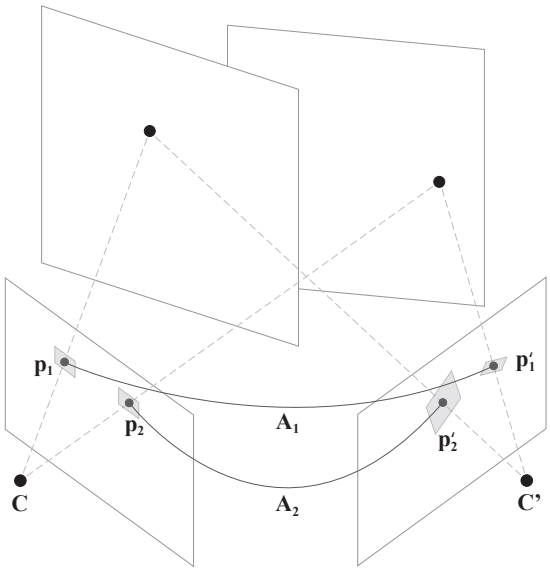


FIGURE 4.5: Projections of two spatial points are given on cameras $\mathbf{K}_1$ and $\mathbf{K}_2$. Corresponding local affine transformations $\mathbf{A}$ and $\mathbf{A}'$ transforms the infinitesimally close vicinities of point pairs $(\mathbf{p}_1, \mathbf{p}_2)$ and $(\mathbf{p}'_1, \mathbf{p}'_2)$ between the image pair.

For the ith ($i \in \{1, 2\}$) correspondence, the combination of formulas Eqs. 4.5, 4.6, and Eq. 4.13 can be written as $\mathbf{C}_i\mathbf{x} = 0$, where $\mathbf{x} = [e_1\ e_2\ e_3\ e_4\ e_5\ e_6\ e_7\ e_8\ e_9]^\mathrm{T}$ is the vector of the unknown elements of the essential matrix. Matrix $\mathbf{C}_i$ is the coefficient matrix consisting of three rows, where the first two are the coefficients of Eqs. 4.5, 4.6. The third one contains the coefficients related to the well-known formula $\mathbf{p}_2^\mathrm{T}\mathbf{E}\mathbf{p} = 0$. Note that the algorithm can straightforwardly be extended to $n > 2$ points by concatenating their $\mathbf{C}_i$ matrices. If at least three correspondences are given, the solution vector $\mathbf{x}$ is obtained as the eigenvector related to the smallest eigenvalue of matrix $\mathbf{C}^\mathrm{T}\mathbf{C}$, where matrix $\mathbf{C}$ is the concatenated coefficient matrix and of size $3n \times 9$.

Considering the two point case, matrix $\mathbf{C}_1$ is of size 6×9 as $\mathbf{C} = \begin{bmatrix}\mathbf{C}_1^\mathrm{T} & \mathbf{C}_2^\mathrm{T}\end{bmatrix}^\mathrm{T}$. Its null space is 3-dimensional, therefore, the solution of the equation system is given

by the linear combination of the three corresponding singular vectors of K as

$$\mathbf{x} = \alpha\mathbf{d} + \beta\mathbf{e} + \gamma\mathbf{f}, \tag{4.16}$$

where $\mathbf{d}$, $\mathbf{e}$, and $\mathbf{f}$ are the singular vectors. Parameters α, β, and γ are unknown non-zero scalar values. These scalars are defined up to a common scale, therefore, one of them can be chosen to an arbitrary value. In the proposed algorithm, $\gamma = 1$.

By substituting this formula to the trace (Eq. 4.14) and determinant (Eq. 4.15) constraints, ten polynomial equations are given. They can be formed as $\mathbf{Q}\mathbf{y} = \mathbf{b}$, where $\mathbf{Q}$ and $\mathbf{b}$ are the coefficient matrix and the inhomogeneous part (coefficients of monomial 1), respectively. Vector $\mathbf{y} = \begin{bmatrix}\alpha^3 & \beta^3 & \alpha^2\beta & \alpha\beta^2 & \alpha^2 & \beta^2 & \alpha\beta & \alpha & \beta\end{bmatrix}$ consists of the monomials of the system. $\mathbf{Q}$ is of size 10×9, therefore, the system is solvable and overdetermined since ten equations are given for nine unknowns. Its optimal solution in least squares sense is given by $\mathbf{y} = \mathbf{Q}^{\dagger}\mathbf{b}$, where matrix $\mathbf{Q}^{\dagger}$ is the Moore-Penrose pseudo-inverse of matrix $\mathbf{Q}$. Since the solution vector $\mathbf{y}$ consists of different powers of α and β, the one is chosen for each, for which the obtained (from Eq. 4.16) essential matrix minimizes Eqs. 4.5, 4.6. The fundamental matrix is finally calculated as $\mathbf{F} = \mathbf{K}_2^{-\mathrm{T}}\mathbf{E}\mathbf{K}_1^{-1}$.

Transformation of Local Affinities by the Camera Matrices. The aim of this section is to show how the multiplication of the point coordinates by the intrinsic parameters modifies the corresponding local affinities. Unlike to the rest of the section, we assume here that points $\mathbf{p}_1$ and $\mathbf{p}_2$ are not multiplied by $\mathbf{K}_1^{-1}$ and $\mathbf{K}_2^{-1}$. The original relationship between the affine parameters comes from Eq. 4.1 by replacing the normals by $\mathbf{F}^{\mathrm{T}}\mathbf{p}_2$ and $\mathbf{F}\mathbf{p}_1$ as follows:

$$(\hat{\mathbf{A}}^{-\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{p}_2)_{(1:2)} = -(\mathbf{F}\mathbf{p})_{(1:2)}, \tag{4.17}$$

where $\hat{\mathbf{A}}$ is of size 3×3 as follows:

$$\hat{\mathbf{A}} = \begin{bmatrix}\mathbf{A} & 0 \\ 0 & 1\end{bmatrix}.$$

Because of $\mathbf{F} = \mathbf{K}_2^{-\mathrm{T}}\mathbf{E}\mathbf{K}_1^{-1}$, Eq. 4.17 is modified as

$$(\hat{\mathbf{A}}^{-\mathrm{T}}\mathbf{K}^{-\mathrm{T}}\mathbf{E}^{\mathrm{T}}\mathbf{K}_2^{-1}\mathbf{p}_2)_{(1:2)} = -(\mathbf{K}_2^{-\mathrm{T}}\mathbf{E}\mathbf{K}_1^{-1}\mathbf{p})_{(1:2)}.$$

Let us denote $\mathbf{K}_2^{-1}\mathbf{p}_2$ and $\mathbf{K}_1^{-1}\mathbf{p}_1$ with $\mathbf{q}_2$ and $\mathbf{q}_1$, respectively. After elementary modifications, it can be written as

$$(\mathbf{E}^{\mathrm{T}}\mathbf{q}_2)_{(1:2)} = -(\mathbf{K}_1^{\mathrm{T}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{K}_2^{-\mathrm{T}}\mathbf{E}\mathbf{q}_1)_{(1:2)}.$$

Therefore, due to the transformation of the intrinsic parameters, the original local affinity $\mathbf{A}$ must be modified as

$$\widetilde{\mathbf{A}} = (\mathbf{K}_2^{-1}\hat{\mathbf{A}}\mathbf{K}_1)_{(1:2,1:2)}. \tag{4.18}$$

However, matrix $\mathbf{A}$ remains the same if $\mathbf{K}_1 = \mathbf{K}_2$ and the shear is zero for both cameras.

Normalization of Affine Parameters. It is well-known that numerical instability makes the normalization of the input data essential [47]. It is shown how to calculate the normalized affine transformation $\widetilde{\mathbf{A}}$ in this section if the overdetermined case is considered. Let us denote the normalizing transformations in the two images with $\mathbf{T}_1$ and $\mathbf{T}_2$ which translate the point sets into the origin and their mean distance from that to$\sqrt{2}$. The normalization of the point coordinates (which have been premultiplied by the intrinsic parameters) is trivial as $\widetilde{\mathbf{p}} = \mathbf{T}_1\mathbf{p}_1$ and $\widetilde{\mathbf{p}}' = \mathbf{T}_2\mathbf{p}_2$ [43]. The normalized essential matrix can be calculated from the original as follows: $\widetilde{\mathbf{E}} = \mathbf{T}_2^{-\mathrm{T}}\mathbf{E}\mathbf{T}_1^{-1}$. After point normalization the relationship of the essential matrix and the affine transformation (Eq. 4.2.2) is modified as follows:

$$(\hat{\mathbf{A}}^{-\mathrm{T}}(\mathbf{T}_2^{\mathrm{T}}\widetilde{\mathbf{E}}\mathbf{T}_1)^{\mathrm{T}}\mathbf{p}_2)_{(1:2)} = -(\mathbf{T}_2^{\mathrm{T}}\widetilde{\mathbf{E}}\mathbf{T}_1\mathbf{p})_{(1:2)},$$

where $\hat{\mathbf{A}}$ is the same 3×3 matrix as in the previous section. After elementary modifications, it can be written as

$$(\widetilde{\mathbf{E}}^{\mathrm{T}}\mathbf{T}_2\mathbf{p}_2)_{(1:2)} = -(\mathbf{T}_1^{-\mathrm{T}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{T}_2^{\mathrm{T}}\widetilde{\mathbf{E}}\mathbf{T}_1\mathbf{p})_{(1:2)}.$$

Thus

$$\widetilde{\mathbf{A}}^{\mathrm{T}} = (\mathbf{T}_1^{-\mathrm{T}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{T}_2^{\mathrm{T}})_{(1:2,1:2)}.$$

The normalized affine transformation $\widetilde{\mathbf{A}}$ is calculated as

$$\widetilde{\mathbf{A}} = (\mathbf{T}_2\hat{\mathbf{A}}\mathbf{T}_1^{-1})_{(1:2,1:2)}.$$

This equation is the same as Eq. 4.18 and holds for all transformations that can be written by 3×3 matrices e.g. the camera intrinsic parameters and the normalizing transformations in the image space.

The affinities used during the estimation are normalized by both the normalizing transformations and the intrinsic parameters. Thus affine transformation $\mathbf{A}$ is modified as follows:

$$\overline{\mathbf{A}} = (\mathbf{T}_2\mathbf{K}_2^{-1}\hat{\mathbf{A}}\mathbf{K}\mathbf{T}_1^{-1})_{(1:2,1:2)}$$

Note that the proposed normalization is possible only if more than two correspondences are given. Otherwise, only the normalization by the intrinsic parameters is required.

4.4.3 Experimental Results

Validation on Synthesized Tests. In order to test the proposed method in a fully controlled synthetic environment, two perspective cameras are generated by their projection matrices $\mathbf{p}_1$ and $\mathbf{p}_2$. Their common intrinsic parameters are focal lengths $f_x = f_y = 600$ and principal point $\mathbf{p}_0 = \begin{bmatrix}300 & 300\end{bmatrix}^{\mathrm{T}}$. For the tests, three types of camera motions are considered: forward, sideways and random motions. The lengths of these motions are 2 and the distances of the plane origins from the camera centers are 10 along axis z and around 0.1 along axes x and y. We do not check whether a point is visible on both cameras or not since it does not affect the results of the methods. Having more than one plane is required to get a non-degenerate set up, thus points are sampled on 100 different random planes and projected onto the cameras. Zero-mean Gaussian-noise is added to the point locations. Homography is calculated using the plane parameters [43]. The affine transformation related to each point pair is calculated exploiting the noisy coordinates and the ground truth

homography. The obtained essential matrices are decomposed into translation and rotation components [43] and compared to the ground truth motion.

In Fig. 4.6, we compare four methods: the proposed algorithm applied to two correspondences (Proposed), the normalized version of the proposed method applied to five point pairs (Normalized Prop.), the five-point algorithm [111] (Nistér) and the technique proposed in [9] (Raposo et al.). The top row shows the mean error (vertical axis) of the obtained rotation matrices plotted as the function of the noise σ (horizontal axis). The error is computed as the Frobenious-norm of the difference matrix between the ground truth and estimated rotation matrices. The bottom row reports the quality of the estimated translation vectors. The mean angular error (in radians, vertical axis) w.r.t. the ground truth translation is plotted as the function of the noise σ (horizontal axis).

For Fig. 4.6(a), forward motion and no rotation is applied to the cameras. It can be seen that the proposed method exploiting two correspondences outperforms both the five-point algorithm and that of Raposo et al. The translation vector obtained by the normalized algorithm is sensitive to this kind of motion, however, the estimated rotation matrix is the most accurate. Fig. 4.6(b) reports the error if only sideways motion is considered. In these tests, the proposed method and that of Raposo et al. achieved similar accuracy. The normalized version is superior to all competitor methods in both terms. For Fig. 4.6(c) random motion is applied, the rotation obtained by the proposed two-point algorithm outperforms both the methods of [111] and [9], while achieving similar results to [9] for the translation vector. The normalized algorithm provided the most accurate results in both aspects. Fig. 4.6(d) reports the results for nearly planar scenes. Only a small Gaussian-noise with 10^{-5} standard deviation is added to the plane tangents having the same base point. It can be seen that the 5-point algorithm leads to the most accurate translation vectors, however, the proposed methods outperform the competitor ones for estimating the camera rotation.

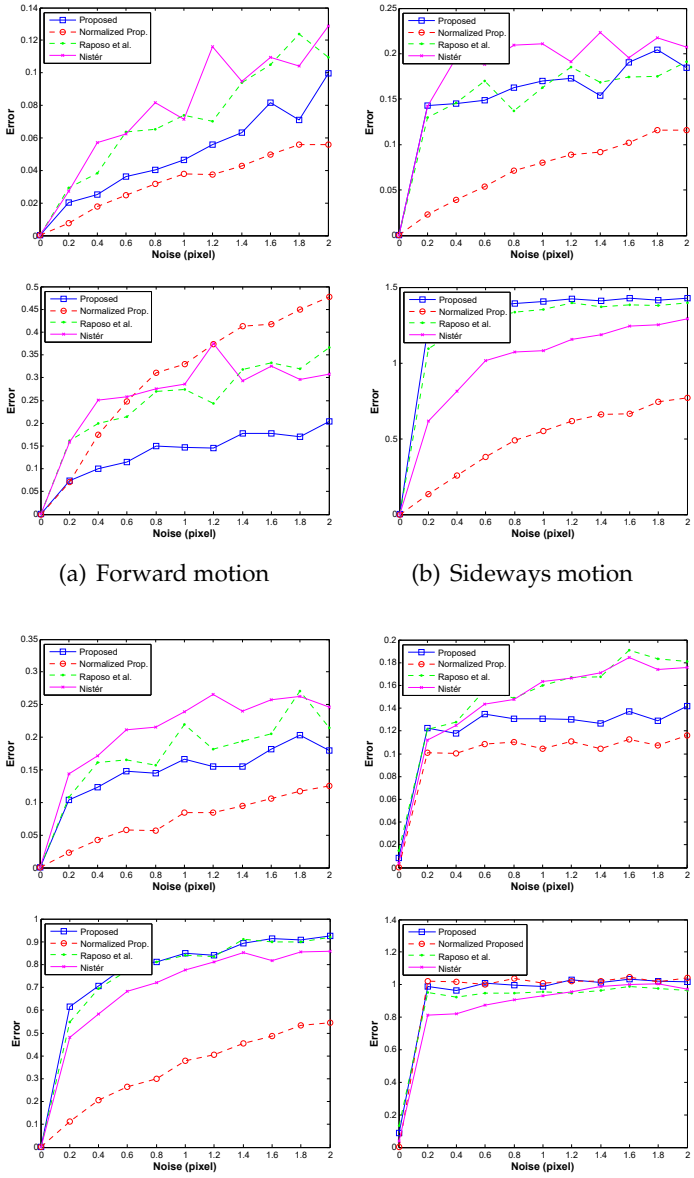
Concluding the synthesized tests, the proposed algorithm (without normalization) outperformed the competitor ones in four out of the eight test cases and achieve similar results in the remaining ones. The normalized version applied to five correspondences is superior to all methods in both terms except two test cases.

Real World Experiments. To validate the 2-point method on real world photos, the Daisy dataset [115] is exploited. The contained images are of resolutions 512×384 up to 1024×768 with known intrinsic camera matrices. To acquire local affine transformations and point correspondences, an affine-covariant feature detector is applied to each image pair. There are several possible options such as ASIFT [2], ASURF, AORB, MODS [46], Harris-Affine, Hessian-Affine, MSER [57]. According to our experience [28] the most accurate one is Hessian-Affine detector combined with the view-synthesizer of ASIFT. We call it AHessian-Affine.

For the determination of the essential matrix, PROSAC[8] [116] is selected as robust estimator since it is as accurate as the widely-used RANSAC but significantly faster. The applied distance function is the Sampson-error and the threshold of PROSAC is $\epsilon = 3.0$ (in pixels). PROSAC selects two affine correspondences in each iteration (minimal subset) and creates a hypothesis by the proposed technique.

Fig. 4.7 shows example results. The two points which are scored the best for hypothesis creation by PROSAC are painted by red circles in each image pair. The epipolar lines related to 50 random inliers are drawn by colors.

[8] Available at https://github.com/erfannoury/sac



(a) Forward motion (b) Sideways motion

(c) Random motion (d) Nearly planar scene

FIGURE 4.6: Plots (a)–(d) represent camera motions: (a) pure forward, and (b) sideways motion, (c) random motion, and (d) nearly planar scene with cameras having random motion. The top row in each plot pair is the error (vertical axis) of the rotation matrix, i.e. the Frobenious-norm of the difference matrix of the ground truth rotation and the obtained one. The bottom row is the angular error (in radians, vertical axis) of the estimated translations. The horizontal axis reports the noise (in pixels) added to the coordinates and the affine parameters. Errors are computed as the mean of 1000 runs on each noise level. The reported algorithms: the proposed one applied to a minimal sample (Proposed), the normalized version of the proposed method applied to five correspondences (Normalized Prop.), the technique of Raposo and Barreto [9], and the 5-point algorithm proposed by David Nister [111].

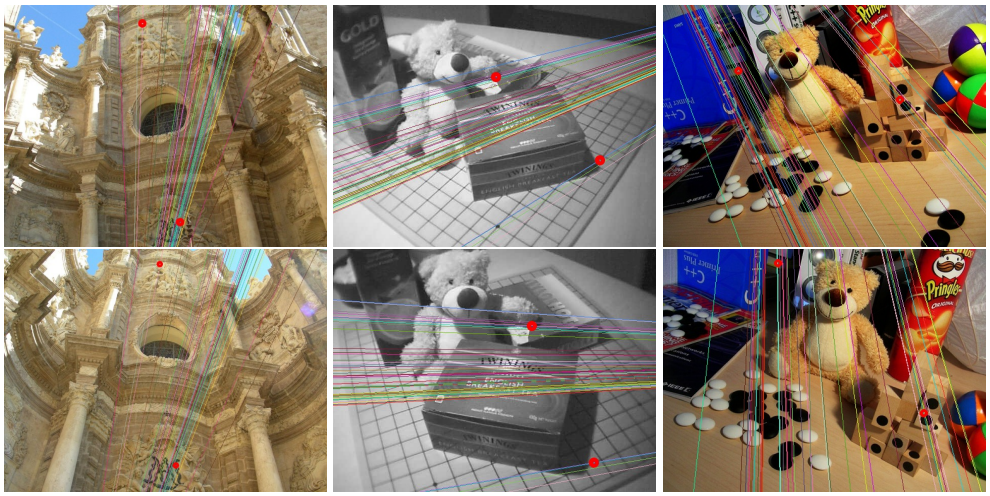


FIGURE 4.7: The results of the 2-point algorithm on real image pairs (columns). Red circles visualize the two points scored the best by PROSAC. Epipolar lines of 50 random inliers are drawn to the images using colors.

TABLE 4.3: Comparison of methods w.r.t. iteration number and processing time of PROSAC [116]. The name of each test pair and the correspondence number N are written in the first two columns. Other columns are the mean iteration numbers and processing times (in seconds) of 500 runs on Daisy dataset. Competitor algorithms are: 3-point [8], 5-point [111], 6-point [109], normalized 7-point [43], and 8-point [43] algorithms. Each method is included into PROSAC with threshold $\epsilon = 3.0$ pixels.

	N	**2-pt**		3-pt		5-pt		6-pt		7-pt		8-pt	
Daisy 1	1614	15	**0.03**	89	2.92	32	0.06	**12**	0.05	49	0.06	28	**0.03**
Daisy 2	2223	49	0.10	124	4.00	43	**0.09**	**28**	0.13	113	0.17	77	0.12
Daisy 3	848	97	0.08	348	9.63	**78**	**0.07**	85	0.16	453	0.27	376	0.24
Daisy 4	569	**50**	**0.03**	231	7.42	59	0.04	57	0.08	209	0.09	188	0.09
Daisy 5	1072	**44**	**0.05**	187	8.09	138	0.14	74	0.17	96	0.07	79	0.07
Daisy 6	4101	**90**	**0.26**	606	16.13	175	0.51	129	0.98	437	1.32	609	1.81
`avg`		**58**	**0.09**	264	8.03	88	0.15	64	0.26	226	0.33	226	0.39
`med`		**50**	**0.07**	209	7.76	69	0.08	66	0.14	161	0.13	134	0.10

Table 4.3 reports the mean iteration numbers of PROSAC. Each row is a test pair. Columns are the results of different methods implemented in C++. The following methods are applied: the proposed 2-point algorithm, 3-point[9] [8], 5-point[10] [111], 6-point[11] [109], 7-point [43], 8-point[12] [47] techniques. It is reported that PROSAC makes the fewest iterations using the 2-point method except two cases. *Its mean and median iteration number and processing time are the lowest.* It can be seen that the 3-point method included into PROSAC is extremely slow compared with the others. The reason is that it is solved as a high-degree polynomial which is unstable. Even so, this test is slightly unfair since all algorithms but 2-point and 3-point methods exploit only the point locations which can be obtained faster than affine-covariant detectors providing their results.

[9] Own implementation is used.
[10] Available at `http://nghiaho.com/?p=1675`
[11] `http://users.cecs.anu.edu.au/~hartley/Software/5pt-6pt-Li-Hartley.zip`
[12] Normalized 7-point and 8-point methods are implemented in OpenCV.

Processing Time. The proposed algorithm consists of two main steps. First, the null space of a 6×9 matrix is calculated. Then the final solution is given as the pseudo-inverse of a matrix of size 10×9. Both steps have negligible time demand, therefore, the proposed algorithm is applicable even to online tasks. The generalization to n correspondences modifies only the first matrix to size $3n \times 9$ ($n \geq 2$). The mean processing time of 1000 runs of the 2-point version implemented in C++ is approx. 53×10^{-5} seconds. The time demand of the n-point version is around 49×10^{-3} seconds for $n = 4000$.

Augmenting a robust estimator, e.g. RANSAC [1], with the 2-point algorithm is beneficial since it yields significantly faster convergence. See Table 2.1 reporting the theoretical iteration number of RANSAC combined with different minimal methods. It is clear that the estimation exploiting two correspondences is advantageous to achieve real time performance even for high outlier ratio.

4.4.4 Application: Multi-motion Fitting

The clustering of correspondences to multiple rigid motions in two-views is usually solved by applying a multi-model fitting algorithm, e.g. PEARL [13] or Multi-X [22], combined with a minimal method as an engine estimating fundamental matrices. Recent approaches are based on a RANSAC-like initialization, therefore, their results highly depend on the applied minimal method, especially, on the size of the minimal sample – the probability of finding an accurate model increases if the model is estimable using less correspondences.

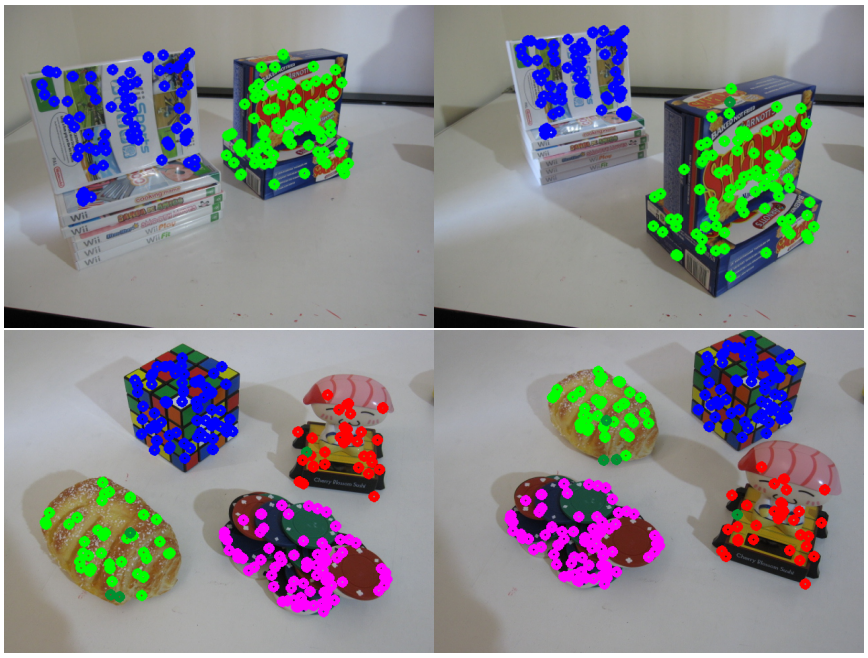


FIGURE 4.8: Example two-view multi-motion fitting on pairs Gamebiscuit and Cubebreadtoychips from the AdelaideRMF dataset. Color denotes motions.

Table 4.4 reports the results of Multi-X method fitting multiple rigid motions, i.e. fundamental matrices, simultaneously. Each row contains the results of a minimal method: the seven- (7PT) and eight-point (8PT) algorithms and the proposed one (2PT). The errors are the misclassification errors (ME), i.e. the ratio of misclassified

correspondences:

$$\text{ME} = \frac{\#\text{Misclassified Points}}{\#\text{Points}},$$

reported in percentage. Columns are the test pairs of the AdelaideRMF dataset[13] which consists of 18 image pairs of size 640×480 each containing point correspondences assigned to rigid motions manually. Since the proposed method requires affine correspondences, we applied AHessian-Affine to the image pairs detecting as many correspondences as we can. For all annotated correspondences, i.e. the point pairs provided in the dataset, we searched the closest match in the detected correspondence set, and replaced them with the matched ones. Note that this could introduce error into the annotation, however, these point pairs are used for all tests, including the proposed and competitor methods, thus the comparison remains fair. According to Table 4.4, it is clear that Multi-X leads to the most accurate clustering if it is combined with the two-point algorithm.

TABLE 4.4: Two-view multi-motion fitting on the AdelaideRMF dataset using Multi-X method augmented with different minimal methods (rows): the proposed two-point algorithm (2PT), the seven-point (7PT) and eight-point (8PT) methods. The reported errors are misclassification errors in percentage, i.e. the ratio of the misclassified correspondences. Test pairs: (1) biscuitbookbox, (2) breadcartoychips, (3) breadcubechips, (4) breadtoycar, (5) carchipscube, (6) cubebreadtoychips, (7) dinobooks, (8) toycubecar, (9) biscuit, (10) boardgame, (11) book, (12) breadcube, (13) breadtoy, (14) cube, (15) cubetoy, (16) game, (17) gamebiscuit, (18) cubechips.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	avg	med
2PT	5.0	**5.1**	2.2	**7.2**	**6.1**	4.9	**7.2**	**5.5**	**29.4**	8.2	2.7	5.2	11.5	**27.8**	**3.7**	**7.3**	3.7	**7.0**	**8.3**	**5.8**
7PT	**3.9**	5.5	**1.7**	7.8	**6.1**	**4.3**	11.4	6.0	30.3	8.6	2.7	2.5	11.8	29.8	5.2	7.7	3.0	8.1	8.7	6.1
8PT	4.6	8.4	2.2	**7.2**	7.3	6.1	10.6	6.5	32.1	8.6	2.7	3.3	**8.3**	28.5	4.8	8.6	**2.7**	9.2	9.0	7.3

4.4.5 Summary

Exploiting the two linear constraint which a local affine transformation provides, the essential matrix can efficiently be recovered using two affine correspondences. Even though the proposed solution assumes perspective camera model, it can straightforwardly be generalized to arbitrary one, e.g. omni-directional cameras.

It has been validated both on synthesized and real world data that the proposed 2-point algorithm works accurately and fast. Its processing time is far smaller than that of affine-covariant detectors. However, using GPU implementation of ASIFT [117] or that of other affine-covariant detectors could make the estimation real time. Applying the method to minimal samples led to camera motions superior to the state-of-the-art in half of the cases and was not worse in the remaining ones. Considering a non-minimal sample, e.g. consisting of five correspondences, for the robust estimation, the results were more accurate in all except two test cases. Augmenting a robust estimator, e.g. RANSAC, with it is beneficial and leads to significantly faster convergence. Moreover, it makes two-view multi-motion estimation more accurate as well.

[13] https://cs.adelaide.edu.au/~hwong/doku.php?id=data

4.5 A Minimal Solution for Two-view Focal-length Estimation using Two Affine Correspondences

The recovery of camera parameters and scene structure have been studied for over two decades since several applications, such as 3D vision from multiple views [88], are heavily dependent on the quality of the camera calibration. In particular, two major calibration types can be considered: aiming at the determination of the intrinsic and/or extrinsic parameters. The former ones include focal lengths, principal point, aspect ratio, and non-perspective distortion parameters, while the extrinsic parameters are the relative pose. Assuming two cameras with unknown extrinsic and *a priori* intrinsic parameters except a common focal length is called the *semi-calibrated case* [107]. It leads to the *unknown focal-length problem*: estimation of the relative motion and common focal length, simultaneously. The semi-calibrated case is realistic since (1) the aspect ratio is determined by the shape of the pixels on the sensors, it is usually 1:1; (2) the principal point is close to the center of the image, thus it is a reasonable approximation and (3) the distortion can be omitted if narrow field-of-view lenses are applied. Considering solely the locations of point pairs makes the problem solvable using at least six point pairs [106], [107], [118]. The objective of this paper is to *solve the problem exploiting only two local affine transformations*.

In general, 3D vision approaches [88] including state-of-the-art structure-from-motion pipelines [93], [119]–[121] apply a robust estimator, e.g. RANSAC [1], augmented with a minimal method, such as the five [111] or six-point [107] algorithm as an engine. Selecting a method exploiting as few point pairs as possible gains accuracy and drastically reduces the processing time. Benefiting from estimators which use less input data, the understanding of low-textured environment becomes significantly easier [9]. Moreover, minimal methods are advantageous from theoretical point-of-view leading to deeper understanding.

Local affine transformations represent the warp between the infinitely close vicinities of corresponding point pairs [5] and have been investigated for a decade. Their application field includes homography [32] and surface normal [5], [34] estimation; recovery of the epipoles [94]; triangulation of points in 3D [5]; camera pose estimation [92]; structure-from-motion [9]. In practice, local affinities can be accurately retrieved [28], [57] using e.g. affine-covariant feature detectors, such Affine-SIFT [2] and Hessian-Affine [3]. To the best of our knowledge, no paper has dealt with the unknown focal length problem using local affine transformations.

Forming a multivariate polynomial system and solving it by the *hidden-variable technique* [122], the proposed method is efficient and estimates the focal length and the relative motion using only two affinities. In order to eliminate invalid roots, a novel condition is introduced investigating the geometry of local affinities. To select the best candidate out of the remaining ones, we propose a root selection technique which is as accurate as the state-of-the-art for small noise and outperforms it for high-level noise.

4.5.1 Preliminaries

Semi-calibrated case is assumed in this paper as only the common focal-length f is considered to be unknown. Without loss of generality, the intrinsic camera matrix is $\mathbf{K} = \mathbf{K}^{\mathrm{T}} = \mathrm{diag}(f, f, 1)$, where f is the unknown focal-length. In order to replace $\mathbf{E}$ with $\mathbf{F}$ in Eq. 4.14 we define matrix $\mathbf{Q}$ as follows:

$$\mathbf{Q} = \mathrm{diag}\left(1, 1, \tau\right), \quad \tau = f^{-2}. \tag{4.19}$$

Due to the fact that $\mathbf{K}$ is non-singular, and trace($\mathbf{E}\mathbf{E}^{\mathrm{T}}$) identifies a scalar value, Eq. 4.14 can be simplified by multiplying with $\mathbf{K}^{-\mathrm{T}}$ and $\mathbf{K}^{-1}$ from the left and the right sides, respectively. Moreover, trace is invariant under cyclic permutations. As a consequence, Eq. 4.14 is written as [123], [124]

$$2\mathbf{F}\mathbf{Q}\mathbf{F}^{\mathrm{T}}\mathbf{Q}\mathbf{F} - \mathrm{tr}(\mathbf{F}\mathbf{Q}\mathbf{F}^{\mathrm{T}}\mathbf{Q})\mathbf{F} = 0. \tag{4.20}$$

This relationship will help us to recover the focal length and the fundamental matrix using two affine correspondences.

We use the *hidden variable technique* in the proposed method. It is a resultant technique in algebraic geometry for the elimination of variables from a multivariate polynomial system [122]. Suppose that m polynomial equations in n variables are given. In brief, one can assume an unknown variable as a parameter and rewrite the equation system as $\mathbf{C}(y_1)\mathbf{x} = 0$, where $\mathbf{C}$ is a coefficient matrix depending on the unknown y_1 (hidden variable) and vector $\mathbf{x}$ is the vector of $n-1$ unknowns. If the number of equations equals to that of the unknown monomials in $\mathbf{x}$, i.e. matrix $\mathbf{C}$ is square, the non-trivial solution can be carried out as $\det(\mathbf{C}(y_1)) = 0$. Solving the resultant equation for y_1 and back-substituting it, the whole system is solved.

4.5.2 Focal-length using Two Correspondences

This section aims the recovery of the unknown focal length and fundamental matrix using two affine correspondences.

Two-point Solver. Suppose that two affine correspondences ($\mathbf{p}_1^1$, $\mathbf{p}_2^1$, $\mathbf{A}^1$) and ($\mathbf{p}_1^2$, $\mathbf{p}_2^2$, $\mathbf{A}^2$) are given. Coefficient matrix

$$\mathbf{C}^i = \begin{bmatrix} x_2 + a_{11}x_1 & a_{11}y_1 & a_{11} & y_2 + a_{21}x_1 & a_{21}y_1 & a_{21} & 1 & 0 & 0 \\ a_{12}x_1 & x_2 + a_{12}y_1 & a_{12} & a_{22}x_1 & y_2 + a_{22}y_1 & a_{22} & 0 & 1 & 0 \\ x_1x_2 & y_1x_2 & x_2 & x_1y_2 & y_1y_2 & y_2 & x_1 & y_1 & 1 \end{bmatrix}$$

related to the ith ($i \in \{1, 2\}$) correspondence is formed as the combination of Eqs. 4.13, 4.5, 4.6 and satisfies formula $\mathbf{C}^i\mathbf{x} = 0$, where $\mathbf{x} = [f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad f_6 \quad f_7 \quad f_8 \quad f_9]^{\mathrm{T}}$ is the vector of unknown elements of the fundamental matrix. We denote the concatenated coefficient matrix of both correspondences as follows:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}^1 \\ \mathbf{C}^2 \end{bmatrix}. \tag{4.21}$$

It is of size 6×9, therefore, its left null space is three-dimensional. The solution is carried out as

$$\mathbf{x} = \alpha\mathbf{a} + \beta\mathbf{b} + \gamma\mathbf{c}, \tag{4.22}$$

where $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ are the singular vectors and α, β, γ are unknown non-zero scalar values.

Remember that only the common focal length is unknown from the intrinsic parameters, therefore, we are able to exploit the trace constraint. Eq. 4.20 yields ten cubic equations for four unknowns α, β, γ and τ, where $\tau = f^{-2}$ encapsulates the unknown focal length. We consider τ as the hidden variable and form coefficient matrix $\mathbf{C}(\tau)$ w.r.t. the other three ones – thus the rows of $\mathbf{C}(\tau)$ are univariate polynomials with variable τ. Even though α, β and γ are defined up to a common scale, we do not fix this scale in order to keep the homogeneity of the system. The monomials of

this polynomial system are as $\mathbf{y} = [\alpha^3 \;\; \alpha^2\beta \;\; \alpha^2\gamma \;\; \alpha\beta^2 \;\; \alpha\beta\gamma \;\; \alpha\gamma^2 \;\; \beta^3 \;\; \beta^2\gamma \;\; \beta\gamma^2 \;\; \gamma^3]^{\mathrm{T}}$. Table 4.5 demonstrates the coefficient matrix.

Since the scale of monomial vector $\mathbf{x}$ has not been fixed, the non-trivial solution of equation $\mathbf{C}(\tau)\mathbf{y} = 0$ is when the determinant vanishes as

$$\det(\mathbf{C}(\tau)) = 0. \tag{4.23}$$

Therefore, the hidden-variable resultant – a polynomial of the hidden variable – is $\det(\mathbf{C}(\tau))$. As the current problem is fairly similar to that of [107], we adopt the proposed algorithm. It is proved that $\det(\mathbf{C}(\tau))$ is actually a 15th degree polynomial and it obtains the candidate values for τ. Then the solution for α, β, γ and τ is given as $\mathbf{y} = \text{null}(\mathbf{C}(\tau))$. Finally, fundamental matrix $\mathbf{F}$ regarding to each obtained focal length can be directly estimated using Eq. 4.22.

TABLE 4.5: The coefficient matrix $\mathbf{C}(\tau)$ related to the ten polynomial equations of the trace constraint.

$\mathbf{C}(\tau)$	1 α^3	2 $\alpha^2\beta$	3 $\alpha^2\gamma$	4 $\alpha\beta^2$	5 $\alpha\beta\gamma$	6 $\alpha\gamma^2$	7 β^3	8 $\beta^2\gamma$	9 $\beta\gamma^2$	10 γ^3
1	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
.	.	.	.	.	.	.	.	.	.	.
10	c_{91}	c_{92}	c_{93}	c_{94}	c_{95}	c_{96}	c_{97}	c_{98}	c_{99}	c_{100}

4.5.3 Elimination and Selection of Roots

In this section, a novel technique is proposed to omit roots on the basis of the underlying geometry. Then we show a heuristics considering the properties of digital cameras to remove invalid focal lengths. In the end, we introduce a root selection algorithm.

Elimination of Invalid Focal Lengths. A solution is proposed here based on the underlying geometry to eliminate invalid focal lengths. Suppose that a point pair $(\mathbf{p}_1, \mathbf{p}_2)$, the related local affinity $\mathbf{A}$, the fundamental matrix $\mathbf{F}$, and an obtained focal length f are given. As the semi-calibrated case is assumed, $\mathbf{F}$ and f exactly determines the projection matrices $\mathbf{p}_1$ and $\mathbf{p}_2$ of both cameras [88]. Denote the 3D coordinates and the surface normal induced by point pair $(\mathbf{p}_1, \mathbf{p}_2)$, local affinity $\mathbf{A}$ and the projection matrices with $\mathbf{q} = [x \;\; y \;\; z]^{\mathrm{T}}$ and $\mathbf{n} = [n_x \;\; n_y \;\; n_z]^{\mathrm{T}}$, respectively. According to our experiences, linear triangulation [88] is a suitable and efficient choice to estimate $\mathbf{q}$. Surface normal $\mathbf{n}$ is estimated exploiting affinity $\mathbf{A}$ by the method proposed in [34].[14]

Without loss of generality, we assume that a point of a 3D surface cannot be observed from behind. As a consequence, the angle between vectors $\mathbf{c}_i - \mathbf{q}$ and $\mathbf{n}$ must be smaller than $90°$ for both cameras, where $\mathbf{c}_i$ is the position of the ith camera ($i \in \{1, 2\}$). This can be interpreted as follows: each camera selects a half unit-sphere around the observed point $\mathbf{q}$. Surface normal $\mathbf{n}$ must lie in the intersection of these half spheres. These half spheres are described by a rectangle in the spherical coordinate system as follows: $\text{rect}_i = \left[\theta_i - \frac{\pi}{2} \;\; \sigma_i - \frac{\pi}{4} \;\; \pi \;\; \frac{\pi}{2}\right]$, where θ_i, σ_i are the corresponding spherical coordinates and rect_i is of format $[\text{corner}_\theta \;\; \text{corner}_\sigma$

[14] http://web.eee.sztaki.hu/~dbarath/

width height]. The intersection area induced by the two cameras is as

$$\text{rect}_{\cap} = \bigcap_{i \in [1,2]} \text{rect}_i.$$

Point $\mathbf{q}$ is observable from both cameras *if and only if* surface normal $\mathbf{n}$, represented by spherical coordinates Θ and Σ, lies in the intersection area: $\begin{bmatrix}\Theta & \Sigma\end{bmatrix} \in \text{rect}_{\cap}$. A setup, induced by focal length f, not satisfying this criteria is an invalid one and can be omitted. Note that this constraint can be straightforwardly extended to the multi-view case making the intersection area more restrictive.

Physical Properties of Cameras. We introduce restrictions on the estimated roots considering the physical limits of the cameras. The focal length within camera matrix $\mathbf{K}$ is not equivalent to the focal length of the lenses, since it is the ratio of the optical focal length and the pixel size [88]. Particularly, the latter one is a few micrometers, while the optical focal length are within interval $[1 \dots 500]$ mm. Therefore, coarse lower and upper limits for a realistic camera are 100 and 500.000. Focal lengths out of this interval are automatically discarded. Note that these limits can be easily changed considering cameras with different properties.

Root Selection. To resolve the ambiguity of multiple roots and to minimize the effect of the noise, the classical way is to exploit multiple measurements eliminating the inconsistent ones. Since Eq. 4.23 is a high-degree polynomial it is sensitive to noise – small changes in the coordinates and affine elements cause significantly different coefficients.

RANSAC [1] is a successful technique for that problem, e.g. in the five-point relative-orientation one [111]. Recent methods, i.e. Kernel Voting, exploit the property that the roots form a peak around the real solution [107], [125], [126]. Kernel Voting maximizes a kernel density function like a maximum-likelihood-decision-maker. To our experiences, this technique works accurately if the noise in the coordinates does not exceed $1-2$ pixels on average. Over that, the roots may form several strongly supported peaks and it is not guaranteed that the true solution is found.

Thus we formulate the problem as a mode-seeking in a one dimensional domain: the real focal length appears as the most supported mode. Among several mode-seeking techniques [127] the most robust one is the Median-Shift [128] according to extensive experimentation. Median-Shift providing Tukey-medians [129] as modes does not generate new elements in the domain it is applied to. In particular, there is no significant difference in the results of Tukey- [129] and Weiszfeld-medians [130], however, the former one is slightly faster to compute. Finally, in order to overcome the discrete nature of Median-Shift – since it does not add new instances, only operates with the given ones –, we apply a gradient descent from the retrieved mode x_0 maximizing function

$$f(x) = \sum_{i=1}^{n} \frac{\kappa(x_i - x)}{h}, \tag{4.24}$$

where n is the number of focal lengths, κ is a kernel function – we chose Gaussian-kernel –, x_i is the ith focal length, and h is a bandwidth same as for the Median-Shift.

4.5.4 Experimental Results

Synthesized tests. For synthesized testing, two perspective cameras are generated by their projection matrices $\mathbf{p}_1$ and $\mathbf{p}_2$. The first camera is at position $[0 \quad 0 \quad 1]^T$ looking towards the origin, and the distance of the second one from the first is 0.15 far in a random direction. One hundred random planes passing over the origin are generated and each is sampled in a random location. The obtained 3D points are projected onto the cameras. Zero-mean Gaussian-noise is added to the point coordinates. The local affine transformations are calculated from the homographies induced by the tangent planes at the noisy point correspondences similarly to [32].

The competitor methods are: the six-point algorithm of Hongdong Li [107] and Hartley et al. [109]; the method of Perdoch et al. [6] approximating the affine correspondences by affine frames combined with both six-point algorithms. The implementations of these methods are available at `http://cmp.felk.cvut.cz/mini/`.

Figure 4.9 reports the kernel density function with Gaussian-kernel width 10 plotted as the function of the relative error (in percentage). Candidate focal lengths are estimated as follows:

1. Select two affine correspondences.
2. Apply the proposed 2-point method.
3. Repeat from Step 1.

The iteration limit is chosen to 100. The blue horizontal line reports the result of Median-Shift, the green one is that of Kernel Voting. The σ value of the zero-mean Gaussian-noise added to the point locations and affinities is (a) 0.01 pixels, (b) 0.1 pixels, (c) 1.0 pixels, (d) 3.0 pixels, (e) 3.0 pixels and there are 10% outliers, (f) 1.0 pixels with some errors in the aspect ratio: the true one is 1.00 but 0.95 is used. The real focal length is 600.

Confirming the validity of the proposed theory, the peak is over the ground truth focal length: 0% relative error. The proposed root selection is more robust than the Kernel Voting approach since the blue line is closer to the zero relative error even if the noise is high.

Fig. 4.10 reports the mean (top) and median (bottom) errors of the estimated fundamental matrices plotted as the function of the noise σ and compared with the results of Hartley et al.[109] and Perdoch et al.[131]. The error is the Frobenious norm of the estimated and ground truth fundamental matrices. 100 runs were performed on each noise level. It can be seen that the accuracy of the estimated fundamental matrices is similar to that of Hartley et al. [109].

Tests on Real Images. To test the proposed method on real world photos, 104 image pairs were downloaded[15] each containing the ground truth focal length in the EXIF data (see Fig. 4.12 for examples). Affine correspondences are detected by ASIFT [2] and the same procedure is applied as for the synthesized tests. Fig. 4.11(a) reports the histogram of the relative errors (in percentage) in the focal length estimates on all the 104 pairs. It can be seen that in most of the cases the obtained results are accurate, the relative error is close to zero. Fig. 4.11(b) shows the first image of an example pair and the point correspondences.

[15] `http://www2c.airnet.ne.jp/kawa/photo/ste-idxe.htm`

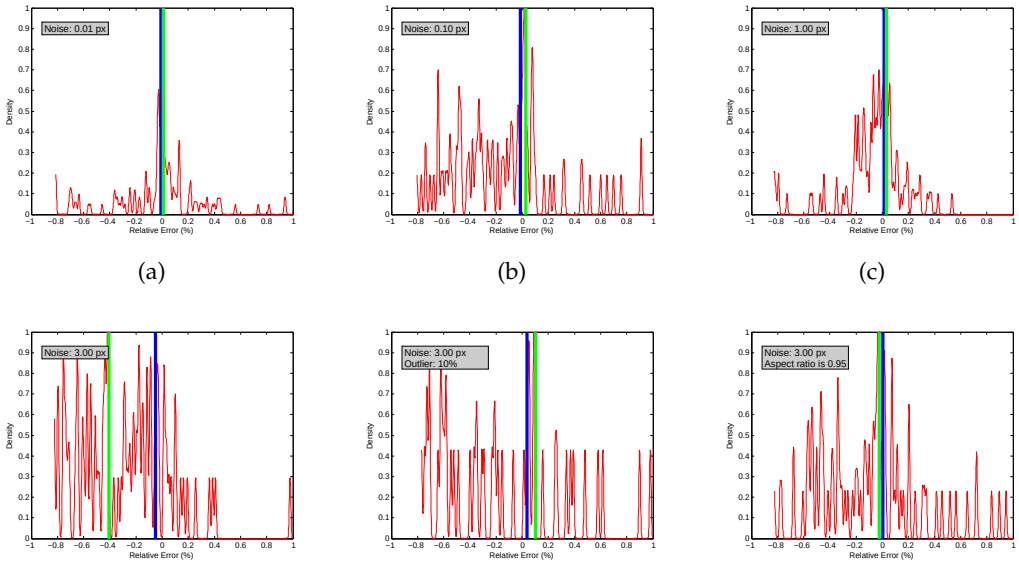


FIGURE 4.9: The kernel density function (vertical axis) with Gaussian-kernel width 10 plotted as the function of the relative error (%). Five planes are generated and each is sampled in 20 locations – points are projected onto the cameras and local affinities are calculated. The blue horizontal line is the result of Median-Shift, the green one is that of the Kernel Voting. The σ value of the zero-mean Gaussian-noise added to the point locations and affinities is (a) 0.01 pixels, (b) 0.1 pixels, (c) 1.0 pixels, (d) 3.0 pixels, (e) 3.0 pixels and there are 10% outliers, (f) 1.0 pixels with some errors in the aspect ratio: the true one is 1.00 but 0.95 is used. Ground truth focal length is 600. Best viewed in color.

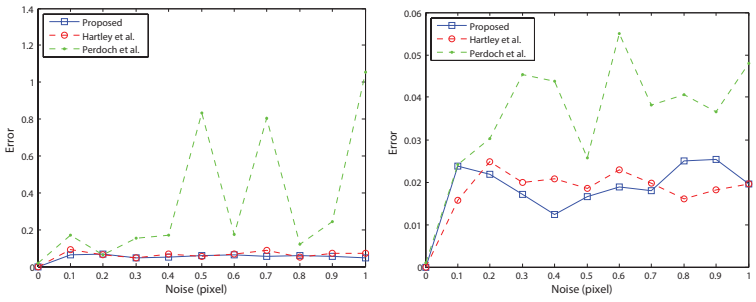
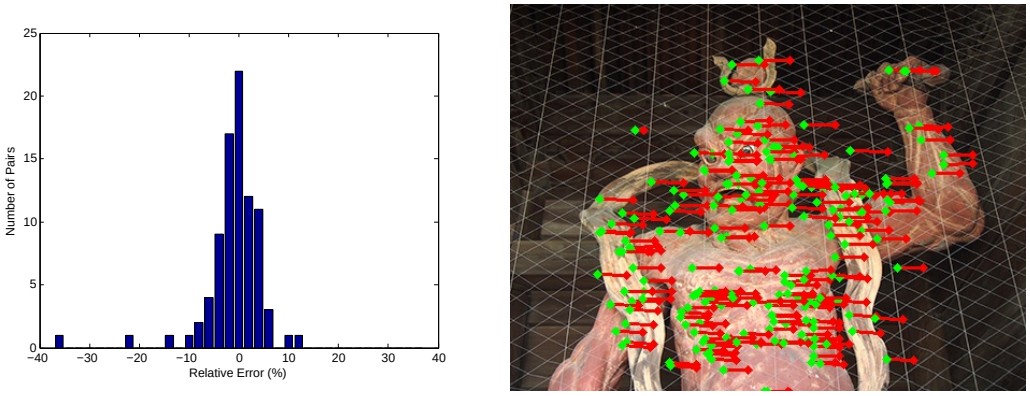


FIGURE 4.10: The mean (top) and median (bottom) Frobenious norms of the estimated and the ground truth fundamental matrices plotted as the function of the noise σ. 100 runs on each noise level were performed.



(a) (b)

FIGURE 4.11: (a) Histogram of focal length estimation on 104 image pairs. The horizontal axis is the number of the pairs plotted as the function of the relative error (%, vertical axis) in the focal length. (b) The first image of an example pair. Point coordinates on the first image (green dots), on the second one (red dots) and the point movements (red lines).

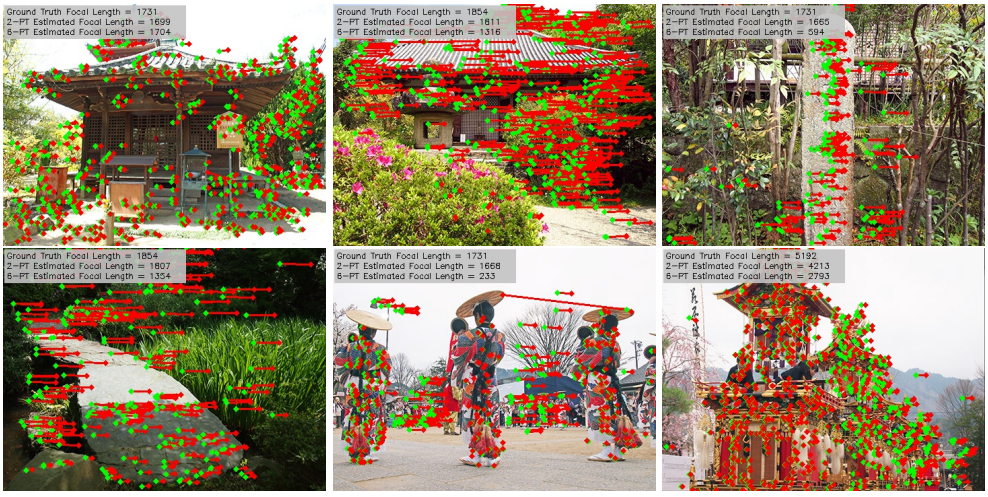


FIGURE 4.12: The first images of example pairs. Point coordinates on the first image (green dots), on the second one (red dots) and the point movements (red lines). The ground truth focal lengths, the results of the 6-point [109] and the proposed methods are written in gray rectangles.

4.5.5 Summary

A theory and an efficient method is proposed to estimate the unknown focal-length and the fundamental matrix using only two affine correspondences. The 2-point method is validated on both synthesized and real world data. Compared with the state-of-the-art methods, it obtained the most accurate focal lengths with fundamental matrices having similar quality as the recent algorithms. Combining the minimal solver with a robust statistics, e.g. RANSAC, allows significant reduction in computation. Particularly, its time demand is around a few milliseconds, thus it is much faster than affine-covariant detectors providing the input.

The proposed algorithm can also be applied in reconstruction or multi-view pipelines, e.g. that of Bujnak et al. [132], if at least two images of the same camera with fixed focal length are available.

Chapter 5

Robust Multi-Model Fitting

5.1 Introduction

In this chapter, we focus on robust model fitting, its theory and applications. Robust model fitting, even when we talk about single- or multi-model fitting, is a major component in most of the computer vision approaches, e.g. to calibrate cameras, for matching and retrieval, structure-from-motion, wide-baseline matching and, probably, for all tasks exploiting measured input data.

The structure of the chapter is as follows: (i) first, we propose a technique to reject outliers, i.e. incorrect point matches, from a set of point correspondences. The method assumes no a priori model in general, thus it is applicable even when standard approaches are not, e.g. to non-rigid scenes. (ii) The second technique we propose, called Graph-Cut RANSAC, combines RANSAC [1] with a local optimization step exploiting the spatial coherence of the input data to achieve state-of-the-art results. Benefiting from the new local optimization step, GC-RANSAC is superior to LO-RANSAC and its recent variants in term of geometric accuracy, and does not pay for this superiority with noticeable deterioration in processing time. (iii) We propose a multi-homography fitting algorithm, Multi-H, combining point-wise homography estimation and an energy-minimization-based relaxation. Multi-H significantly outperforms general model fitting approaches both in terms of accuracy and speed on publicly available datasets. Moreover, we propose a new dataset more challenging than the available ones to evaluate multi-homography fitting algorithms. (iv) In the end, we formalize multi-class multi-model fitting which has not been done before to our knowledge. Using this formulation we propose a method, called Multi-X, superior to the state-of-the-art, and techniques to set most of the parameters automatically on the basis of the input data.

5.2 Efficient Energy-based Topological Outlier Rejection

The most popular approaches to solve computer vision problems; including 3D reconstruction, camera calibration, image matching and retrieval; are usually based on point correspondences between two views. Even the matching in most of the recent multi-view systems, e.g. PMVS [133] or CMVS [134], relies on pair-wise correspondences registering each image pair separately as a first step. These correspondences, as they count as measured data, might be contaminated by noise and contain outliers which can corrupt the following estimation processes.

To deal with noise, optimal methods exist such as least-squares fitting. They have well-established mathematics, favorable statistical properties and have been used for decades. Removing the outliers from a set of noisy point correspondences is a more complex task requiring some a priori information about the observed scene in

most cases. Typically, this information is characterized by a model, e.g. fundamental matrix or homography. In this section, we address the problem of outlier filtering from a set of 2D point correspondences without necessarily assuming an underlying model.

One of the early methods for robust model fitting is the Hough-transform [135] which was first introduced as a method of detecting complex patterns of points in binary image data. It achieves this pattern detection by determining specific values of parameters which characterize these patterns. In 1981, Fischer and Bolles [1] introduced RANSAC, which is based on a very simple approach: hypothesis generation and validation. It requires a model to estimate and has probabilistic guaranties to find the best one minimizing a discrete, i.e. close or far, loss function. Even though (or because of) its simplicity, RANSAC still has very high impact in the computer vision community, it has thousands of citations and several modifications have been published year-by-year. Its novel variants, including PROSAC [116], MSAC [136] or LO-RANSAC [10], exploit RANSAC's modularity by changing its sampler, cost function or adding a local optimization step applied to the so-far-the-best model to achieve higher accuracy or faster convergence. The drawback of these approaches is the necessity of a single underlying model which cannot be guaranteed in all cases, e.g. fundamental matrix requires a rigid scene or the points must be coplanar for homography fitting.

Extending the "single model" approach to multiple ones, e.g. multiple rigid motions in two-views can be interpreted as multiple fundamental matrices, the range of describable scenes is widened. However, extending RANSAC to the multi-instance case has had limited success. Sequential RANSAC [137], [138] detects instances one after another in a greedy manner, removing their inliers. In this approach, data points are assigned to the first instance, typically the one with the largest support, for which they cannot be deemed outliers, rather than to the best instance. Multi-RANSAC [139] forms compound hypothesis about n instances. Besides requiring the number n of the instances to be known a priori, the approach increases the size of the minimum sample and thus the number of hypotheses that have to be validated. Recently, a popular group of methods [13], [15], [27] adopts a two step process: initialization by RANSAC-like instance generation followed by a point-to-instance assignment optimization by energy minimization using graph labeling techniques [140]. Another group of methods uses preference analysis, introduced by RHA [17], which is based on the distribution of residuals of individual data points with respect to the instances [83], [84], [141]. Even though the range of describable scenes is widened, these methods still need a model thus restricting the problem and introducing another uncertainty factor as the number of the models present in the scene.

A recently proposed technique, the DT-RANSAC [142], does not require an a priori model to solve the outlier rejection problem. First, it applies Delaunay triangulation [143] to the point sets in both the first and second images. Then it measures the distortion caused by each correspondence in the triangulations and all point pairs are labeled outliers for which the distortion exceeds an user-defined threshold. Its major advantage is that it does not need a model, however, does not optimize the topology thus ending up far from the optimum with not enough inliers kept in many cases.

In this section, we propose a method which exploits the topology of the point correspondences in order to avoid the need of a model describing the scene. However, to make it usable for wide range of problems, it can be easily combined with

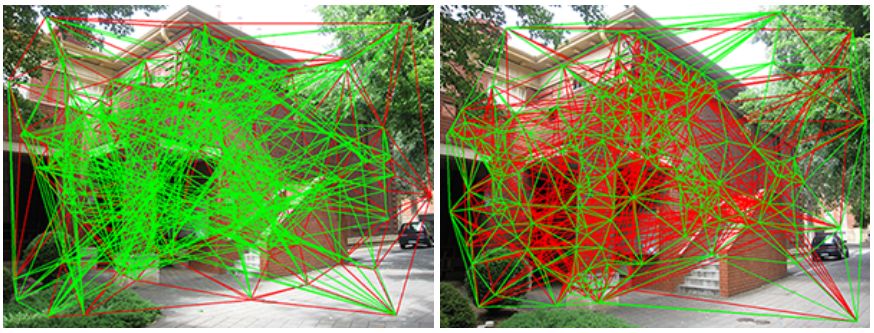


FIGURE 5.1: Structural difference of the neighborhoods in test pair `johnsona`. The neighborhood-graph is determined by Delaunay-triangulation. Red lines visualize conflict edges – edges which do not appear in both graphs. The Topological Distortion Penalty (TD-Penalty) is determined by the number of red edges.

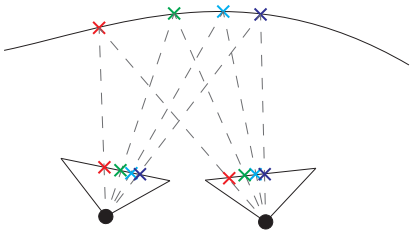
model fitting as it will be demonstrated later in the section. The used topological information is similar to that of DT-RANSAC, however, unlike them, we optimize the point topology in a global manner. The proposed method is based on energy minimization of a binary labeling, thus the solution is obtainable in polynomial time by a grab-cut-like algorithm [144]: alternation of graph-cut and re-fitting. For most of the tasks, the method is real time. It will be shown, that it outperforms RANSAC and its recent variants in term of the ratio of rejected outliers on publicly available datasets. Additionally, it is applicable to scenes which are degenerate for fundamental matrix estimation, e.g. non-rigid ones.

5.2.1 Energy-based Topological Outlier Filtering

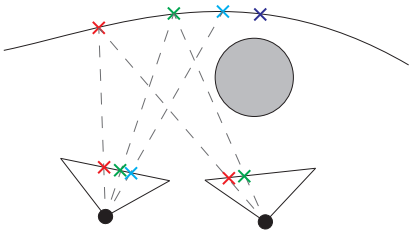
In this section, we propose a cost function, called Topological Distortion Penalty (TD-Penalty, see Fig. 5.1), to measure the distortion caused by the outliers in the neighborhood structures (e.g. determined by the K-Nearest-Neighbors algorithm) in the two images of 2D point correspondences. Then an energy minimization-based approach, which can optionally be combined with model estimation, is proposed to minimize the topological distortion.

Topological Distortion Penalty. Suppose that a set of point correspondences $\mathcal{P} = \{(\mathbf{p}_1^i, \mathbf{p}_2^i)\}_{i=1}^N$ is given in two images. Without assuming an underlying model, e.g. fundamental matrix or homography, a possible way to investigate the scene structure is to take the spatial coherence of the points into account, i.e. the neighboring information. To describe the spatial relations of the correspondences, a trivial preliminary step is to build a neighborhood-graph. Having two images leads to the question of the space on which the neighborhood-graph should be built: (a) the concatenated $\mathbb{R}^4$ coordinate space, or (b) separately observing the point structures in each image. In case (a), it is not trivial to define a cost which penalizes the structural difference between the images, however, case (b) offers a straightforward way to do so. Thus we chose case (b).

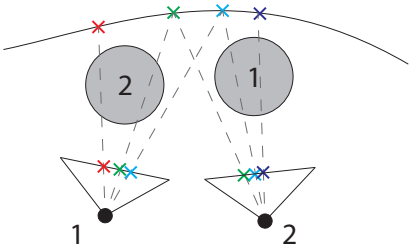
An important note that in this section the *similarity of the neighborhoods* of a point correspondence is interpreted as follows: given corresponding points $\mathbf{p}_1$ and $\mathbf{p}_2$. Get an arbitrary line $\mathbf{l}_1$ going through $\mathbf{p}_1$ and the corresponding line $\mathbf{l}_2$ intersecting the corresponding pixels (thus $\mathbf{p}_2$ as well) in the second image. Suppose that the lines go through the corresponding pixels keeping their orderings. The similarity of the neighborhoods means that every possible line correspondence keeps the ordering of



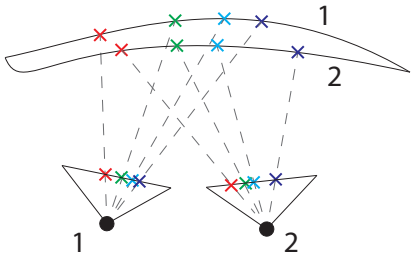
(a) Convex scene



(b) Concave scene



(c) Rigid motion. Numbers (1) and (2) denote the stages of the motion and the corresponding cameras.



(d) Two-cameras observing a non-rigid surface. Numbers (1) and (2) denote the stages of the motion and the corresponding cameras.

the corresponding pixels. To write this property formally, let us define a few things. Symbols L^2 and P^3 define the 2D line and 3D projective ($\mathbb{R}^3$ plus the homogeneous coordinate) spaces, respectively. Function $I : L^2 \times P^2 \rightarrow \{\text{True}, \text{False}\}$ is true if the input 2D line intersects the input homogeneous 2D point, otherwise false. Function $O : L^2 \times P^2 \rightarrow \mathbb{R}$ projects the input point to the input line and returns the distance of the projected point from a fixed one on the line (basically, returns parameter t regarding to the point from the parametric line formula $\mathbf{p} = \mathbf{p}_0 + t\mathbf{v}$, where $\mathbf{p}_0$ is the fixed point and $\mathbf{v}$ is the tangent direction). Set $\Theta_{\mathbf{l}_1,\mathbf{l}_2} = \{\mathbf{X} \mid \mathbf{X} \in P^3 \wedge I(\mathbf{l}_1, \mathbf{P}_1 X) \wedge I(\mathbf{l}_2, \mathbf{P}_2 X)\}$, where $\mathbf{P}_i$ is the projection matrix of the ith camera ($i \in \{1, 2\}$). Two lines correspond if $\forall \mathbf{X} \in P^3 : I(\mathbf{l}_1, \mathbf{P}_1 X) \Leftrightarrow I(\mathbf{l}_2, \mathbf{P}_2 X)$.

Definition 1 (Similarity of neighborhoods)**.** *Given corresponding points* $\mathbf{p}_1$ *and* $\mathbf{p}_2$ *in two images which are represented by* 3×4 *projection matrices* $\mathbf{P}_1$ *and* $\mathbf{P}_2$*. The neighborhoods of the points are similar if and only if* $\forall \mathbf{l}_1, \mathbf{l}_2 \in L^2$*, where* $\mathbf{l}_1$ *and* $\mathbf{l}_2$ *are corresponding lines,* $I(\mathbf{l}_1, \mathbf{p}_1)$ *and* $I(\mathbf{l}_2, \mathbf{p}_2)$*,* $\forall \mathbf{X}_1, \mathbf{X}_2 \in \Theta_{\mathbf{l}_1,\mathbf{l}_2} : O(\mathbf{l}_1, \mathbf{P}_1\mathbf{X}_1) < O(\mathbf{l}_1, \mathbf{P}_1\mathbf{X}_2) \Leftrightarrow O(\mathbf{l}_2, \mathbf{P}_2\mathbf{X}_1) < O(\mathbf{l}_2, \mathbf{P}_2\mathbf{X}_2))$

First, assume that a convex, rigid scene is observed (see Fig. 5.2(a)). In that case, the neighbors of each projected point must be the same in both images since perspective projection does not change the permutation of the points. To be more precise, choosing a direction and ordering the points along that direction in both images lead to the same point sequences. Thus if a point has a neighbor in the first image, the pair of that neighbor should be the neighbor of its corresponding pair on the second one. To be more precise, given two neighborhood-graphs $\mathbf{N}^k$ ($k \in \{1, 2\}$) in the two images. Sets $\mathcal{N}_i^1$ and $\mathcal{N}_i^2$ consist of the indices of the neighbors of point $\mathbf{p}_i^1$ in the first image and that of its corresponding pair $\mathbf{p}_i^2$ in the second one, respectively. Due to the proposed condition $\mathcal{N}_i^1$ is equal to $\mathcal{N}_i^2$ if and only if there are no outliers and the scene is convex.

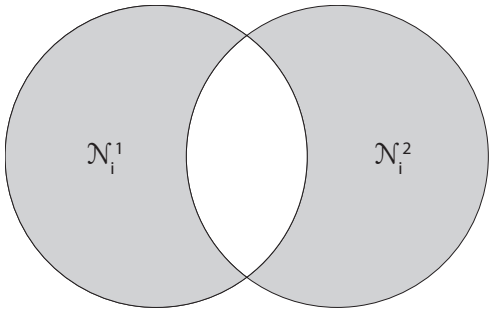


FIGURE 5.2: The symmetric difference of sets $\mathcal{N}_i^1$ and $\mathcal{N}_i^2$ is visualized by the gray regions.

Leaving the assumption of convexity (see Fig. 5.2(b)), the previously described condition holds no more globally. Even so, inside local convex-like regions it still holds, e.g. in Fig. 5.2(b), the projections of the green and red points are neighbors in both images. Non-rigidity affects this property in a fairly similar way, even we talk about the movement of rigid objects (see Fig. 5.2(c)) or some non-rigid materials (see Fig. 5.2(d)), e.g. endoscope images inside a human body.

Fig. 5.1 visualizes the neighborhoods (created using Delaunay Triangulation) on test pair `johnsona` from the `AdelaideRMF`[1] dataset. Red lines denote *conflict edges* which are edges not presenting in both neighborhood structures. Green ones are for edges which appear in both images, i.e. the two points of the correspondences are neighbors in both the first and second images.

To formalize this problem as an optimization, this observation have to be written as a cost function (called TD-Penalty) measuring the similarity of the point structures in the two images. Outliers cause differences in the neighboring sets of each correspondence. Cost

$$\mathrm{TD}(i) = |\mathcal{N}_i^1 \triangle \mathcal{N}_i^2| \tag{5.1}$$

for the ith correspondence should be minimized, where $\triangle$ is the standard symmetric difference operator of sets (see Fig. 5.2). For the sake of easier understanding, let us show this through an example. Suppose that the neighborhoods of the first point pair $\mathbf{p}_1^1$, $\mathbf{p}_1^2$ are $\mathcal{N}_1^1 = \{2, 4, 5\}$ and $\mathcal{N}_1^2 = \{2, 3, 5, 6\}$. Thus the vicinities of this correspondence in the neighborhood-graph of the first and second images consist of points $\mathbf{p}_2^1$, $\mathbf{p}_4^1$, $\mathbf{p}_5^1$ and $\mathbf{p}_2^2$, $\mathbf{p}_3^2$, $\mathbf{p}_5^2$, $\mathbf{p}_6^2$, respectively. Thus the implied cost is the cardinality of the symmetric difference of these two sets $\mathrm{TD}(1) = \left|\mathcal{N}_1^1 \triangle \mathcal{N}_1^2\right| = |\{3, 4, 6\}| = 3$.

Conflict edge. Using the previous formulation, we are able to define *conflict edges* in a mathematical way. Suppose that there is an edge e_{ij}^k connecting points p_i^k and p_j^k in the kth ($k \in \{1, 2\}$) neighborhood. Edge e_{ij}^k is not a conflict edge if $\forall k : \exists e_{ij}^k$, otherwise it is. This states that every edge which does not appear in both neighborhood structures called *conflict edge*.

Formulation as Energy Minimization. Having the topological distortion formalized, the problem is to find a correspondence set which minimizes the sum of the indicated topological energies as follows:

$$E_{\mathrm{sd}}(L) = \sum_{i=1}^{N} [L(i) = \mathrm{Inlier}] \cdot \mathrm{TD}(i), \tag{5.2}$$

[1] https://cs.adelaide.edu.au/~hwong/doku.php?id=data

where $L(i) : \mathbb{N} \rightarrow \{\text{Inlier}, \text{Outlier}\}$ is a labeling function assigning a label to the ith ($i \in [1, n]$) correspondence and $[.]$ is the Iverson bracket which is equal to one if the condition inside holds and zero otherwise.

Reflecting the fact that there might be several labelings solving the problem, e.g. the one which assigns all correspondences to the outlier class, a second energy has to be defined penalizing the point removal operation. This second term is as follows:

$$E_{\text{pr}}(l) = \sum_{i=1}^{n} [L(i) = \text{Outlier}] \cdot w_i, \tag{5.3}$$

where $w_i \in \mathbb{R}$ is a weight associated with all point pairs labeled outlier. Weight w_i introduces the way how the method can be combined with model estimation, e.g. fundamental matrix. It is calculated as follows:

$$w_i = \begin{cases} -d_i & \neg\text{Deg} \\ 1 & \text{otherwise}, \end{cases} \tag{5.4}$$

where parameter $\text{Deg} \in \{0, 1\}$ determines whether the scene is a degenerate one for fundamental matrix estimation or not. Distance d_i measures the fitness of the correspondence to fundamental matrix $\mathbf{F}$ as

$$d_i = e^{-\frac{S_i^2}{2\gamma}}, \quad S_i = \frac{((\mathbf{p}_i^2)^{\mathrm{T}} \mathbf{F} \mathbf{p}_i^1)^2}{\mathbf{F}p_{i,x}^1 + \mathbf{F}p_{i,y}^1 + \mathbf{F}^{\mathrm{T}}p_{i,x}^2 + \mathbf{F}^{\mathrm{T}}p_{i,y}^2}, \tag{5.5}$$

where $\gamma \in \mathbb{R}$ and $\mathbf{p}_i^k = [p_{i,x}^k \quad p_{i,y}^k \quad 1]^{\mathrm{T}}$ are a variance parameter of the Gaussian-kernel and the homogeneous form of the ith point in the kth image, respectively. S_i is the first-order geometric error, i.e. the Sampson-distance, of the ith correspondence w.r.t. the fundamental matrix. To our experience, this cost function leads to the highest outlier removal rate while keeping the most inliers. Degenerate cases can efficiently be determined using DEGENSAC [145] or manually set for trivially non-rigid scenes, e.g. endoscope images of a human body.

Combining the two proposed terms, the following energy is given

$$E(L) = \frac{1}{\lambda} E_{\text{sd}}(l) + \lambda E_{\text{pr}}(l), \tag{5.6}$$

where λ is a parameter balancing the terms. The optimization problem is formalized as $\arg_L \min E(L)$ and its solution is the labeling which minimizes the energy.

Minimization Strategy. As it is well-known in the field of energy minimization, the global optimum of a binary labeling problem can be found in polynomial time applying the s-t graph cut algorithm [140]. For the current problem, a single graph-cut does not find the optimum since changing a label of an individual correspondence changes (i) the energy originated from its neighbors and (ii) E_{pr} because the fundamental matrix depends on the current inlier set. Thus we propose an algorithm (see Alg. 4) similar to grab-cut which is, in brief, an alternated graph-cut and re-fitting.

The first step of Alg. 4 is the generation of an initial labeling for which we apply RANSAC with fundamental matrix estimation using a relatively high, 3 pixels, threshold. According to our experience, this suits for most of the tasks achieving a rough initial set up. The K-Nearest-Neighbors algorithm is applied to the determine a neighborhood structure in each image. If the scene is not degenerate, the first

step of the alternation is the estimation of $\mathbf{F}'$ w.r.t. the current inlier set. To avoid increasing energy, $\mathbf{F}'$ in the kth iteration is compared against $\mathbf{F}$ and $\mathbf{F}$ is updated if the energy (Eq. 5.3) is lower for $\mathbf{F}'$. This step is necessary since the re-estimation of the fundamental matrix could increase the energy, thus we use the re-estimated one only if it reduces the energy. In the next step, the problem graph G is built using the proposed unary terms (see Alg. 5). Function AddTerm1 is discussed by [146] in depth. Graph-cut is applied to G determining the optimal labeling L_i in the ith iteration. The convergence is achieved and the process terminates as soon as the energy does not change in two iterations.

Note that the fundamental matrix estimation step is not performed for deformable or non-rigid scenes. This is controlled by parameter Deg.

Algorithm 4 The main algorithm.

Input: $\mathcal{P} = (p_1^i, p_2^i)_{i=1}^n$ – data points; k – nearest neighbor
ϵ – threshold, Deg – degenerate scene;
λ – energy weight;
Output: L^* – labeling

$L_0 \leftarrow$ RANSAC($\mathcal{P}$, ϵ); ▷ Default $\epsilon = 3.0$
N^1, $N^2 \leftarrow$ KNN($(\mathbf{p}_1^i)_{i=1}^n$, k), KNN($(\mathbf{p}_2^i)_{i=1}^n$, k); ▷ Using FLANN algorithm
E_0, i, $\mathbf{F} \leftarrow \infty$, 0, 0;
repeat
  if $\neg Deg$ **then**
    $\mathbf{F}' \leftarrow$ FindFundamentalMat($\mathcal{P}$, L_i)
    if $i = 1$ || $E_{\text{pr}}(L_i, \mathbf{F}') < E_{\text{pr}}(L_i, \mathbf{F})$ **then**
      $\mathbf{F} \leftarrow \mathbf{F}'$
  $G \leftarrow$ ConstructGraph($\mathcal{P}$, $\mathbf{F}$, L_i, $\mathcal{N}^1$, $\mathcal{N}^2$, Deg, λ)
  $L_{i+1}, E_{i+1} \leftarrow$ GraphCut(G)
  Convergence, $i \leftarrow E_{i+1} = E_i$, $i + 1$
until Convergence
$L^* \leftarrow L_{i-1}$

Algorithm 5 Problem Graph Construction.

Input: $\mathcal{P} = (\mathbf{p}_1^i, \mathbf{p}_2^i)_{i=1}^n$ – data points; L – labeling
$\mathbf{F}$ – fundamental matrix, $\mathcal{N}^1$ – 1st neighborhood;
$\mathcal{N}^2$ – 2nd neighborhood, Deg – degenerate scene;
λ – energy weight;
Output: G – problem graph;

$G \leftarrow$ EmptyGraph(), 0;
for $j = 1..n$ **do**
  $c_0, c_1 \leftarrow 0, 0$
  if $L(j)$ = Inlier **then**
    $c_0 \leftarrow \frac{1}{\lambda}\mathcal{N}_j^1 \triangle \mathcal{N}_j^2$
  else
    $c_1 \leftarrow \lambda w_j$ ▷ Eq. 5.4
  $G \leftarrow$ AddTerm1(G, p, c_0, c_1).

TABLE 5.1: Applied parameter set up. The first row is the name of the parameter and the second one consists of the corresponding values.

λ	ϵ	k
1.73	3.00	4

Convergence. It can be easily seen that the main iteration in Alg. 4 does not increase the energy, thus it converges in finite steps. In each loop, graph-cut obtains the optimal labeling w.r.t. the current fundamental matrix. The energy cannot increase during this step due to the guaranties of graph-cut. Re-estimating the fundamental matrix could increase w_j, therefore, the new fundamental matrix is used only if the energy is reduced. Not having monotonically decreasing energy is guaranteed by the fact that there is a finite number of possible labelings. To our experience, the algorithm converges in maximum 6 iterations.

Implementation Details. We implemented the proposed method using C++ together with OpenCV. For Graph-Cut, the code from `http://vision.csd.uwo.ca/code/` is used. Fast Approximated Nearest Neighbors (FLANN) [147] algorithm is used to construct the neighborhood graphs.

Table 5.1 shows the used parameter set up. According to extensive evaluation on publicly available datasets, we found that $\lambda = 1.7$ leads to the solution with the highest outlier removal ratio. To determine an initial labeling, $\epsilon = 3.0$ pixel suits for all tasks, even for non-rigid scenes. The k value for the nearest neighbors algorithm is set to 4.

5.2.2 Experimental Results

In this section, we validate the proposed algorithm on various publicly available real world datasets presenting rigid or non-rigid scenes and compare it with the state-of-the-art techniques. Each property of a method, such as the outlier filtering ratio, is computed as the mean of the successful tests. A test is considered successful if at least one outlier is removed and one inlier remained. A filtering is considered perfect if all the outliers are removed.

Rigid Scenes. To test the method on rigid scenes, we used the `AdelaideRMF` dataset[2]. It consists of 18 image pairs with point correspondences each manually assigned to a plane using a label. Correspondences marked by label zero are the outliers, i.e. incorrect point matches. In Fig. 5.3, test pair `oldclassicswing` is shown. In each figure, the left, the middle, and the right columns consist of the neighborhood of the original point pairs, the neighborhood after the initialization and the final one, respectively. The original graphs contain many conflict edges which are mostly caused by the outliers. The results become much better after the label initialization step, however, they still contain several dissimilarities. The final results do not contain any conflict edges in both images.

The proposed algorithm is compared with the following methods: normalized RANSAC fundamental matrix (FM) estimation, normalized LMeDS FM estimation[3], normalized MLESAC FM estimation[4], and DT-RANSAC [142][5]. All algorithms are

[2] `http://cs.adelaide.edu.au/~hwong/doku.php?id=data`
[3] We applied the OpenCV 3 implementation of RANSAC and LMeDS.
[4] `https://code.google.com/p/itlab-computer-vision/`.
[5] Our implementation is used.

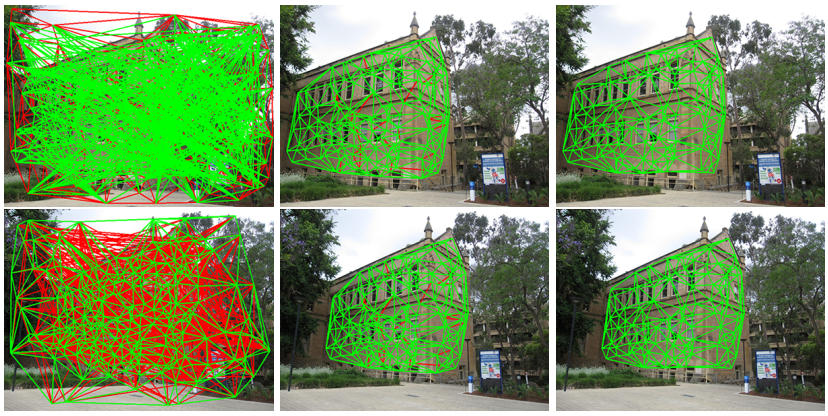


FIGURE 5.3: Image pair `oldclassicswing` (rows). The left, middle, and right columns visualize the original input data, the points after the initialization, and the resulting neighborhood structure, respectively.

implemented in C++. The used threshold and other parameters are tuned for each method separately to obtain the most accurate mean result on all test cases. The final fundamental matrix is estimated applying the normalized eight-point algorithm followed by the Levenberg-Marquardt [85] optimization exploiting the remaining inliers.

Table 5.2 reports the results on the `AdelaideRMF` dataset. For each method, the first, second and third columns report the percentage of removed outliers (O), kept inliers (I) and the error of the estimated fundamental matrix ($\mathcal{E}$), respectively. It can be seen that the proposed one achieves the highest outlier removal rate for all but two test scenes, its mean and median accuracy is also the highest. As a consequence, *the estimated fundamental matrix is closer to the ground truth*, i.e. the obtained mean and median errors are lower than that of the competitor methods. Even though the ratio of the kept inliers is the second lowest, the fundamental matrix was estimable in all test cases.

To get a comprehensive picture, Fig. 5.4 reports all the tested methods and all the aspects of comparison. The red column shows the outlier removal capability. The blue and green ones show the percentages of the kept inliers and frequencies of perfectly filtered cases, respectively. These values are computed on different subsets of the annotated correspondences. The first 20, 30, 40, 50, ..., n, points are processed separately by all methods. Thus the reported values are computed as the mean of 705 runs. The outlier removal and perfect filtering rates of the proposed technique are the highest. Even though the number of the kept inliers is the lowest, it is usually sufficient for model fitting, i.e. to estimate a fundamental matrix.

The left and right plots of Fig. 5.5 visualize the outlier removal ability and the percentage of the remaining inliers for each method as the function of the outlier level in the input, respectively. Fig. 5.5 shows the same trend as Fig. 5.4. LMedS is not shown over 50% of outliers since it is not applicable to that cases. However, its filtering accuracy is not as high as that of the other methods even below that. It can be seen that the proposed algorithm yields the highest outlier recognition rate.

Table 5.3 reports the mean and median processing times in milliseconds. Even though LMeDS and RANSAC are the fastest ones, the proposed method is still applicable in real time achieving around 2-3 times slower processing time than RANSAC.

TABLE 5.2: The outlier removal rate (O, in percentage), the ratio of the kept inliers (I, in percentage) and the error of the estimated fundamental matrices ($\mathcal{E}$) using the obtained labeling are reported. $\mathcal{E}$ is the Frobenious-norm of the difference matrix of the ground truth and estimated fundamental matrices. The methods are applied to the `AdelaideRMF` homography dataset consisting of 18 image pairs of rigid scenes (rows): (1) `hartley`, (2) `johnsona`, (3) `johnsonb`, (4) `ladysymon`, (5) `neem`, (6) `oldclassicswing`, (7) `sene`, (8) `physics`, (9) `bonython`, (10) `unionhouse`, (11) `elderhalla`, (12) `library`, (13) `napiera`, (14) `barrsmith`, (15) `elderhallb`, (16) `napierb`, (17) `unihouse`, (18) `bonhall`.

	Proposed			DT-RSC			LMedS			RANSAC			MLESAC		
	O	I	$\mathcal{E}$	O	I	$\mathcal{E}$	O	I	$\mathcal{E}$	O	I	$\mathcal{E}$	O	I	$\mathcal{E}$
(1)	98	23	**0.32**	**99**	63	3.01	46	**100**	2.42	**99**	44	2.41	**99**	4	1.91
(2)	**100**	38	0.01	92	91	**0.00**	**100**	**94**	**0.00**	**100**	73	0.01	**100**	22	0.02
(3)	**100**	57	**0.00**	95	**93**	0.01	**100**	88	**0.00**	**100**	64	**0.00**	99	36	0.09
(4)	**100**	37	0.02	95	85	0.31	99	**96**	**0.00**	**100**	79	0.01	97	52	0.03
(5)	98	57	0.04	**100**	81	**0.00**	94	**98**	**0.00**	99	49	0.03	89	74	0.11
(6)	**100**	47	**0.00**	96	87	0.04	98	**100**	**0.00**	**100**	61	0.01	99	58	0.08
(7)	**100**	34	**0.00**	98	80	0.04	96	**100**	**0.00**	**100**	45	0.01	97	15	0.14
(8)	98	34	**0.00**	**100**	76	**0.00**	**100**	**100**	**0.00**	**100**	47	**0.00**	88	3	0.01
(9)	**100**	50	0.03	99	71	0.03	3	**100**	0.02	98	67	**0.01**	**100**	6	–
(10)	**100**	37	0.07		–		16	**100**	**0.05**	99	38	0.06	95	1	0.26
(11)	**100**	39	0.01	98	86	0.01	35	**100**	**0.00**	99	37	0.01	**100**	1	–
(12)	**99**	34	**0.01**	97	76	0.19	77	**100**	**0.01**	**99**	49	0.01	98	3	0.41
(13)	**99**	23	0.05	95	55	0.07	52	**100**	**0.00**	**99**	22	0.06	98	6	0.04
(14)	99	28	**0.00**	99	55	0.02	27	**100**	0.01	100	43	**0.00**	**100**	8	0.08
(15)	**100**	51	**0.01**		–		94	**100**	0.05	99	47	0.02	99	10	0.07
(16)	**100**	29	**0.01**	95	82	0.27	96	**100**	0.03	**100**	42	0.12	**100**	12	0.03
(17)	**99**	70	0.02	82	82	0.07	81	**100**	**0.01**	93	64	**0.01**	85	89	0.04
(18)	**100**	71	**0.00**		–		77	**100**	0.02	74	59	0.00	88	63	0.01
avg	**99**	42	**0.03**	96	77	0.27	72	**98**	0.15	98	52	0.16	96	26	0.21
med	**100**	38	**0.01**	97	81	0.04	88	**100**	**0.01**	99	48	**0.01**	99	11	0.08

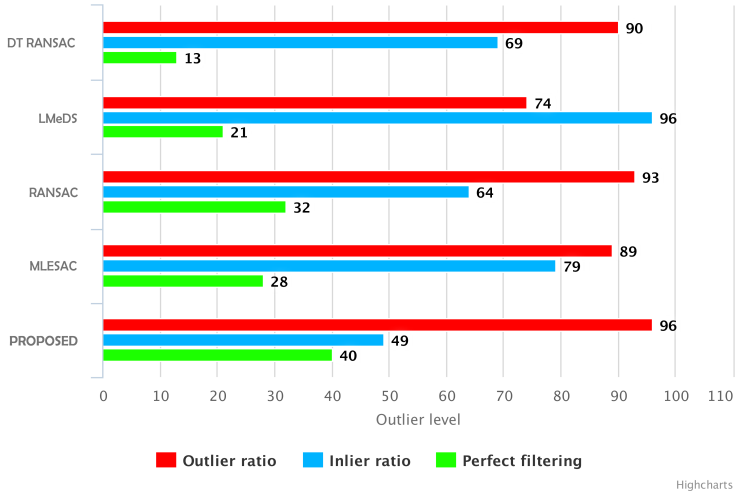


FIGURE 5.4: Performance comparison of robust methods. The red bar visualizes the percentage of the removed outliers for each method. The blue one shows the ratio of the kept inliers. The green line presents the percentage of the cases when all of the outliers are removed successfully.

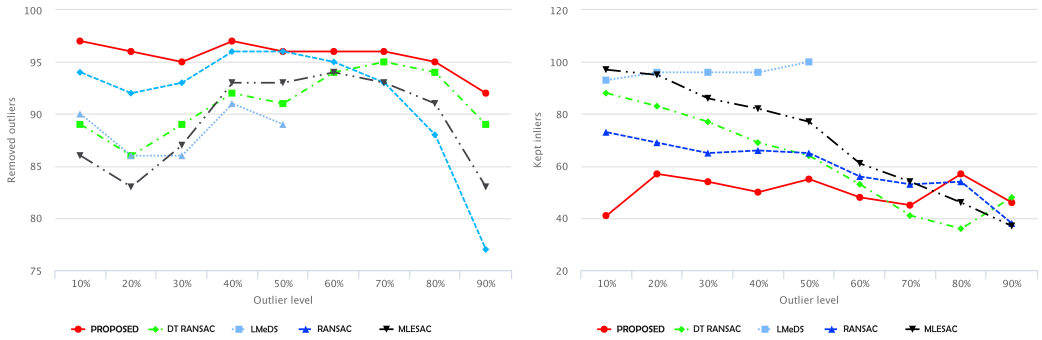


FIGURE 5.5: The outlier removal accuracy (left) and the percentage of kept inliers (right) is reported w.r.t. increasing outlier level.

TABLE 5.3: The processing time in milliseconds of each method applied to the `AdelaideRMF` dataset.

	Proposed	DT-RSC	LMedS	RANSAC	MLESAC
Mean Time (msec)	54	931	18	**16**	914
Median Time (msec)	28	820	**11**	18	543

Multiple Rigid Motions. In order to test the proposed method on image pairs for which a single fundamental matrix is not estimable, the `AdelaideRMF` motion dataset is exploited. Each image pair contains point correspondences manually assigned to a rigid motion or to the outlier class. Usually, two-view multiple rigid motion detection is solved by a multi-model fitting algorithm, e.g. PEARL [13], estimating multiple fundamental matrices simultaneously. Correspondences not belonging to any motions are considered outlier. For tasks, that only require the removal of outliers, applying the proposed method is beneficial. It is able to remove the outliers without assuming restrictive constraints, e.g. scene rigidity, thus generalizing and speeding up the process.

The proposed technique is compared with PEARL [13], T-Linkage [84], MFIPG [15], and RPA [141]. We choose these methods since their implementations are publicly available and they can be considered as state-of-the-art. Multi-model fitting methods are applied to each scene, then correspondences which are assigned to a motion considered inlier. All methods, including the proposed one, used a fixed parameter set up during the tests. The parameters for each method maximizing the mean outlier removal accuracy on all test cases are determined by extensive experimentation.

Table 5.4 reports the results of each method (rows) applied to each scene (columns). Even and odd columns show the outlier removal rates and the ratio of the kept inliers, respectively. It can be seen that the proposed method achieves the highest accuracy in both aspects. The reason, to our experience, is that multi-model fitting algorithms are very sensitive to the parameters, thus using a fixed set up leads to high reduction in accuracy. Fig. 5.5 presents that the processing time of the proposed method is the lowest – an order of magnitude faster than PEARL – and applicable in real time. Even so, this comparison with the multi-model fitting algorithms is slightly unfair since they aim at a more complex problem than outlier filtering. However, to the best of our knowledge, there is no other alternative to solve such problems.

TABLE 5.4: The accuracy of each method (columns) applied to scenes (rows) containing multiple rigid motions. Columns marked by O show the percentage of removed outliers and I shows the ratio of kept inliers. Incorrectly assigned correspondences are considered as outliers and point pairs belonging to a rigid motion are as inliers. See Table 5.5 for the processing times. Test pairs: (1) `book`, (2) `breadcartoychips`, (3) `breadcube`, (4) `breadcubechips`, (5) `breadtoy`, (6) `breadtoycar`, (7) `carchipscube`, (8) `cube`, (9) `cubebreadtoychips`, (10) `cubechips`, (11) `cubetoy`, (12) `dinobooks`, (13) `game`, (14) `gamebiscuit`, (15) `toycubecar`.

	Proposed		MFIPG		PEARL		T-Link		RPA	
	O (%)	I (%)	O (%)	I (%)	O (%)	I (%)	O (%)	I (%)	O (%)	I (%)
(1)	**98**	**79**	87	6	83	34	66	47	96	45
(2)	**100**	**27**	97	13	95	3	82	20	98	16
(3)	96	45	91	8	97	31	81	27	**99**	**69**
(4)	96	**30**	91	13	93	17	79	11	**99**	16
(5)	**100**	**76**	94	13	94	43	78	14	97	12
(6)	**96**	22	91	10	89	13	75	26	**96**	**43**
(7)	**98**	**44**	92	13	90	7	65	27	95	30
(8)	**100**	**69**	92	9	97	6	87	11	96	12
(9)	97	36	92	9	98	7	92	13	**98**	**40**
(10)	**100**	40	86	13	93	20	90	6	**100**	**52**
(11)	98	**44**	94	13	96	1	81	15	**99**	15
(12)	90	35	91	16	88	16	12	**97**	**97**	15
(13)	**100**	**49**	97	3	79	9	82	16	98	28
(14)	**100**	64	90	7	92	11	90	13	97	**65**
(15)	97	**43**	90	12	88	3	74	14	**98**	8
avg	**98**	**47**	92	11	91	15	76	24	**98**	31
med	**98**	**44**	91	12	93	11	81	15	**98**	28

TABLE 5.5: The mean and median processing times (in milliseconds) on multiple rigid motion detection applied to the `AdelaideRMF` motion dataset.

	Proposed	PEARL	MFIPG	T-Link	RPA
Mean Time (msec)	**19**	237	723	2 212	18 445
Median Time (msec)	**22**	155	696	2 111	18 655

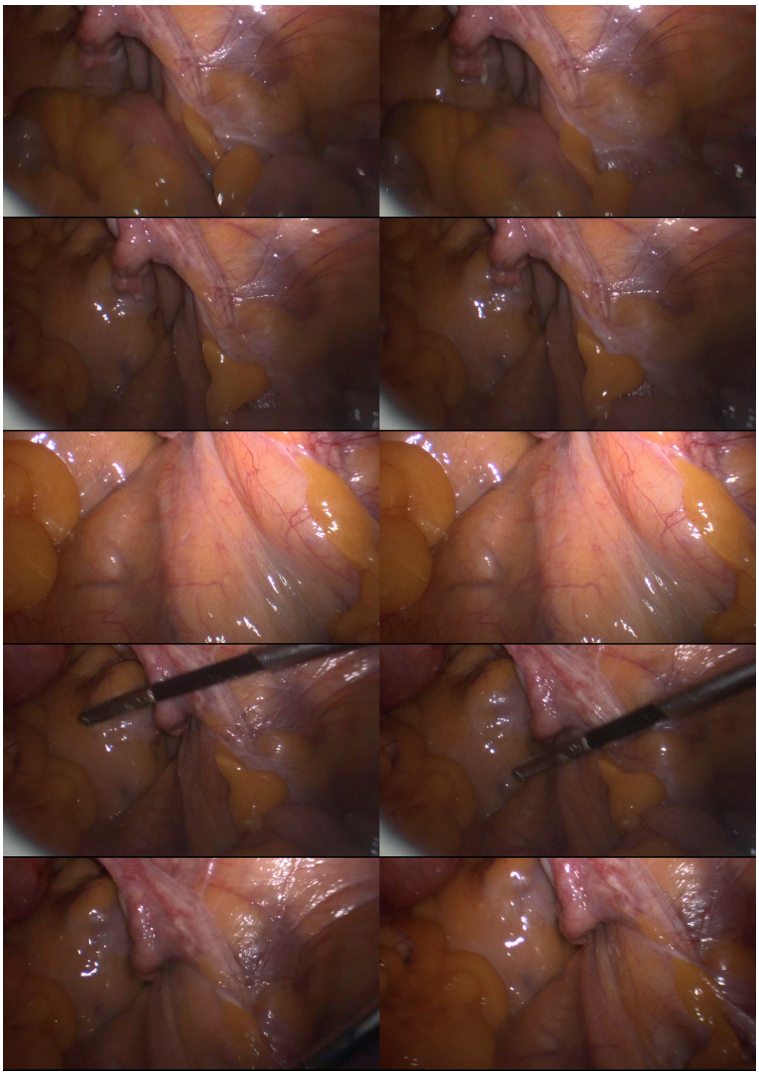


FIGURE 5.6: Pairs (rows) of endoscope images of the peritoneum. The scenes are non-rigid and the surfaces are shiny.

Non-Rigid Scenes. In this section, we show that the proposed method is applicable to scenes which are not describable by a finite combination of rigid motions. Such scenes are showed in Fig. 5.6 visualizing endoscope images of a peritorium. Each row is an image pair. The observed surface is extremely deformable, thus not interpretable by a mathematical model. Because of the shininess and nearly homogeneous regions, the feature matching process yields high outlier ratio even if the baseline is low. In order to obtain some information about the movements in the scene, e.g. camera or surface, the first step is the outlier removal.

Table 5.6 reports the achieved results on each image pair (rows) of Fig. 5.6. The second and third columns show the point and outlier numbers, respectively. The fourth column contains the outlier removal percentages and the last one is the ratio of the remaining inliers. It can be seen that the proposed method removes most of the outliers and keeps high percentage of inliers. Surprisingly, the ratio of the kept inliers is higher than for the other test cases. However, this is explainable by the quasi-convexity of these scenes since small surface deformities do not affect the neighborhood systems.

TABLE 5.6: Results of the proposed method applied to endoscope images of the peritoneum. The scenes are non-rigid and the surface is shiny. Each row corresponds to a row in Fig. 5.6. The second and third columns report the point and outlier number in each test, the last two columns show the percentages of the removed outliers and kept inliers.

Fig. 5.6	Point #	Outlier #	Removed Outliers	Kept Inliers
(1)	76	26	92%	88%
(2)	96	30	97%	90%
(3)	455	25	100%	93%
(4)	56	44	86%	83%
(5)	80	53	98%	74%

5.2.3 Summary

A novel approach is proposed to remove outliers from a set of correspondences. Generally, the proposed technique does not require any a priori models interpreting the scene, thus it is applicable without strict preconditions, e.g. rigidity. In cases, when the data are explainable by a model, e.g. fundamental matrix, the method can straightforwardly be specialized and achieves state-of-the-art outlier rejection ratio. Therefore, the fundamental matrices estimated exploiting the obtained correspondences are the most accurate ones. For scenes containing multiple rigid motions, the proposed approach is orders of magnitude faster than multi-model fitting algorithms and outperforms them in terms of rejection ratio as well as the number of kept inliers. It is applicable to image pairs showing fully deformable materials and obtains accurate results. Due to its real time performance, it will not be the bottleneck of e.g. structure-from-motion pipelines.

5.3 Graph-Cut RANSAC

The RANSAC (RANdom SAmple Consensus) algorithm proposed by Fischler and Bolles [1] in 1981 has become the most widely used robust estimator in computer vision. RANSAC and similar hypothesize-and-verify approaches have been successfully applied to many vision tasks, e.g. to short baseline stereo [148], [149], wide baseline stereo matching [46], [86], [150], motion segmentation [148], image mosaicing [151], detection of geometric primitives [152], multi-model fitting [139], or for initialization of multi-model fitting algorithms [13], [15].

In brief, the RANSAC approach repeatedly selects random subsets of the input data and fits a model to them, e.g. a line to two 2D points or a fundamental matrix to seven point correspondences. In the second step, the model support, i.e. the number of inliers, is obtained. The model with the highest support, polished e.g. by a least-squares fit on inliers, is returned.

In the last three decades, many modification of RANSAC have been proposed. For instance, NAPSAC [153], PROSAC [116] or EVSAC [154] modify the sampling strategy to increase the probability of selecting an all-inlier sample earlier. NAPSAC considers spatial coherence of the input data points, PROSAC exploits the ordering of the points by their inlier probability, EVSAC uses an estimate of confidence in each point. The model support computation step had also been discussed in several papers, e.g. MLESAC [155] and MSAC [43]. The model is estimated by a maximum-likelihood process, albeit under certain assumptions, with all its beneficial properties. In practice, MLESAC results are often superior to the inlier counting

of plain RANSAC and less sensitive to the used-defined threshold. The termination of RANSAC is controlled by a manually set confidence value q and the sampling stops when the probability of finding a model with higher support falls below q[6].

Observing that in practice RANSAC requires more samples than theory predicts, Chum et al. [10] identified a problem that not all all-inlier samples are "good", i.e. lead to a model accurate enough to distinguish all inliers, e.g. due to poor conditioning of the selected random all-inlier sample. Chum et al. [10] address the problem by introducing the locally optimized RANSAC (LO-RANSAC) that augments the original approach with a local optimization step applied to the *so-far-the-best* model. In the original paper [10], local optimization is implemented as an iterated least squares re-fitting with a shrinking inlier-outlier threshold inside an inner RANSAC applied only to the inliers of the current model. In the reported experiments, LO-RANSAC outperforms standard RANSAC in both accuracy and the required number of iterations. The number of LO runs is close to the logarithm of the number of verifications, and it does not create a significant overhead in the processing time in most of the cases tested.

However, it was shown by Lebeda et al. [156] that for models with high inlier counts the local optimization step becomes a computational bottleneck of the process due to the iterated least-squares model fitting. This is fixed by using a $7m$-sized subset of the inliers in each LO step, where m is the size of a minimum sample; the factor of 7 was set by exhaustive experimentation. The idea of local optimization has been included in state-of-the-art RANSAC approaches like USAC [11]. Nevertheless, the LO procedure remains ad hoc, complex and requires multiple parameters.

In this paper, we combine two strands of research to obtain a state-of-the-art RANSAC. So far, in the large body of RANSAC-related literature, the inlier-outlier decision has always been a function of the distance to the model, done individually for each data point. Yet both inliers and outliers are spatially coherent, a point near an outlier or inlier is more likely to be an outlier or inlier respectively. Spatial coherence, leading to the Potts-model [157], has been exploited in many vision problems, e.g. in segmentation [158], multi-model fitting [13], [15] or sampling [153]. It has probably been always considered computationally prohibitive to formulate model verification in RANSAC as a graph-cut problem. But when applied as the LO-step in [10] just on the *so-far-the-best* model, the number of graph-cut is only the logarithm of the number sampled and verified models, and can be achieved in real-time.

The novel method, called Graph Cut RANSAC (GC-RANSAC) is simply an LO-RANSAC with graph-cut as local optimization. GC-RANSAC is superior to LO-RANSAC in a number of aspects. First, as mentioned above, it is capable to model spatial coherence of inliers and outliers. Second, the LO step is conceptually simple, easy to implement, globally optimal and computationally efficient graph cut with only a few intuitive and learnable parameters unlike the ad hoc, iterative and complex LO steps [10]. Third, we show experimentally that GC-RANSAC outperforms LO-RANSAC and its recent variants in both accuracy and the required number of iterations on a wide range of publicly available datasets. On many problems, it is faster than the competitors in terms of wall-clock time. Finally, we were surprised to observe that GC-RANSAC terminates *before* the theoretically expected number of iterations. The reason is that the local optimization that takes spatial proximity into account is often capable of converging to a "good" model even when starting from a sample that is not all-inlier, i.e. it contains an outlier or outliers.

[6] This interpretation of q holds for the standard cost function only.

5.3.1 Local Optimization and Spatial Coherence

In this subsection, we formulate the inlier selection of RANSAC as an energy minimization considering point-to-point proximity. The proposed local optimization is seen as an iterative energy minimization of a binary labeling (outlier – 0 and inlier – 1). For the sake of simplicity, we start from the original RANSAC scheme and then formulate the maximum-likelihood estimation as an energy minimization. The term considering the spatial coherence will be included into the energy. Finally, we propose a technique to set the parameter balancing the energy terms automatically on the basis of the input.

Formulation as Energy Minimization. We assume that a point set $\mathcal{P} \subseteq \mathbb{R}^n$ ($n > 0$), a model represented by a parameter vector $\theta \in \mathbb{R}^m$ ($m > 0$) and a distance function $\phi : \mathcal{P} \times \mathbb{R}^m \to \mathbb{R}$ measuring the point-to-model assignment cost are given.

For the standard RANSAC scheme which applies a top-hat fitness function (1 – close, 0 – far), the implied unary energy is:

$$E_{\{0;1\}}(L) = \sum_{p \in \mathcal{P}} ||L_p||_{\{0;1\}},$$

where

$$||L_p||_{\{0;1\}} = \begin{cases} 0 & \text{if } (L_p = 1 \wedge \phi(p, \theta) < \epsilon) \vee \\ & \quad (L_p = 0 \wedge \phi(p, \theta) \geq \epsilon) \\ 1 & \text{otherwise.} \end{cases}$$

Parameter $L \in \{0, 1\}^{|\mathcal{P}|}$ is a labeling, ignored in standard RANSAC, $L_p \in L$ is the label of point $p \in \mathcal{P}$, $|\mathcal{P}|$ is the number of points, and ϵ is the inlier-outlier threshold. Using energy $E_{\{0,1\}}$ we get the same result as RANSAC since it does not penalize only two cases: (i) when p is labeled inlier and it is closer to the model than the threshold, or (ii) when p is labeled outlier and it is farer from the model than ϵ. This is exactly what RANSAC does.

Since the publication of RANSAC, several papers discussed, e.g. [156], replacing the $\{0, 1\}$ loss with a kernel function $K : \mathbb{R} \times \mathbb{R} \to [0, 1]$, e.g. the Gaussian-kernel. Such choice is close to maximum likelihood estimation as proposed in MLESAC [155]. This improves the accuracy and reduces the sensitivity on threshold ϵ. Unary term E_K exploiting this continuous loss is as follows:

$$E_K(L) = \sum_{p \in \mathcal{P}} ||L_p||_K,$$

where

$$||L_p||_K = \begin{cases} K(\phi(p, \theta), \epsilon) & \text{if } L_p = 1 \\ 1 - K(\phi(p, \theta), \epsilon) & \text{if } L_p = 0 \end{cases} \tag{5.7}$$

and

$$K(\delta, \epsilon) = e^{-\frac{\delta^2}{2\epsilon^2}}. \tag{5.8}$$

In GC-RANSAC, we use E_K as the unary energy term in the graph-cut based verification.

Spatial Coherence. Benefiting from a binary labeling energy minimization, we are able to include additional energy terms, i.e. consider spatial coherence of the points,

yet keep the problem *solvable efficiently and globally via the standard graph-cut algorithm*.

Considering point proximity is a well-known approach for sampling [153] or multi-model fitting [13], [15], [27]. To the best of our knowledge, there is no paper exploiting it in the local optimization step of methods like LO-RANSAC. Applying the Potts-model which penalizes all neighbors having different labels would be a justifiable choice to be the pair-wise energy. The problem arises when the data contains significantly more outliers, probably close to desired model, than inliers. In that case, penalizing differently labeled neighbors using the same penalty for all classes many times leads to the domination of outliers forcing all inliers to be labeled outlier. To overcome this problem, we modified the Potts-model to use different penalty for each neighboring point pair on the basis of their distances. The proposed pair-wise energy term is

$$E_S(L) = \sum_{(p,q)\in\mathcal{A}} \begin{cases} 1 & \text{if } L_p \neq L_q \\ \frac{1}{2}(K_p + K_q) & \text{if } L_p = L_q = 0 \\ 1 - \frac{1}{2}(K_p + K_q) & \text{if } L_p = L_q = 1 \end{cases}, \tag{5.9}$$

where $K_p = K(\phi(p,\theta),\epsilon)$, $K_q = K(\phi(q,\theta),\epsilon)$ and (p,q) is an edge of neighborhood graph $\mathcal{A}$ between points p and q. In E_S, if both points labeled outlier the penalty is $\frac{1}{2}(K_p + K_q)$ thus "rewarding" label 0 if the neighboring points are far from the model. The penalty of considering a point as inlier is $1 - \frac{1}{2}(K_p + K_q)$ which rewards the label if the points are close to the model.

The proposed overall energy measuring the fitness of points to a model and considering spatial coherence is $E(L) = E_K(L) + \lambda E_S(L)$, where λ is a parameter balancing the terms. The globally optimal labeling $L^* = \arg\min_L E(L)$ can easily be determined in polynomial time using graph-cut algorithm.

5.3.2 GC-RANSAC

In this subsection, we include the proposed energy minimization-based local optimization into RANSAC. Benefiting from this new approach, the LO step is getting simpler and cleaner than that of LO-RANSAC.

The main algorithm is shown in Alg. 6. The first step is the determination of neighborhood graph $\mathcal{A}$ for which we use a sphere with a predefined radius r – this is a parameter of the algorithm. Remark that for anisotropic spaces, a hyper-ellipsoid should be used instead of a (hyper-)sphere. In Alg. 6, function H is as follows [1]:

$$H(|L^*|,\mu) = \frac{\log(\mu)}{\log(1 - P_I)} \tag{5.10}$$

where $P_I = \binom{|L^*|}{m} / \binom{|P|}{m}$. It calculates the required iteration number of RANSAC on the basis of desired probability μ, the size of the required minimal point set m and the inlier number $|L^*|$ regarding to the current *so-far-the-best* model. Note that norm $|\cdot|$ applied to the labeling counts the inliers.

Every kth iteration draws a minimal sample using a sampling strategy, e.g. PROSAC [116], then computes the parameters θ_k of the implied model and its support

$$w_k = \sum_{p\in\mathcal{P}} \mathrm{K}(\phi(p,\theta_k),\epsilon) \tag{5.11}$$

w.r.t. the data points, where function K is a Gaussian-kernel as proposed in Eq. 5.8. If w_k is higher than that of the *so-far-the-best* model w^*, this model is considered the new *so-far-the-best*, all parameters are updated, i.e. the labeling, model parameters and support, and local optimization is applied if needed. Note that the application criterion of the local optimization step is discussed later.

The proposed local optimization is written in Alg. 7. The main iteration can be considered as a grab-cut-like [144] alternation consisting of two major steps: (i) graph-cut and (ii) model re-fitting. The construction of problem graph G using unary and pair-wise terms Eqs. 5.7, 5.9 is shown in Alg. 8. Functions AddTerm1 and AddTerm2 are discussed by [146] in depth. Graph-cut is applied to G determining the optimal labeling L which considers the spatial coherence of the points and their distances from the *so-far-the-best* model. Model parameters θ are computed using a $7m$-sized random subset of the inliers in L, thus speeding up the process, similarly to [156] does, where m is the size of a minimal sample, e.g. $m = 2$ for lines. Note that $7m$ is set by exhaustive experimentation in [156] and this value also suited for us. Finally, the support w of θ is computed and the *so-far-the-best* model is updated if the new one has higher support, otherwise the process terminates. After the main algorithm, a local optimization step is applied if it is not performed at least once during the algorithm, and the parameters of the obtained *so-far-the-best* model is re-estimated using the whole inlier set similarly to plain RANSAC does.

Remark: Adding to the local optimization step a RANSAC-like procedure selecting $7m$-size samples is straightforward. In our experiments, it had a high computational overhead without adding significantly to accuracy.

Algorithm 6 The GC-RANSAC Algorithm.

Input: $\mathcal{P}$ – data points; r – sphere radius, ϵ – threshold
ϵ_{conf} – LO application threshold, μ – confidence;
Output: θ - model parameters; L – labeling

$w^*, n_{LO} \leftarrow 0, 0$.
$\mathcal{A} \leftarrow$ Build neighborhood-graph using r.
for k = 1 $\rightarrow H(|L^*|, \mu)$ **do** ▷ Eq. 5.10
  $S_k \leftarrow$ Draw a minimal sample.
  $\theta_k \leftarrow$ Estimate a model using S_k.
  $w_k \leftarrow$ Compute the support of θ_k. ▷ Eq. 5.11
  if $w_k > w^*$ **then**
    $\theta^*, L^*, w^* \leftarrow \theta_k, L_k, w_k$
    if ApplyLocalOptimization(ϵ_{conf}) **then**
      $\theta_{LO}, L_{LO}, w_{LO} \leftarrow$ Local opt. ▷ Alg. 7
      $n_{LO} \leftarrow n_{LO} + 1$.
      if $w_{LO} > w^*$ **then**
        $\theta^*, L^*, w^* \leftarrow \theta_{LO}, L_{LO}, w_{LO}$
if $n_{LO} = 0$ **then**
  $\theta^*, L^*, w^* \leftarrow$ Local opt. ▷ Alg. 7
$\theta^* \leftarrow$ model fitting using L^*. ▷ E.g. least squares fitting

The criterion for applying the LO step was proposed to be: (i) the model is so-far-the-best and (ii) after a user-defined iteration limit, in [156]. However, in our experiments, this approach still spends significant time on optimizing models which

Algorithm 7 Local optimization.

Input: $\mathcal{P}$ – data points, L^* – labeling,
w^* – support, θ^* – model;
Output: L^*_{LO} – labeling, w^*_{LO} – support, θ^*_{LO} – model;

$w^*_{LO}, L^*_{LO}, \theta^*_{LO}, changed \leftarrow w^*, L^*, \theta^*, 1.$
while $changed$ **do**
  $G \leftarrow$ Build the problem graph. ▷ Alg. 8
  $L \leftarrow$ Apply graph-cut to G.
  $I_{7m} \leftarrow$ Select a $7m$-sized random inlier set.
  $\theta \leftarrow$ Fit a model using labeling I_{7m}.
  $w \leftarrow$ Compute the support of θ.
  $changed \leftarrow 0$.
  if $w > w^*_{LO}$ **then**
    $\theta^*_{LO}, L^*_{LO}, w^*_{LO}, changed \leftarrow \theta, L, w, 1.$

Algorithm 8 Problem Graph Construction.

Input: $\mathcal{P}$ – data points, $\mathcal{A}$ – neighborhood-graph
θ – model parameters, θ^* – model;
Output: G – problem graph;

$G \leftarrow$ EmptyGraph().
for $p \in \mathcal{P}$ **do**
  $c_0, c_1 \leftarrow \mathrm{K}(\phi(p, \theta), 1 - \mathrm{K}(\phi(p, \theta), \epsilon)$
  $G \leftarrow$ AddTerm1(G, p, c_0, c_1).
for $(p, q) \in \mathcal{A}$ **do**
  $c_{01}, c_{10} \leftarrow 1, 1.$
  $c_{00} \leftarrow 0.5(\mathrm{K}(\phi(q, \theta) + \mathrm{K}(\phi(p, \theta)).$
  $c_{11} \leftarrow 1 - 0.5(\mathrm{K}(\phi(q, \theta) + \mathrm{K}(\phi(p, \theta)).$
  $G \leftarrow$ AddTerm2(G, p, q, c_{00}, c_{01}, c_{10}, c_{11}).

TABLE 5.7: Setting for the tests. Outlier threshold (ϵ), radius used for proximity computation (r), weight of the pair-wise term (λ), and the threshold of the confidence change (ϵ_{conf}).

ϵ	r	λ	ϵ_{conf}
0.31	20 px	0.10	10

are not promising enough. We introduce a simple heuristics for replacing the iteration limit with a data driven strategy which allows to apply LO only a few times without deterioration in accuracy.

As it is well-known for RANSAC, the required iteration number k, w.r.t. the inlier ratio η, sample size m and confidence μ, is calculated as $k = \log(1-\mu)/\log(1-\eta^m)$. Re-arranging this formula to μ leads to equation $\mu = 1 - 10^{k \log(1-\eta^m)}$ which determines the confidence of finding the desired model in the kth iteration if the inlier ratio is η.

Suppose that the algorithm finds a new so-far-the-best model with inlier ratio η_2 in the k_2th iteration, whilst the previous best model was found in the k_1th iteration with inlier ratio η_1 ($k_2 > k_1$, $\eta_2 > \eta_1$). The ratio of the confidences μ_{12} in those two models is calculated as follows:

$$\mu_{12} = \frac{\mu_2}{\mu_1} = \frac{1 - 10^{k_2 \log(1-\eta_2^m)}}{1 - 10^{k_1 \log(1-\eta_1^m)}}. \tag{5.12}$$

In experiments, we observed that a model that leads to termination if optimized often shows a significant increase in the confidence. Replacing the parameter blocking LO in the first k iterations, we adopt a criterion $q_{12} > \epsilon_{\text{conf}}$, where ϵ_{conf} is a user-defined parameter determining a significant increase.

5.3.3 Experimental Results

In this section, GC-RANSAC is validated both on synthesized and publicly available real world data and compared with plain RANSAC [1], LO-RANSAC [10], LO$^+$-RANSAC, LO'-RANSAC [156], and EP-RANSAC [159]. The parameter setting is reported in Table 5.7. For EP-RANSAC[7], we tuned the threshold parameter to achieve the lowest mean error and the other parameters were set to the values reported by the authors. Note that the comparison of the processing time with this method is affected by the availability of a Matlab implementation only. All methods apply PROSAC [116] sampling and encapsulates the *point-to-model* distance, e.g. re-projection error for homographies, with a Gaussian-kernel using $\epsilon = 0.31$, which is set by an exhaustive search. EP-RANSAC uses inlier maximization strategy since its cost function cannot be replaced straightforwardly. The radius of the sphere to determine neighboring points is 20 pixels and it is applied to the concatenated $4D$ coordinates of the correspondences. Parameter λ for GC-RANSAC was set to 0.1 and $\epsilon_{\text{conf}} = 10$.

Synthetic Tests on 2D Lines. To compare GC-RANSAC with the state-of-the-art in a fully controlled environment, we chose two simple tests: detection of a 2D straight or dashed line. For each trial, a 600×600 window and a random line was generated in its implicit form, sampled at 100 locations and zero-mean Gaussian-noise with σ

[7]The Matlab source is available at http://cs.adelaide.edu.au/~huu/publication/exact_penalty/

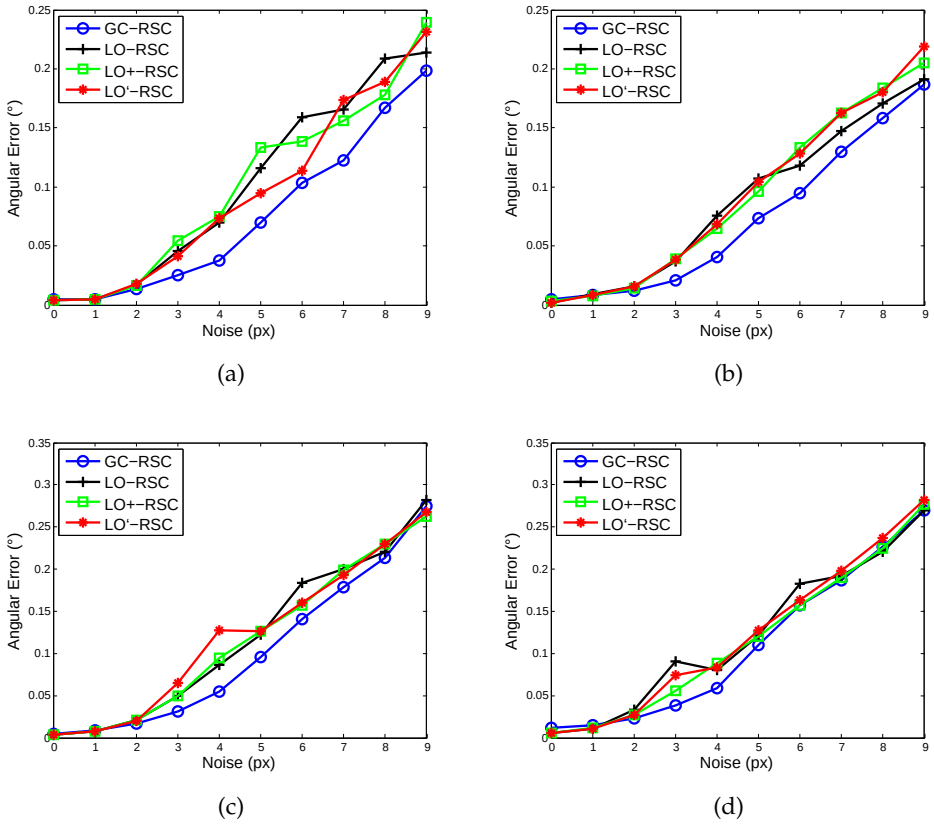


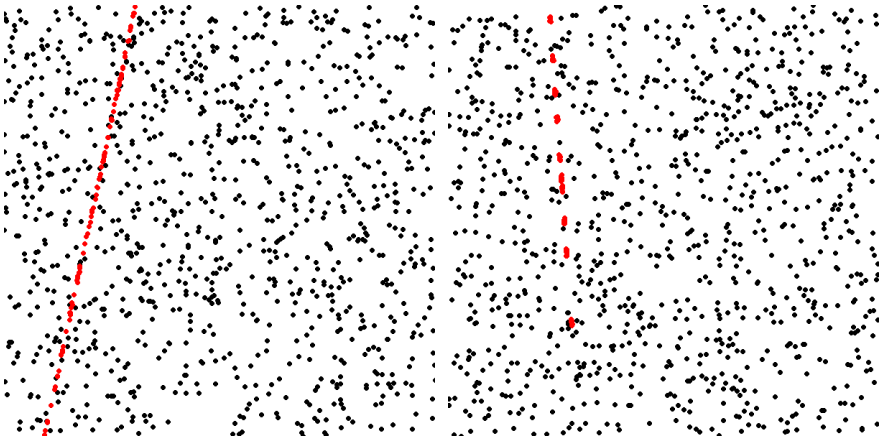
FIGURE 5.7: The mean angular error (in degrees) of the obtained 2D lines plotted as the function of noise σ (in pixels). On each noise σ, 1000 runs were performed. The line type and outlier number is (a) straight line, 100%, (b) straight line, 500% (c) dashed line, 100% and (c) dashed line, 500%.

standard deviation was added to the coordinates. For a straight line, the points were generated using uniform distribution (see Fig. 5.8(a)). For a dashed line, 10 knots were put randomly into the window, then the line is sampled at 10 locations with uniform distribution around each knot, at most 10 pixels far (see Fig. 5.8(b)). Finally, k outliers were added to the scene. 1000 tests were performed on every noise level.

Fig. 5.7 shows the mean angular error (in degrees) plotted as the function of the noise σ. The first and second rows report the results of the straight and dashed line cases. For the two columns, 100 and 500 outliers were added, respectively. According to Fig. 5.7, *GC-RANSAC obtains more accurate lines than the competitor algorithms*.

TABLE 5.8: Percentage of "not-all-inlier" minimal samples leading to the correct solution during line (**L**) and fundamental matrix (**F**) fitting. For lines, the average over 1000 runs on three different outlier percentage (100%, 500%, 1000%) and noise levels $0.0 - 9.0$ px is reported, thus 15000 runs were performed. For **F**, the mean of 1000 runs on the `AdelaideRMF` dataset is shown.

	LO	LO$^+$	LO'	**GC**
L	6%	5%	4%	**15%**
F	29%	30%	24%	**32%**



(a) (b)

FIGURE 5.8: An example input for (a) straight and (b) dashed lines. The 1000 black points are outliers, the 100 red ones are inliers. *Best viewed in color.*

Estimation of Fundamental Matrix. To evaluate the performance of GC-RANSAC on fundamental matrix estimation, we used `kusvod2` (24 pairs)[8], `Multi-H`[9] (5 pairs), and `AdelaideRMF`[10] (19 pairs) datasets (see Fig. 5.9 for examples). `Kusvod2` consists of 24 image pairs of different sizes with point correspondences and fundamental matrices estimated using manually selected inliers. `AdelaideRMF` and `Multi-H` consist a total of 24 image pairs with point correspondences, each assigned manually to a homography (or the outlier class). For them, all points which are assigned to a homography were considered as inliers and others as outliers. On total, the proposed method was tested on 48 image pairs from three publicly available datasets for fundamental matrix estimation. All methods applied the 7-point method [43] to estimate **F**, thus drawing minimal sets of size seven in each RANSAC iteration. For the model re-estimation from a non-minimal sample in the LO step, the normalized 8-point algorithm [47] is used. Note that all fundamental matrices were discarded for which the *oriented* epipolar constraint [160] did not hold.

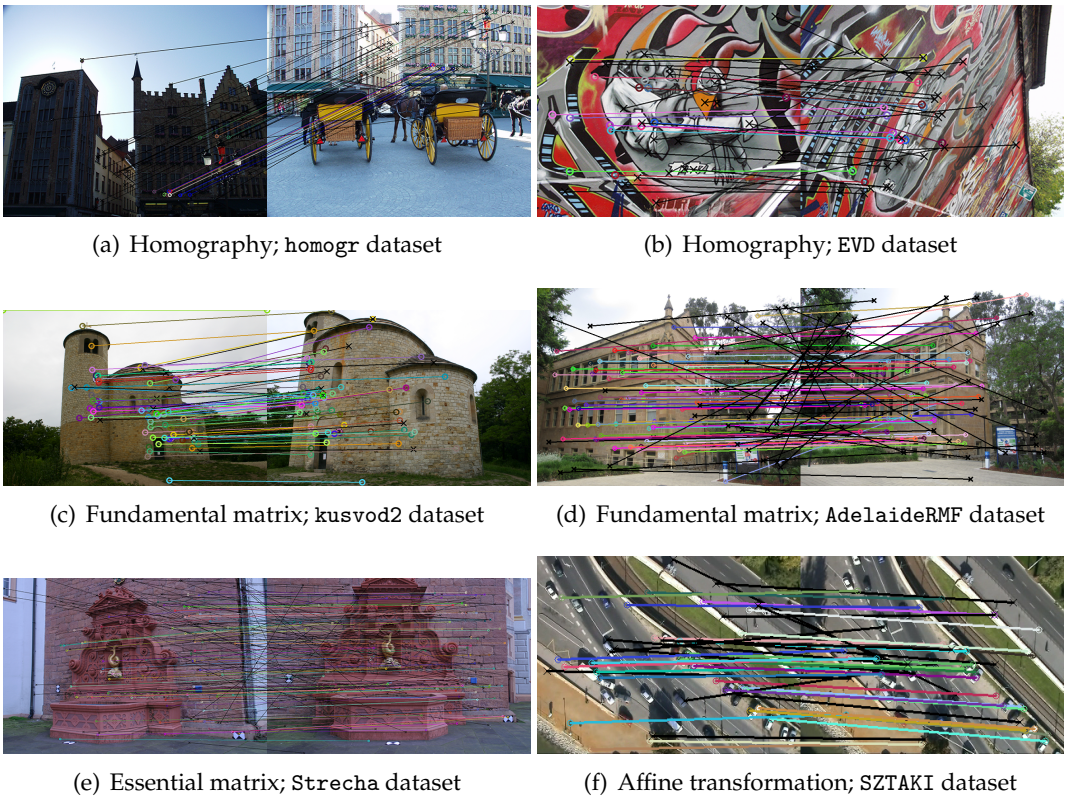
The first three blocks of Table 5.9, each consisting of four rows, report the quality of the epipolar geometry estimation on each dataset as the average of 1000 runs on every image pair. The first two columns show the name of the tests and the investigated properties: **(1)** LO: the number of applied local optimization steps (graph-cut steps are shown in brackets). **(2)** $\mathcal{E}$ is the geometric error (in pixels) of the obtained model w.r.t. the manually annotated inliers. For fundamental matrices and homographies, it is defined as the average Sampson distance and re-projection error, respectively. For essential matrices, it is the mean Sampson distance of the implied fundamental matrix and the correspondences. **(3)** $\mathcal{T}$ is the mean processing time in milliseconds. **(4)** $\mathcal{S}$ is the average number of minimal samples have to be drawn until convergence, basically, the number of RANSAC iterations.

It can be clearly seen that for fundamental matrix estimation GC-RANSAC *always obtains the most accurate model* using less samples than the competitive methods.

[8] `http://cmp.felk.cvut.cz/data/geometry2view/`

[9] `http://web.eee.sztaki.hu/~dbarath/`

[10] `cs.adelaide.edu.au/~hwong/doku.php?id=data`



(a) Homography; `homogr` dataset (b) Homography; `EVD` dataset

(c) Fundamental matrix; `kusvod2` dataset (d) Fundamental matrix; `AdelaideRMF` dataset

(e) Essential matrix; `Strecha` dataset (f) Affine transformation; `SZTAKI` dataset

FIGURE 5.9: Results of GC-RANSAC on example pairs from each dataset and problem. Correspondences are drawn by lines and circles, outliers by black lines and crosses, every third correspondence is drawn.

Estimation of Homography. In order to test homography estimation we downloaded `homogr`[11] (16 pairs) and `EVD`[12] (15 pairs) datasets (see Fig. 5.9 for examples). Each consists of image pairs of different sizes from 329×278 up to 1712×1712 with point correspondences and manually selected inliers – correctly matched point pairs. `Homogr` dataset consists of short baseline stereo pairs, whilst the pairs of `EVD` undergo an extreme view change, i.e. wide baseline. All methods apply the normalized four-point algorithm [43] for homography estimation both in the model generation and local optimization steps. Therefore, each minimal sample consists of four correspondences.

The 4th and 5th blocks of Fig. 5.9 show the mean results computed using all the image pairs of each dataset. It can be seen that GC-RANSAC obtains the most accurate models for all but one, i.e. `EVD` dataset with time limit, test cases.

Estimation of Essential Matrix. To estimate essential matrices, we used the `strecha` dataset [161] consisting of image sequences of buildings. All image sizes are 3072×2048. The ground truth projection matrices are provided. The methods were applied to all possible image pairs in each sequence. The SIFT detector [4] was used to obtain correspondences. For each image pair, a reference point set with ground truth inliers was obtained by calculating the fundamental matrix from the projection matrices [43]. Correspondences were considered as inliers if the symmetric epipolar

[11] http://cmp.felk.cvut.cz/data/geometry2view/
[12] http://cmp.felk.cvut.cz/wbs/

distance was smaller than 1.0 pixel. All image pairs with less than 20 inliers found were discarded. In total, 467 image pairs were used in the evaluation.

The results are reported in the 6th block of Table 5.9. The reason of the high processing time is that the mean inlier ratio is relatively low (27%) and there are many correspondences, 2323, on average. GC-RANSAC obtains the most accurate essential matrices both in the wall-clock time limited and solution confidence above 95% experiments. A significant drop can be seen in accuracy for all methods if a time limit is given.

Estimation of Affine Transformation. The `SZTAKI Earth Observation` dataset[13] [162] (83 image pairs of size 320×240) was used to test estimation of affine transformations. The dataset contains images of busy road scenes taken from a balloon. Due to the altitude of the balloon, the image pair relation is well approximate by an affine transformation. Point correspondences were detected by the SIFT detector. For ground truth, 20 inliers were selected manually. Point pairs with the distance from the ground truth affine transformation lower than 1.0 pixel were defined as inliers.

The estimation results are shown in the 7th block of Table 5.9. The reported geometric error is $|\mathbf{A}\mathbf{p}_1 - \mathbf{p}_2|$, where $\mathbf{A}$ is the estimated affine transformation and $\mathbf{p}_k$ is the point in the kth image ($k \in \{1, 2\}$). It can be seen that the methods obtained fairly similar results, however, GC-RANSAC is slightly more accurate. It is marginally slower due to the neighborhood computation. However, it is still faster than real time.

Convergence from a Not-All-Inlier Sample. Table 5.8 reports the frequencies when a "not-all-inlier" sample led to the correct model. For lines (**L**), it is computed using 1000 runs on each outlier (100, 500 and 1000) and noise level (from 0.0 up to 9.0 pixels). Thus 15000 runs were performed. A minimal sample is counted as a "not-all-inlier" if it contains at least one point farther from the ground truth model than the ground truth noise σ.

For fundamental matrices (**F**), the frequencies of success from a "not-all-inlier" sample are computed as the mean of 1000 runs on all pairs of the `AdelaideRMF` dataset. In this dataset, all inliers are labeled manually, thus it is easy to check whether a sample point is inlier or not.

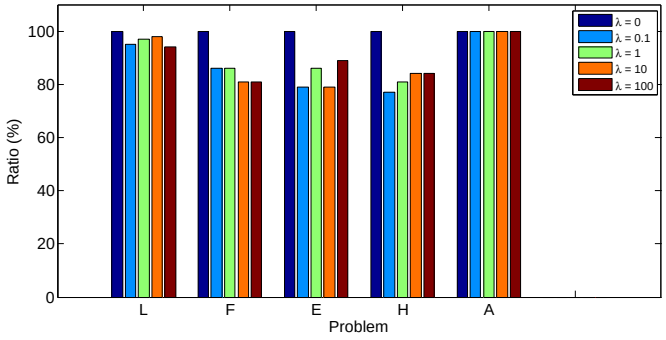
Evaluation of the λ setting. To evaluate the effect of the λ parameter balancing the spatial coherence term, we applied GC-RANSAC to all problems with varying λ. The evaluated values are: (i) $\lambda = 0$, which turns off the spatial coherence term, (ii) $\lambda = 0.1$, (iii) $\lambda = 1$, (iv) $\lambda = 10$, and (v) $\lambda = 100$. Fig. 5.10a shows the ratio of the geometric errors for $\lambda \neq 0$ and $\lambda = 0$ (in percent). For all investigated non-zero λ values, the error is lower than for $\lambda = 0$. Since $\lambda = 0.1$ led to the most accurate results on average, we chose this setting in the tests.

Evaluation of the criterion for the local optimization. The proposed criterion (Eq. 5.12) ensuring that local optimization is applied only to the most promising model candidates is tested in this section. We applied GC-RANSAC to all problems combined with the proposed and the standard approaches. The standard technique sets an iteration limit (default value: 50) and the LO procedure is afterwards applied

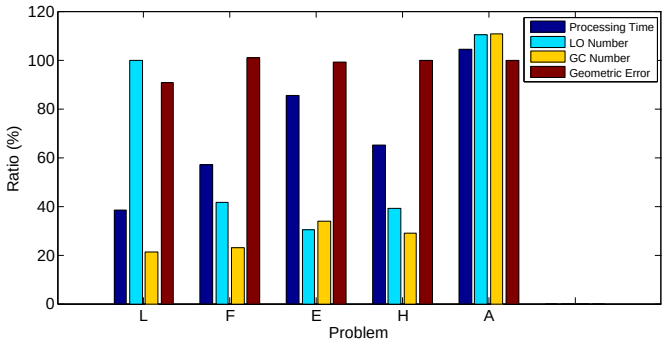
[13] http://mplab.sztaki.hu/remotesensing

TABLE 5.9: Fundamental matrix estimation applied to `kusvod2` (24 pairs), `AdelaideRMF` (19 pairs) and `Multi-H` (4 pairs) datasets, homography estimation on `homogr` (16 pairs) and `EVD` (15 pairs) datasets, essential matrix estimation on the `strecha` dataset (467 pairs), and affine transformation estimation on the `SZTAKI Earth Observation` benchmark (52 pairs). Thus the methods were tested on total on 597 image pairs. The datasets, the problem (**F**/**H**/**E**/**A**), the number of the image pairs (#) and the reported properties are shown in the first three columns. The next five report the results at 99% confidence with a time limit set to 60 FPS, i.e. the run is interrupted after $1/60$ secs (EP-RANSAC is removed since it cannot be applied in real time). For the other columns, there was no time limit but the confidence was set to 95%. Values are the means of 1000 runs. LO is the number of local optimizations and the number of graph-cut runs are shown in brackets. The geometric error ($\mathcal{E}$, in pixels) of the estimated model w.r.t. the manually selected inliers is written in each second row; the mean processing time ($\mathcal{T}$, in milliseconds) and the required number of samples ($\mathcal{S}$) are written in every 3th and 4th rows. The geometric error is the Sampson distance for **F** and **E**, and the projection error for **H** and **A**.

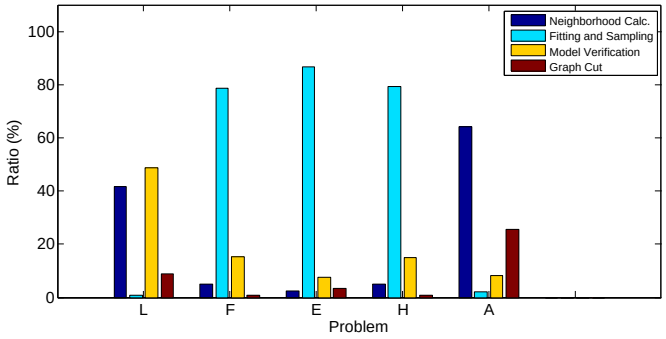
			Approx. 60 FPS (or 99% confidence)					Confidence 95%					
			RSC	LO	LO$^+$	LO′	**GC**	RSC	LO	LO$^+$	LO′	EP-RSC	**GC**
`kusvod2`	**F**, #24	LO	–	2	2	2	1 (3)	–	1	1	1	–	2 (3)
		$\mathcal{E}$	5.01	4.95	4.97	5.02	**4.65**	5.18	5.08	5.03	5.22	7.87	**4.69**
		$\mathcal{T}$	6.2	6.1	6.3	5.9	**4.6**	4.9	5.2	5.1	4.9	439.9	**3.6**
		$\mathcal{S}$	117	96	99	111	**70**	93	76	78	87	–	**53**
`Adelaide`	**F**, #19	LO	–	2	2	2	1 (3)	–	2	2	3	–	2 (4)
		$\mathcal{E}$	0.55	0.53	0.52	0.55	**0.50**	0.44	0.45	**0.43**	0.44	0.71	**0.43**
		$\mathcal{T}$	14.2	14.8	14.9	**14.1**	18.9	262.7	**194.2**	210.9	237.1	2 121.9	227.1
		$\mathcal{S}$	124	**113**	**113**	122	116	1 363	1 126	1 205	1 305.00	–	**1 115**
`Multi-H`	**F**, #4	LO	–	1	1	1	1 (3)	–	2	1	2	–	1 (3)
		$\mathcal{E}$	0.35	0.34	0.34	0.34	**0.32**	0.33	0.33	0.33	0.34	0.44	**0.32**
		$\mathcal{T}$	**10.3**	11.5	11.1	**10.3**	14.6	12.8	15.1	14.1	**12.4**	2 371.8	36.0
		$\mathcal{S}$	83	76	76	82	**74**	107	89	90	100	–	**78**
`EVD`	**H**, #15	LO	–	2	2	2	2 (2)	–	4	4	4	–	3 (6)
		$\mathcal{E}$	1.53	1.63	**1.51**	1.58	1.53	0.96	0.95	0.95	0.96	1.17	**0.92**
		$\mathcal{T}$	16.8	18.3	18.0	**16.8**	19.2	247.3	248.0	251.3	**247.0**	$> 10^4$	249.9
		$\mathcal{S}$	320	**298**	301	318	301	4 303	**4 203**	4 248	4 291	–	4 204
`homogr`	**H**, #16	LO	–	2	2	2	1 (3)	–	2	2	2	–	1 (4)
		$\mathcal{E}$	0.53	0.53	0.53	0.53	**0.51**	0.50	0.50	0.49	0.50	0.58	**0.47**
		$\mathcal{T}$	7.1	10.4	9.8	**7.1**	7.6	17.1	10.1	9.9	8.5	3 339.7	**7.9**
		$\mathcal{S}$	193	175	175	189	**159**	450	212	214	226	–	**165**
`strecha`	**E**, #467	LO	–	1	1	1	1 (1)	–	7	7	7	–	7 (7)
		$\mathcal{E}$	11.81	12.34	12.07	12.12	**11.6**	3.03	2.95	2.94	2.87	3.32	**2.83**
		$\mathcal{T}$	11.6	17.3	**17.2**	**17.2**	17.3	3 581.9	3 638.5	3 648.4	3 570.0	$> 10^6$	**3 466.4**
		$\mathcal{S}$	31	**30**	31	31	**30**	3 654	3 646	**3 634**	3 653	–	3 651
`SZTAKI`	**A**, #52	LO	–	1	1	1	1 (3)	–	1	1	1	–	1 (3)
		$\mathcal{E}$	0.41	0.41	0.41	0.41	**0.40**	0.45	0.46	0.44	0.45	0.48	**0.41**
		$\mathcal{T}$	3.5	**3.2**	**3.2**	**3.2**	10.3	**1.7**	**1.7**	**1.7**	**1.7**	4 718.2	10.2
		$\mathcal{S}$	26	26	26	26	26	**9**	**9**	**9**	**9**	–	**9**



(a)



(b)



(c)

FIGURE 5.10: (**a**) The effect of the λ choice weighting the spatial term. The ratio of the geometric error (in percentage) compared to the $\lambda = 0$ case (no spatial coherence) for each problem (**L** – lines, **F** – fundamental matrix, **E** – essential matrix, **H** – homography, **A** – affine transformation). (**b**) The effect of replacing the iteration limit before the first LO applied with the proposed criterion, i.e. the confidence radically increases. The ratios (in percentage) of each property of the proposed and that of standard approaches. (**c**) The breakdown of the processing times in percentage w.r.t. the total runtime. All values were computed as the mean of all tests. *Best viewed in color.*

to all models that are so far the best. Fig. 5.10b reports the ratio of each property (processing time – dark blue, LO – light blue, and GC steps – yellow, geometric error – brown) of the proposed and standard approaches. The new criterion leads to significant improvement in the processing time with no deterioration in accuracy.

Processing Time. Fig. 5.10c shows the breakdown of the processing times of GC-RANSAC applied to each problem. The time demand of the neighborhood computation (dark blue) linearly depends on the point number. The light blue one is the time demand of the sampling and model fitting step, the yellow and brown bars show the model verification (support computation) and the proposed local optimization step, respectively. The sampling and model fitting part dominates the process.

5.4 Summary

GC-RANSAC was presented. It is more geometrically accurate than state-of-the-art methods. It runs in real-time for many problems at a speed approximately equal to the less accurate alternatives. It is much simpler to implement in a reproducible manner than any of the competitors (RANSAC's with local optimization). Its local optimization step is globally optimal for the so-far-the-best model parameters. We also proposed a criterion for the application of the local optimization step. This criterion leads to a significant improvement in processing time with no deterioration in accuracy. GC-RANSAC can be easily inserted into USAC [11] and be combined with its "bells and whistles" like PROSAC sampling, degeneracy testing and fast evaluation with early termination.

5.5 Multi-H: Efficient Recovery of Tangent Planes in Stereo Images

Understanding the structure of indoor and outdoor environments is important in many applications of computer vision. Man-made objects commonly consist of planar regions, particularly in an urban environment or indoor scenes. Many algorithms, for diverse problems, exploit the information captured by planes or planar correspondences. Such problems include camera calibration [70]–[72], robot navigation [74], [87], augmented reality [163] and 3D reconstruction [68], [69].

This paper addresses the problem of accurate tangent plane estimation by partitioning the feature correspondences satisfying the epipolar constraint according to the similarity of their tangent planes. A plane-to-plane correspondence in two images is defined by a homography [43] which can be estimated in many ways. Methods based on point [43], line [43], conic [79], [80], local affine frame [30] or region [77] correspondences have been proposed.

Several techniques are available for the estimation of multiple homographies. The popular RANSAC paradigm has been extended to multiple plane fitting by sequential RANSAC [137], [138] and multiRANSAC [139]. However, the RANSAC strategy suffers from the low inlier ratio of each individual homography. J-Linkage [83] and the recently proposed T-Linkage [84] are based on the analysis of randomly selected clusters in the preference space which is defined by the assignment costs of data points to clusters. J-Linkage merges the initial clusters in the order of their Jaccard distances i.e. the overlap between two sets. T-Linkage extends this approach to

a continuous preference space and modifies the distance function between two clusters to the Tanimoto distance. Both algorithms decide whether a plane is significant on the basis of the number of the associated inliers.

The closest work is the PEARL algorithm of Boykov et al. [13]. In PEARL, the multi-model fitting problem is cleanly formulated as optimization of a global energy functional. The hypothesizes are initialized by stochastic sampling. The data term of the energy functional captures the cost of a point to homography assignment. A second term introduces spatial regularization reflecting an assumption that the geometric models have non-overlapping spatial supports and that correspondences which are close are more likely to belong to the same model. A third term penalizes the number of the models.

Like PEARL, we formulate the problem as a search for energy minimizing labeling. The energy proposed here is similar: it consists of the same data and spatial regularization terms. However, in the proposed algorithm, called Multi-H, the third term of PEARL is omitted as we control the model complexity by a combination of Mean-Shift [164] and α-expansion [140].

Multi-H benefits from a deterministic initialization which we show that together with a repeated use of Mean-Shift leads to results superior to PEARL. The proposed method exploits the result of Barath et al. [30] and estimates a homography from a single correspondence and the related affinity. Another strong point is that hard decisions whether a plane is significant or not are avoided since that depends on the application field. Small planes are beneficial e.g. for reconstruction, however, we introduce a significance criterion for the problem of dominant plane retrieval.

The contributions of the paper are: **(1)** the method for assigning point correspondences to planes according to the similarity of their tangents that leads to high-quality estimates of surface normals. Not deciding whether a plane is significant, we benefit from both weakly and strongly supported planes. **(2)** It is shown that the common stochastic sampling stage of multi-homography fitting algorithms can be improved upon. The Multi-H partitioning significantly outperforms state-of-the-art multi-homography fitting techniques. **(3)** We introduce new, more challenging image pairs for multi-homography estimation and make them publicly available together with the annotation[14].

5.5.1 Multiple Homography Estimation – Multi-H

Multi-H estimates tangent plane parameters at each point correspondence by assigning them to shared planes. Its only required input is an image pair. The output of the algorithm is a set of homographies defining the tangent planes and a label for each point correspondence associating it to a homography.

Point Correspondences with Local Affine Transformations. Several methods are available for the estimation of a local affine transformation at a detected point pair. We prefer to use affine-covariant feature detectors [57] since they provide point correspondences and affinities at the same time. We use MODS[15] [46] since it is significantly faster than ASIFT [2]. MODS provides high quality local affine transformations as well as the epipolar geometry $\mathbf{F}$. The output point correspondences are consistent with fundamental matrix $\mathbf{F}$. A different source of point correspondences with local affinities can be used, but the transformations must be consistent with $\mathbf{F}$ since Multi-H exploits this property.

[14] http://web.eee.sztaki.hu/~dbarath/

[15] Available at http://cmp.felk.cvut.cz/wbs/

Let us denote the ith homogeneous point in kth image with $\mathbf{p}_k^i = [p_k^{i,x} \quad p_k^{i,y} \quad 1]^\mathrm{T}$, $i \in [1, n]$, $k \in \{1, 2\}$, and the related local affinity with $\mathbf{A}_k^i$. The transformation between the infinitely close vicinities of the two points is the one transforming the first affinity to the second as $\mathbf{A}^i\mathbf{A}_1^i = \mathbf{A}_2^i$. Thus $\mathbf{A}^i = \mathbf{A}_2^i(\mathbf{A}_1^i)^{-1}$. The elements of $\mathbf{A}^i$ in row-major order are a_{11}^i, a_{12}^i, a_{21}^i, and a_{22}^i. Fig. 5.11 visualizes some local

FIGURE 5.11: Corresponding local affine transformations visualized by ellipses.

affine transformations using ellipses. To make the measured affinities as accurate as possible, the EG-L_2-Opt correction is applied [165].

Homography $\mathbf{H}_i$ is calculated for every affine transformation $\mathbf{A}_i$ and the corresponding point pair by the Homography from Affine transformation and Fundamental matrix method (HAF) [30]. HAF estimates a homography from only one affine correspondence if the fundamental matrix is given by solving a system of linear, inhomogeneous equations $\mathbf{Cx} = \mathbf{b}$ with coefficient matrix

$$\mathbf{C} = \begin{bmatrix} a_{11}^i p_1^{i,x} + p_2^{i,x} - e^x & a_{11}^i p_1^{i,y} & a_{11}^i \\ a_{12}^i p_1^{i,y} + p_2^{i,x} - e^x & a_{12}^i p_1^{i,x} & a_{12}^i \\ a_{21}^i p_1^{i,x} + p_2^{i,y} - e^y & a_{21}^i p_1^{i,y} & a_{21}^i \\ a_{22}^i p_1^{i,y} + p_2^{i,y} - e^y & a_{22}^i p_1^{i,x} & a_{22}^i \end{bmatrix}, \tag{5.13}$$

where $\mathbf{e} = [e^x \quad e^y]^\mathrm{T}$ is the epipole on the second image. Vector $\mathbf{b} = [f_{21} \quad f_{22} \quad -f_{11} \quad -f_{12}]$ is the inhomogeneous part of the four equations and $\mathbf{x} = [h_{31} \quad h_{32} \quad h_{33}]^\mathrm{T}$ is the vector of the unknown parameters. The optimal solution in the least squares sense is given by $\mathbf{x} = \mathbf{C}^\dagger\mathbf{b}$ where $\mathbf{C}^\dagger$ is the Moore-Penrose pseudo-inverse of matrix $\mathbf{C}$. The homography matrix is finally calculated using its last row [30] as follows: $h_{1j} = e^x h_{3j} + f_{2j}$, $h_{2j} = e^y h_{3j} + f_{1j}$, where $j \in \{1, 2\}$ and $f_{lm}, l, m \in \{1, 2, 3\}$, are elements of the fundamental matrix $\mathbf{F}$.

Alternating Minimization. After the initialization described in the preceding section, the set of homographies is improved by alternating three steps (see Alg. 9).
(1) Mean-Shift. Fig. 5.12 shows that after initialization some of the homographies estimated from a single correspondence coincide with a surface tangent plane (columns one and two) and some do not (columns three and four). In each column of Fig. 5.12, the correspondence initializing the homography is marked green, and its ϵ-inliers are in red, with threshold $\epsilon = 3.0$ pixels. The tangent planes are visualized by blue quadrangles.

We assume that tangent plane homographies are shared by a number of points and their parameters emerge as modes in the homography space. Since we do not know the number of tangent planes in the scene, the mode-seeking Mean-Shift [164] algorithm is adopted. The projection of the ith homography in the constructed $6D$

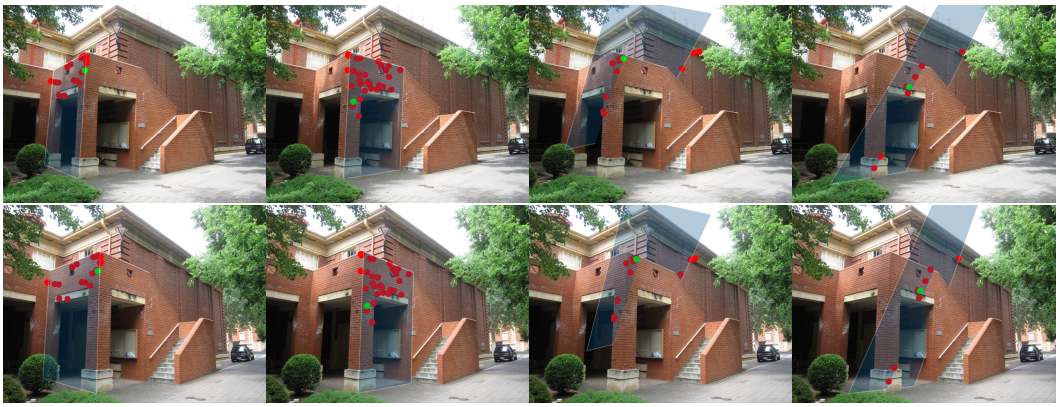


FIGURE 5.12: The images (top, bottom) of the `johnsona` pair. Blue shaded quadrangles visualise homographies coinciding (columns 1 and 2) and not coinciding (3 and 4) with a surface tangent plane. The correspondence initializing the homography is marked green. The red points are inliers obtained by thresholding the re-projection error at 3.0 pixels.

Algorithm 9 The Multi-H Algorithm.

Input: I_1, I_2 – images; $P, A, F := \text{MODS}(I_1, I_2)$ [46]
P - point correspondences; A – affine transformations; F – fundamental matrix
Output: $\mathcal{H}$ – obtained homographies; L – obtained labeling

1: $\mathcal{H}^0 := \text{HAF}(P, A, F)$ [30] ▷ Initialization with point-wise homographies
2: $i := 0$;
3: **repeat** ▷ Alternating Minimization
4: $i := i + 1$;
5: $\mathcal{H}^i := \text{MeanShift}(\mathcal{H}^{i-1})$ ▷ Default $\epsilon = 2.7$
6: $L^i := \alpha\text{-expansion}(P, \mathcal{H}^i)$ ▷ Default $\lambda = 0.5$, $\gamma = 0.005$
7: $\mathcal{H}^i := \text{LSQHomographyRefinement}(P, A, L^i, F)$
8: **until** Convergence ▷ if $\mathcal{H}^i = \mathcal{H}^{i-1} \wedge L^i = L^{i-1}$
9: $\mathcal{H} := \mathcal{H}^i$; $L := L^i$

homography space is

$$\mathbf{v}^i = \begin{bmatrix} w_1^{i,x} & w_1^{i,y} & w_2^{i,x} & w_2^{i,y} & w_3^{i,x} & w_3^{i,y} \end{bmatrix}, \tag{5.14}$$

where

$$\mathbf{w}_1^i = \frac{\mathbf{H}^i [0 \quad 0 \quad 1]^\text{T}}{h_{33}^i}, \quad \mathbf{w}_2^i = \frac{\mathbf{H}^i [1 \quad 0 \quad 1]^\text{T}}{h_{13}^i + h_{33}^i}, \quad \mathbf{w}_3^i = \frac{\mathbf{H}^i [0 \quad 1 \quad 1]^\text{T}}{h_{23}^i + h_{33}^i}.$$

The denominator of each $\mathbf{w}^i$ is the projective depth of the transformed point in the numerator. Each vector $\mathbf{v}^i$ determines a homography which can be recovered from three points $[0 \quad 0 \quad 1]^\text{T}, [1 \quad 0 \quad 1]^\text{T}, [0 \quad 1 \quad 1]^\text{T}$ and their projections if the fundamental matrix is known [30], [43]. Even though there are several possible representations for a homography (e.g. using its elements, projecting four points, etc.), we prefer to use a low-dimensional one – the processing time of Mean-Shift highly depends on the dimension of the problem. Since each coordinate pair $[v_k^i \quad v_{k+1}^i]$, $k \in \{1, 3, 5\}$, is

a given point projected by $\mathbf{H}^i$ the distance function d is chosen as the mean Euclidian distance between the three coordinate pairs where v^i_k is the kth coordinate of vector v^i. The distance between the ith and jth feature vectors is defined as

$$d(\mathbf{v}^i, \mathbf{v}^j) = \frac{1}{3}\sum_{k=1}^{3} ||[v^i_{2(k-1)+1} \quad v^i_{2(k-1)+2}]^\mathrm{T} - [v^j_{2(k-1)+1} \quad v^j_{2(k-1)+2}]^\mathrm{T}||_2.$$

(2) The α-expansion [140] step minimizes the following energy:

$$E(L) = \frac{1}{\lambda}E_\mathrm{d}(L) + \lambda E_\mathrm{s}(L), \tag{5.15}$$

where L is the current labeling, $E_\mathrm{d}(L)$ and $E_\mathrm{s}(L)$ the data and smoothness terms; λ controls their balance. The data term is defined as

$$E_\mathrm{d}(L) = \sum_{i=1}^{N} \|\mathbf{p}^i_2 - \frac{\mathcal{H}^{l_i}\mathbf{p}^i_1}{\mathcal{H}^{l_i}_{31}p^{i,x}_1 + \mathcal{H}^{l_i}_{32}p^{i,y}_1 + \mathcal{H}^{l_i}_{33}}\|_2, \tag{5.16}$$

where $\mathcal{H}^{l_i}$ is the homography associated with label $l_i \in L$ of the ith correspondence. The second term, E_s, reflects the assumption that neighboring points are more likely to belong to the same homography. E_s is equal to the number of neighboring points that are labeled differently:

$$E_\mathrm{s}(L) = \sum_{i=1}^{N}\sum_{j=1}^{N} \mathcal{A}_{ij}[\![l_i \neq l_j]\!], \tag{5.17}$$

where N is the number of correspondences, the Iverson bracket $[\![.]\!]$ is equal to one if the condition inside holds and zero otherwise, and the elements of the adjacency matrix $\mathcal{A}_{ij}$ are equal to 1 if correspondences ith and jth are spatial neighbors, 0 otherwise. The correspondences are considered to be neighbors if their distance in a $4D$ concatenated coordinate space – the vector associated with a correspondence is $[p^x_1 \quad p^y_1 \quad p^x_2 \quad p^y_2]^\mathrm{T}$ – is below γ, a control parameter. Matrix $\mathcal{A}$ is calculated efficiently using FLANN, the Fast Library for Approximate Nearest Neighbors [166].

The energy cannot increase in this step due to the nature of the α-expansion algorithm. A point is assigned to no plane if its distance from the closest one is greater than 3ϵ which is an empirically set threshold.

(3) The Least-Squares Homography Refinement runs the HAF method [30] on the correspondences associated with each homography by the current labeling. The number of the homographies is unchanged. The energy decreases or remains the same since E_d is the sum of the re-projection errors which are minimized. E_s is unchanged since the labeling does not change.

Convergence is reached when both the number of the clusters and the energy remain unchanged in two iterations. As the first stage does not increase the number of clusters, the other stages decrease the energy, and the set of labeling is finite, convergence is ensured. In the reported experiments, Alg. 9 converged no later than after eight iterations.

5.5.2 Experimental Results

Comparison with Multi-homography Fitting Techniques. In this section, Multi-H is tested on the problem of significant plane retrieval. and it outperforms the

TABLE 5.10: Misclassification error (%) for the two-view plane segmentation. The selected image pairs are a subset – the same as used in [84] – of the 19 pairs of AdelaideRMF dataset. The number of the ground truth planes is denoted with *R*.

	R	PEARL	QP-MF	FLOSS	ARJMC	SA-RCM	J-Lnkg	T-Lnkg	**Multi-H**
johnsonna	4	4.02	18.50	4.16	6.48	5.90	5.07	4.02	**2.41**
johnsonnb	7	18.18	24.65	18.18	21.49	17.95	18.33	18.17	**4.46**
ladysymon	2	5.49	18.14	5.91	5.91	7.17	9.25	5.06	**0.00**
neem	3	5.39	31.95	5.39	8.81	5.81	3.73	3.73	**0.00**
oldclassicswing	2	1.58	13.72	1.85	1.85	2.11	0.27	0.26	**0.00**
sene	2	0.80	14.00	0.80	0.80	0.80	0.84	0.40	**0.00**
mean		5.91	20.16	6.05	7.56	6.62	6.25	5.30	**1.19**
median		4.71	18.32	4.78	6.20	5.86	4.40	3.87	**0.00**

TABLE 5.11: Two-view plane segmentation. Mean and median misclassification error (%) on the 19 image pairs of the AdelaideRMF dataset.

	J-Lnkg	T-Lnkg	RPA	SA-RCM	Grdy-RansaCov	ILP-RansaCov	**Multi-H**
avg	25.50	24.66	17.20	28.30	26.85	12.91	**4.40**
med	24.48	24.53	17.78	29.40	28.77	12.34	**2.41**

state-ofthe-art multi-homography fitting techniques.

Determination of significant planes. To determine whether a detected plane is or is not significant without strict restrictions on the minimum number of inliers, the following algorithm is introduced. (1) First, planes with less than four inliers are removed. (2) The homographies are re-computed using the standard normalized 4-point algorithm [43] followed by a numerical refinement stage minimizing the re-projection error by Levenberg-Marquardt optimization. (3) The compatibility constraint [43] for a homography and a fundamental matrix: $\mathbf{H}^{\mathrm{T}}\mathbf{F} + \mathbf{F}^{\mathrm{T}}\mathbf{H} = 0$ is imposed by removing $\mathbf{H}_i$ for which $||\mathbf{H}_i^{\mathrm{T}}\mathbf{F} + \mathbf{F}^{\mathrm{T}}\mathbf{H}_i||_F > \theta$. After extensive experimentation we set $\theta = 1.0$.

Multi-H is tested as in [84] on the AdelaideRMF dataset. For each image pair in the dataset, a set of dominant planes and point pairs on them are provided. However, affine transformations for the point pairs are not available. Thus as many correspondences and affinities as possible are obtained by MODS [46]. Then the closest match for every annotated AdelaideRMF correspondence is found among the MODS correspondences. These correspondences with the local affine transformations are the input of Multi-H.

The misclassification error (ME) is calculated as follows. First, the mapping between the ground truth $l_{\mathrm{gt}} \in L_{\mathrm{gt}}$ and Multi-H $l \in L$ labels is established. We use an iterative method, always assigning the Multi-H output homography with the highest set overlap of correspondences. The assigned Multi-H homography and ground truth one maximizing the overlap are then removed from further consideration. Note that if the assignment is not optimal, the reported misclassification errors of Multi-H are over-estimated. ME is the ratio of the number of different labels $\sum_{i=1}^{n} [\![l_{\mathrm{gt}}^i \neq l^i]\!]$ and the number of ground truth correspondences n.

Multi-H is compared with T-Linkage [84], ARJMC [167], PEaRL [13], QP-MF [168], FLoSS [169], J-Linkage [83] and SA-RCM [170] in Experiment 1 (see Table 5.10). Every algorithm, including Multi-H, has been tuned separately on each image pair. We prefer reporting results for a setting fixed for the whole dataset, and we do that at the end of this section, but to allow comparison with the literature we followed the per-image-parameter-setting methodology. Table 5.10 shows that Multi-H obtains the lowest mean and median misclassification errors on the six test image pairs

FIGURE 5.13: Resulting partitioning of Multi-H on the AdelaideRMF dataset. Planes are denoted by colour. There are a few misclassified points (on the top-left and top-middle images around the edges). They are denoted by small, filled, black circles. Best viewed in colour.

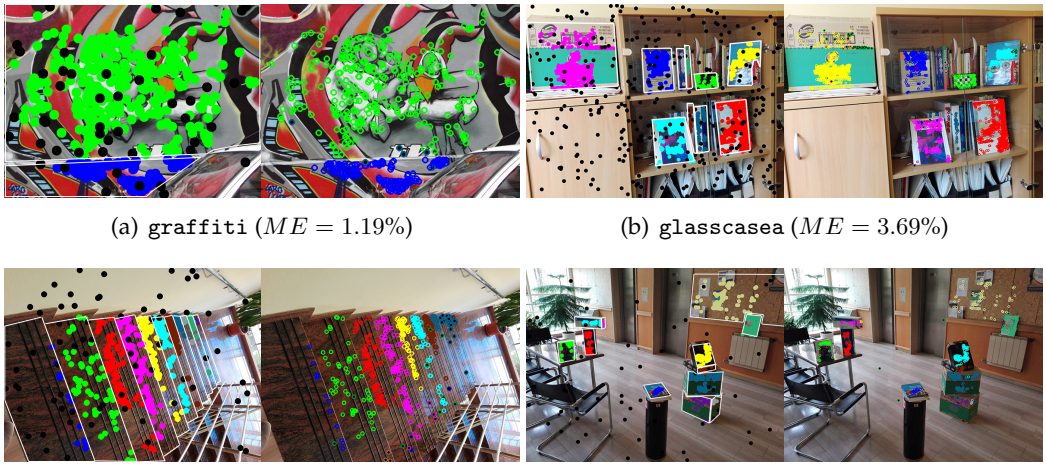
evaluated in the literature [84]. Fig. 5.13 shows the Multi-H points color-coded by the homography they were assigned to.

Table 5.11 shows the mean and median misclassification errors on all 19 image pairs of the AdelaideRMF dataset. The competitor methods are T-Linkage [84], J-Linkage [83], RPA [141], SA-RCM [170], Greedy-RansaCov [14] and ILP-RansaCov [14]. Multi-H significantly outperforms all published methods. Note the significant difference in the mean and median misclassification rates obtained on the six selected image, which are commonly published (Table 5.10), and on the full dataset.

Even though this dataset is the most frequently used one in the multi-plane fitting literature, it consists of easy scenes where the planes are perpendicular or far from each other. In order to test the accuracy of Multi-H, we created a more challenging dataset. Examples of the new images are visualized in Fig. 5.14. On these images, point correspondences are detected by MODS [46] and each is manually annotated to the containing plane. Finally, outliers, i.e. non-corresponding point pairs, are added to the data. For every image pair, the first image is the ground truth and the second one is the obtained planar partitioning. Outliers are visualised by black dots on the ground truth images. Pair 5.14(a) is from the well-known `graffiti` test sequence[16]. Two slightly different planes present in these images. The lower plane is closer to the camera than the upper one, however, the difference is very small. Even so, Multi-H accurately distinguishes the two planes and achieves a low misclassification error of 1.19%. Image pairs 5.14(b) and 5.14(c) are a cabinet with books and a staircase viewed from above. The last two images (5.14(d)) visualize a room with some boxes and planar-like objects. These tests are more challenging than the ones containing buildings since the observed planar regions are very small and their orientations are in many cases similar, see e.g. the books in `glasscasea`.

Proposed general configuration. For practical point of view, it is desirable that a single setting of parameters of the method covers most common cases. Through extensive experimentation, we found that $\lambda = 0.5$, $\epsilon = 2.7$, and $\gamma = 0.005$ are a robust choice. Table 5.12 shows the misclassification error on the AdalaideRMF dataset

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



(a) `graffiti` ($ME = 1.19\%$) (b) `glasscasea` ($ME = 3.69\%$)

(c) `stairs` ($ME = 8.74\%$) (d) `boxesandbooks` (ME$= 3.14\%$)

FIGURE 5.14: Four image pairs of the new dataset. Points coloured according to tangent planes, manual annotation (left) and Multi-H assignment (right). ME is the misclassification error.

	johnsa	johnsb	ladysymon	neem	old	sene	mean	median
Multi-H	**9.33**	**10.14**	**4.49**	**2.00**	**1.79**	**0.00**	**4.79**	**3.74**
T-Lnkg	34.28	24.04	24.67	25.65	20.66	7.63	22.82	24.36
SA-RCM	36.73	16.46	39.50	41.45	21.30	20.20	29.27	29.02
RPA	10.76	26.76	24.67	19.86	25.25	0.42	17.95	22.27

TABLE 5.12: Misclassification error (%) with a fixed parameter setup, average over 5 runs. The following abbreviations are used: `johnsonna` (`johnsa`), `johnsonnb` (`johnsb`), `oldclassicswing` (`old`).

FIGURE 5.15: Correspondence clustering into tangent planes for frames 1, 2 of the fountain-P11 set. Planes denoted by colour, estimated surface normals visualized by white line segments.

Frames	1 – 2	3 – 5	1 – 5	6 – 8	5 – 9
Affine Detector	35.7 | 32.7	24.9 | 20.3	19.0 | 15.8	22.5 | 18.6	20.0 | 15.4
EG-L_2-Optimal	35.5 | 32.5	23.1 | 19.8	16.7 | 13.9	19.9 | 16.6	17.8 | 14.4
Multi-H	**14.4** | **9.4**	**9.0** | **7.5**	**7.0** | **5.8**	**8.8** | **7.3**	**7.1** | **5.7**

TABLE 5.13: Mean and median errors (in degrees) of estimated normals for selected image pairs.

(average of 5 runs). The results are significantly worse than those of the separately tuned ones (see Table 5.10), but much better than the performance of the competitor algorithms[17] with a fixed set-up.

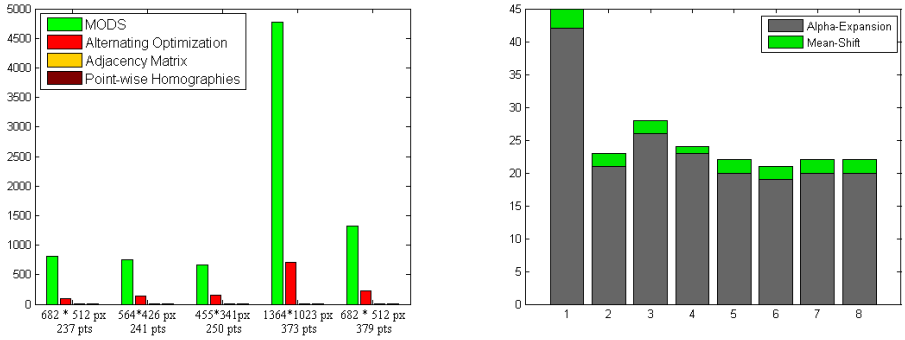
Evaluation of Surface Normal Accuracy. In this section, the accuracy of planes estimated by Multi-H is compared with the point-wise estimates of the affine-covariant detector. All planes returned by Multi-H are used, the significance constraint which was used in the previous section is not applied. The accuracy was assessed on the fountain-P11 dataset [63] which includes 11 images with resolution 3072×2048, projection and calibration camera matrices and reconstructed point clouds with surface normals. Point correspondences of MODS [46] between selected image pairs were obtained. On average, 920 correspondences were found.

Multi-H partitions correspondences on the basis of their tangent planes. The partitioning is visualized in Fig. 5.15. A single homography is fitted using the correspondences in the same tangent plane cluster. The normals at the correspondences are calculated from the homography as the camera parameters are known. Table 5.13 (row 3, Multi-H) shows the mean and median angular errors of the surface normals calculated from the homographies w.r.t. ground truth data. The surface normals determined by the homographies are significantly more accurate then the estimates from the initial local affine transformations output by the detector (Table 5.13, first row). Normals estimated after the EG-L_2-Optimal procedure [165], that improves the local affinities using constraints provided by the fundamental matrix, are significantly less accurate too (Table 5.13, second row).

Note that the projection matrices are used only for the evaluation of the results and not for the estimation of the tangent planes.

Processing Time and Implementation Details. The speed of the Multi-H procedure was measured on two sets consisting of 100 and 500 correspondences. Since

[17] Experimental results are copied from [141]



(a) (b)

FIGURE 5.16: **(a)** Processing time (in milliseconds) of Multi-H applied to different image pairs. The vertical axis at each column shows the resolution of the images and the correspondence number. **(b)** The processing time (in milliseconds) of iterations 1-8 of the alternating minimization on the `hartley` pair.

a randomized version of Mean-Shift was used, the algorithm ran 100 times. The mean number of iterations of Algorithm 9 was approx. 6 in both cases. The average processing times for the 100 and 500 correspondences were 0.04 and 0.80 sec. on a desktop PC with Intel Core i5-4690 CPU, 3.50 GHz using 4 cores.

Each column of Fig. 5.16(a) shows the processing time (in milliseconds) for an image pair. The parts of each bar visualize the time of the different algorithmic steps. The data shows that Multi-H has negligible time demand compared to the feature detection process (MODS). The bars associated with the calculation of the adjacency matrix and point-wise homographies cannot be seen since they require approx. $4-6$ milliseconds.

Fig. 5.16(b) presents the processing time of the alternating minimization. It significantly drops after the first iteration, then it is constant-like. The drop is caused by the Mean-Shift that reduces the number of homographies which speeds-up the α-expansion step.

Multi-H is implemented in C++. The GCOptimization[18] code was used for α-expansion. A fast Mean-Shift implementation was downloaded from the web[19].

5.5.3 Summary

The Multi-H approach for estimation of tangent planes in image pairs by partitioning feature correspondences was proposed. The method is accurate, outperforming state-of-the-art multi-homography fitting techniques for both fixed and per-image parameter setting. Experiments showed that the standard datasets are relatively easy and we therefore augmented the data with several challenging image pairs which we annotated.

In most applications, Multi-H will run significantly faster than the affine-covariant detectors providing the input. It is real-time on a standard CPU if the number of correspondences is below approx. 300. A GPU implementation of α-expansion [171] will be real-time capable for significantly larger problems.

[18] Available at `http://vision.csd.uwo.ca/code/`

[19] Available at `http://scikit-learn.org/stable/modules/clustering.html#mean-shift`

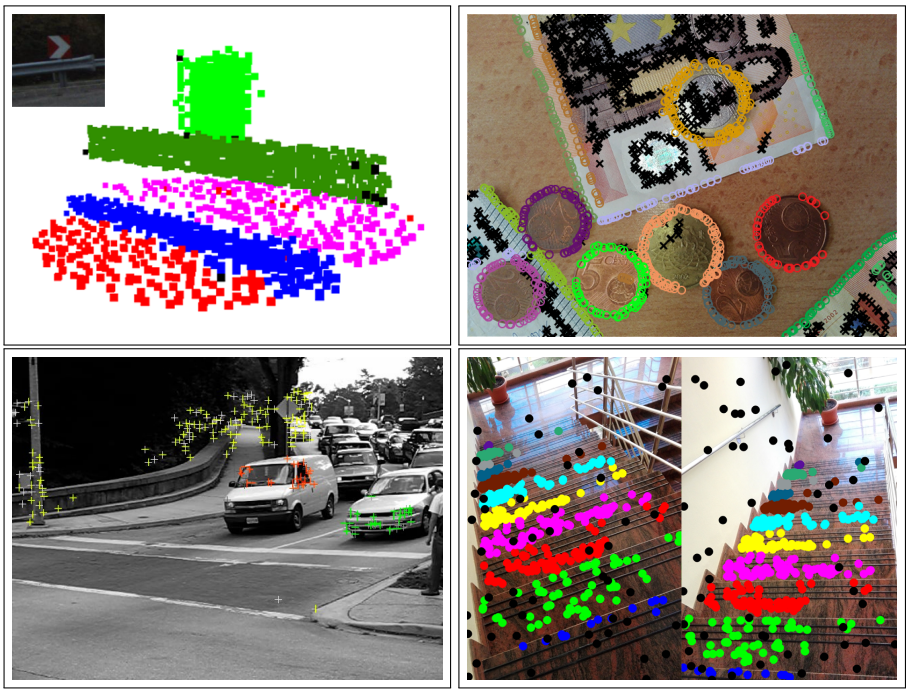


FIGURE 5.17: Multi-class multi-instance fitting examples. Results on simultaneous plane and cylinder (top left), line and circle fitting (top right), motion (bottom left) and plane segmentation (bottom right).

5.6 Multi-Class Model Fitting by Energy Minimization and Mode-Seeking

In multi-class fitting, the input data is interpreted as a mixture of noisy observations originating from multiple instances of multiple model classes, e.g. k lines and l circles in 2D edge maps, k planes and l cylinders in 3D data, multiple homographies or fundamental matrices from correspondences from a non-rigid scene (see Fig. 5.17). Robustness is achieved by considering assignment to an outlier class.

Multi-model fitting has been studied since the early sixties, the Hough-transform [12], [172] being the first popular method for extracting multiple instances of a single class [173]–[176]. A widely used approach for finding a single instance is RANSAC [1] which alternates two steps: the generation of instance hypotheses and their validation. However, extending RANSAC to the multi-instance case has had limited success. Sequential RANSAC detects instance one after another in a greedy manner, removing their inliers [137], [138]. In this approach, data points are assigned to the first instance, typically the one with the largest support for which they cannot be deemed outliers, rather than to the best instance. MultiRANSAC [139] forms compound hypothesis about n instances. Besides requiring the number n of the instances to be known a priori, the approach increases the size of the minimum sample and thus the number of hypotheses that have to be validated.

Most recent approaches [13], [14], [83], [84], [177] focus on the single class case: finding multiple instances of the same model class. A popular group of methods [13], [15], [27], [170], [178] adopts a two step process: initialization by RANSAC-like instance generation followed by a point-to-instance assignment optimization by *energy minimization* using graph labeling techniques [16]. Another group of methods uses *preference analysis*, introduced by RHA [17], which is based on the distribution of residuals of individual data points with respect to the instances [83], [84], [177].

The *multiple instance multiple class case* considers fitting of instances that are not necessarily of the same class. This generalization has received much less attention than the single-class case. To our knowledge, the last significant contribution is that of Stricker and Leonardis [18] who search for multiple parametric models simultaneously by minimizing description length using Tabu search.

The proposed Multi-X method finds multiple instances of multiple model classes drawing on progress in energy minimization extended with a new move in the label space: replacement of a set of labels with the corresponding density mode in the model parameter domain. Mode seeking significantly reduces the label space, thus speeding up the energy minimization, and it overcomes the problem of multiple instances with similar parameters, a weakness of state-of-the-art single-class approaches. The assignment of data to instances of different model classes is handled by the introduction of class-specific distance functions. Multi-X can also be seen as an extension or generalization of the Hough transform: (i) it finds modes of the parameter space density without creating an accumulator and locating local maxima there, which is prohibitive in high dimensional spaces, (ii) it handles multiple classes – running Hough transform for each model type in parallel or sequentially cannot easily handle competition for data points, and (iii) the ability to model spatial coherence of inliers and to consider higher-order geometric priors is added.

Most recent papers [14], [84], [179] report results tuned for each test case separately. The results are impressive, but input-specific tuning, i.e. semi-automatic operation with multiple passes, severely restricts possible applications. We propose an *adaptive parameter setting* strategy within the algorithm, allowing the user to run Multi-X as a black box on a range of problems with no need to set any parameters. Considering that outliers may form structures in the input, as a post-processing step, a cross-validation-based technique removes insignificant instances.

The contributions of the paper are: (i) A general formulation is proposed for multi-class multi-instance model fitting which, to the best of our knowledge, has not been investigated before. (ii) The commonly used energy minimizing technique, introduced by PEARL [13], is extended with a new move in the label space: replacing a set of labels with the corresponding density mode in the model parameter domain. Benefiting from this move, the minimization is speeded up, terminates with lower energy and the estimated model parameters are more accurate. (iii) The proposed pipeline combines state-of-the-art techniques, such as energy-minimization, median-based mode-seeking, cross-validation, to achieve results superior to the recent multi-model fitting algorithms both in terms of accuracy and processing time. Proposing automatic setting for the key optimization parameters, the method is applicable to various real world problems.

5.6.1 Multi-Class Formulation

Before presenting the general definition, let us consider a few examples of multi-instance fitting: to find a pair of *line instances* $h_1, h_2 \in \mathcal{H}_l$ interpreting a set of 2D points $\mathcal{P} \subseteq \mathbb{R}^2$. Line class $\mathcal{H}_l$ is the space of lines $\mathcal{H}_l = \{(\theta_l, \phi_l, \tau_l), \theta_l = [\alpha \quad c]^\mathrm{T}\}$ equipped with a distance function $\phi_l(\theta_l, p) = |\cos(\alpha)x + \sin(\alpha)y + c|$ ($p = [x \quad y]^\mathrm{T} \in \mathcal{P}$) and a function $\tau_l(p_1, ..., p_{m_l}) = \theta_l$ for estimating θ_l from $m_l \in \mathbb{N}$ data points. Another simple example is the fitting n *circle instances* $h_1, h_2, \cdots, h_n \in \mathcal{H}_c$ to the same data. The circle class $\mathcal{H}_c = \{(\theta_c, \phi_c, \tau_c), \theta_c = [c_x \quad c_y \quad r]^\mathrm{T}\}$ is the space of circles, $\phi_c(\theta_c, p) = |r - \sqrt{(c_x - x)^2 + (c_y - y)^2}|$ is a distance function and $\tau_c(p_1, ..., p_{m_c}) = \theta_c$ is an estimator. *Multi-line fitting* is the problem of finding multiple line instances $\{h_1, h_2, ...\} \subseteq \mathcal{H}_l$, while the *multi-class* case is extracting a subset $\mathcal{H} \subseteq \mathcal{H}_\forall$, where

$\mathcal{H}_\forall = \mathcal{H}_l \cup \mathcal{H}_c \cup \mathcal{H}_. \cup \cdots$. The set $\mathcal{H}_\forall$ is the space of all classes, e.g. line and circle. The formulation includes the outlier class $\mathcal{H}_o = \{(\theta_o, \phi_o, \tau_o), \theta_o = \emptyset\}$ where each instance has constant but possibly different distance to all points $\phi_o(\theta_o, p) = k$, $k \in \mathbb{R}^+$ and $\tau_o(p_1, ..., p_{m_o}) = \emptyset$. Note that considering multiple outlier classes allows interpretation of outliers askk originating from different sources.

Definition 2 (Multi-Class Model)**.** *The multi-class model is a space* $\mathcal{H}_\forall = \bigcup \mathcal{H}_i$*, where* $\mathcal{H}_i = \{(\theta_i, \phi_i, \tau_i) \mid d_i \in \mathbb{N}, \theta_i \in \mathbb{R}^{d_i}, \phi_i \in \mathcal{P} \times \mathbb{R}^{d_i} \to \mathbb{R}, \tau_i : \mathcal{P}^* \to \mathbb{R}^{d_i}\}$ *is a single class,* $\mathcal{P}$ *is the set of data points,* d_i *is the dimension of parameter vector* θ_i*,* ϕ_i *is the distance function and* τ_i *is the estimator of the* i*th class.*

The *objective of multi-instance multi-class model fitting* is to determine a set of instances $\mathcal{H} \subseteq \mathcal{H}_\forall$ and labeling $L \in \mathcal{P} \to \mathcal{H}$ assigning each point $p \in \mathcal{P}$ to an instance $h \in \mathcal{H}$ minimizing energy E. We adopt energy

$$E(L) = E_d(L) + w_g E_g(L) + w_c E_c(L) \tag{5.18}$$

to measure the quality of the fitting, where w_g and w_c are weights balancing the different terms described bellow, and E_d, E_c and E_g are the data, complexity terms, and the one considering geometric priors, e.g. spatial coherence or perpendicularity, respectively.

Data term $E_d : (\mathcal{P} \to \mathcal{H}) \to \mathbb{R}$ is defined in most energy minimization approaches as

$$E_d(L) = \sum_{p \in \mathcal{P}} \phi_{L(p)}(\theta_{L(p)}, p), \tag{5.19}$$

penalizing inaccuracies induced by the point-to-instance assignment, where $\phi_{L(p)}$ is the distance function of $h_{L(p)}$.

Geometric prior term E_g considers spatial coherence of the data points, adopted from [13], and possibly higher order geometric terms [15], e.g. perpendicularity of instances. The term favoring spatial coherence, i.e. close points more likely belong to the same instance, is defined as

$$E_g(L) : (\mathcal{P} \to \mathcal{H}) \to \mathbb{R} = \sum_{(p,q) \in N} w_{pq} [\![L(p) \neq L(q)]\!], \tag{5.20}$$

where N are the edges of a predefined neighborhood-graph, the Iverson bracket $[\![.]\!]$ equals to one if the condition inside holds and zero otherwise, and w_{pq} is a pairwise weighting term. In this paper, w_{pq} equals to one. For problems, where it is required to consider higher-order geometric terms, e.g. to find three perpendicular planes, E_g can be replaced with the energy term proposed in [15].

A regularization of the number of instances is proposed by Delong et al. [180] as a label count penalty $E_c(L) : (\mathcal{P} \to \mathcal{H}) \to \mathbb{R} = |L(\mathcal{P})|$, where $L(\mathcal{P})$ is the set of distinct labels of labeling function L. To handle multi-class models which might have different costs on the basis of the model class, we thus propose the following definition:

Definition 3 (Weighted Multi-Class Model)**.** *The weighted multi-class model is a space* $\widehat{\mathcal{H}}_\forall = \bigcup \widehat{\mathcal{H}}_i$*, where* $\widehat{\mathcal{H}}_i = \{(\theta_i, \phi_i, \tau_i, \psi_i) \mid d_i \in \mathbb{N}, \theta_i \in \mathbb{R}^{d_i}, \phi_i \in \mathcal{P} \times \mathbb{R}^{d_i} \to \mathbb{R}, \tau_i : \mathcal{P}^* \to \mathbb{R}^{d_i}, \psi_i \in \mathbb{R}\}$ *is a weighted class,* $\mathcal{P}$ *is the set of data points,* d_i *is the dimension of parameter vector* θ_i*,* ϕ_i *is the distance function,* τ_i *is the estimator, and* ψ_i *is the weight of the* i*th class.*

The term controlling the number of instances is

$$\widehat{E}_c(L) = \sum_{l \in L(\mathcal{P})} \psi_l, \tag{5.21}$$

instead of E_c, where ψ_l is the weight of the weighted multi-class model referred by label l.
Combining terms Eqs. 5.19, 5.20, 5.21 leads to **overall energy** $\widehat{E}(L) = E_d(L) + w_g E_g(L) + w_c \widehat{E}_c(L)$.

5.6.2 Replacing Label Sets

For the optimization of the previously described energy, we build on and extend the PEARL algorithm [13]. PEARL generates a set of initial instances applying a RANSAC-like randomized sampling technique, then alternates two steps until convergence:

(1) Application of α-expansion [140] to obtain labeling L minimizing overall energy $\widehat{E}$ w.r.t. the current instance set.

(2) Re-estimation of the parameter vector θ of each model instance in $\mathcal{H}$ w.r.t. labeling L.

In the PEARL formulation, the only way for a label to be removed, i.e. for an instance to be discarded, is to assign it to no data points. Experiments show that (i) this removal process is often unable to delete instances having similar parameters, (ii) and makes the estimation sensitive to the choice of label cost w_c. We thus propose a new move in the label space: replacing a set of labels with the density mode in the model parameter domain.

Multi-model fitting techniques based on energy-minimization usually generate a high number of instances $\mathcal{H} \subseteq \mathcal{H}_\forall$ randomly as a first step [13], [15] ($|\mathcal{H}| \gg |\mathcal{H}_{\text{real}}|$, where $\mathcal{H}_{\text{real}}$ is the ground truth instance set). Therefore, the presence of many similar instances is typical. We assume, and experimentally validate, that many points supporting the sought instances in $\mathcal{H}_{\text{real}}$ are often assigned in the initialization to a number of instances in $\mathcal{H}$ with similar parameters. The cluster around the ground truth instances in the model parameter domain can be replaced with the modes of the density (see Fig. 5.18).

Given a mode-seeking function $\Theta : \mathcal{H}^*_\forall \rightarrow \mathcal{H}^*_\forall$, e.g. Mean-Shift [164], which obtains the density modes of input instance set $\mathcal{H}_i$ in the ith iteration. The proposed move is as

$$\mathcal{H}_{i+1} := \begin{cases} \Theta(\mathcal{H}_i) & \text{if } E(L_{\Theta(\mathcal{H}_i)}) \leq E(L_i), \\ \mathcal{H}_i & \text{otherwise,} \end{cases} \tag{5.22}$$

where L_i is the labeling in the ith iteration and $L_{\Theta(\mathcal{H}_i)}$ is the optimal labeling which minimizes the energy w.r.t. to instance set $\Theta(\mathcal{H}_i)$. It can be easily seen, that Eq. 5.22 does not break the convergence since it replaces the instances, i.e. the labels, *if and only if* the energy does not increase. Note that clusters with cardinality one – modes supported by a single instance – can be considered as outliers and removed. This step reduces the label space and speeds up the process.

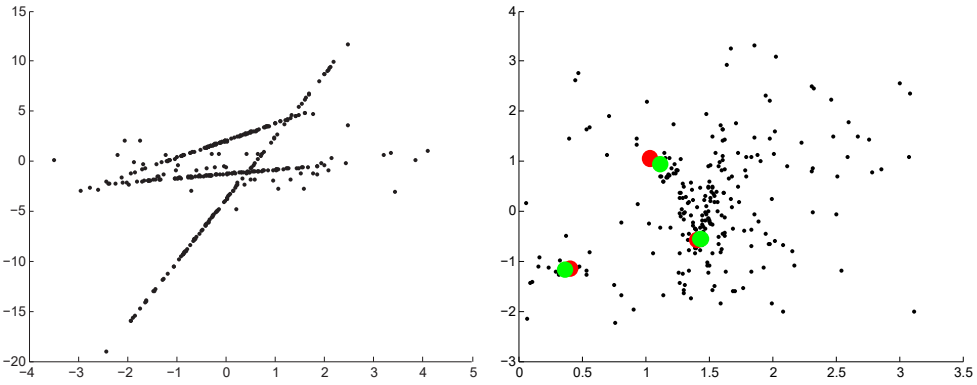


FIGURE 5.18: (**Left**) Three lines each generating 100 points with zero-mean Gaussian noise added, plus 50 outliers. (**Right**) 1000 line instances generated from random point pairs, the ground truth instance parameters (red dots) and the modes (green) provided by Mean-Shift shown in the model parameter domain: α angle – vertical, offset – horizontal axis.

5.6.3 Multi-X

The proposed approach, called Multi-X, combining PEARL, multi-class models and the proposed label replacement move, is summarized in Alg. 10. Next, each step is described.

Algorithm 10 Multi-X

Input: P – data points
Output: H^* – model instances, L^* – labeling

H_0 := InstanceGeneration(P); $i := 1$;
repeat
  H_i := ModeSeeking(H_{i-1}); ▷ by Median-Shift
  L_i := Labeling(H_i, P); ▷ by α-expansion
  H_i := ModelFitting(H_i, L_i, P); ▷ by Weiszfeld
  $i := i + 1$;
until !Convergence(H_i, L_i)
H^* := H_{i-1}, L^* := L_{i-1};
H^*, L^* := ModelValidation(H^*, L^*) ▷ Alg. 11

1. Instance generation step generates a set of initial instances before the alternating optimization is applied. Reflecting the assumption that the data points are spatially coherent, we use the guided sampling of NAPSAC [153]. This approach first selects a random point, then the remaining ones are chosen from the neighborhood of the selected point. The same neighborhood is used as for the spatial coherence term in the α-expansion. Note that this step can easily be replaced by e.g. PROSAC [116] for problems where the spatial coherence does not hold or favors degenerate estimates, e.g. in fundamental matrix estimation.

2. Mode-Seeking is applied in the model parameter domain. Suppose that a set of instances $\mathcal{H}$ is given. Since the number of instances in the solution – the modes in the parameter domain – is unknown, a suitable choice for mode-seeking is the

Mean-Shift algorithm [164] or one of its variants. In preliminary experiments, the most robust choice was the Median-Shift [128] using Weiszfeld- [130] or Tukey-medians [129]. There was no significant difference, but Tukey-median was slightly faster to compute. In contrast to Mean-Shift, it does not generate new elements in the vector space since it always return an element of the input set. With the Tukey-medians as modes, it is more robust than Mean-Shift [128]. However, we replaced Locality Sensitive Hashing [181] with Fast Approximated Nearest Neighbors [147] to achieve higher speed.

Reflecting the fact that a general *instance-to-instance* distance is needed, we represent instances by point sets, e.g. a line by two points and a homography by four correspondences, and define the *instance-to-instance* distance as the Hausdorff distance [182] of the point sets. Even though it yields slightly more parameters than the minimal representation, thus making Median-Shift a bit slower, it is always available as it is used to define spatial neighborhood of points. Another motivation for representing by points is the fact that having a non-homogeneous representation, e.g. a line described by angle and offset, leads to anisotropic distance functions along the axes, thus complicating the distance calculation in the mode-seeking.

There are many point sets defining an instance and a canonical point set representation is needed. For lines, the nearest point to the origin is used and a point on the line at a fixed distance from it. For a homography $\mathbf{H}$, the four points are $\mathbf{H}[0,0,1]^\mathrm{T}$, $\mathbf{H}[1,0,1]^\mathrm{T}$, $\mathbf{H}[0,1,1]^\mathrm{T}$, and $\mathbf{H}[1,1,1]^\mathrm{T}$. The matching step is excluded from the Hausdorff distance, thus speeding up the distance calculation significantly.[20]

The application of Median-Shift Θ_{med} never increases the number of instances $|\mathcal{H}_i|$: $|\Theta_{\text{med}}(\mathcal{H}_i)| \leq |\mathcal{H}_i|$. The equality is achieved *if and only if* the distance between every instance pair is greater than the bandwidth. Note that for each distinct model class, Median-Shift has to be applied separately. According to our experience, applying this label replacement move in the first iteration does not make the estimation less accurate but speeds it up significantly even if the energy slightly increases.

3. Labeling assigns points to model instances obtained in the previous step. A suitable choice for such task is α-expansion [140], since it handles an arbitrary number of labels. Given $\mathcal{H}_i$ and an initial labeling L_{i-1} in the ith iteration, labeling L_i is estimated using α-expansion minimizing energy $\widehat{E}$. Note that L_0 is determined by α-expansion in the first step. The number of the model instances $|\mathcal{H}_i|$ is fixed during this step and the energy must decreases: $\widehat{E}(L_i, \mathcal{H}_i) \leq \widehat{E}(L_{i-1}, \mathcal{H}_i)$. To reduce the sensitivity on the outlier threshold (as it was shown for the single-instance case in [156]), the distance function of each class is included into a Gaussian-kernel.

4. Model Fitting re-estimates the instance parameters w.r.t. the assigned points. The obtained instance set $\mathcal{H}_i$ is re-fitted using the labeling provided by α-expansion. The number of the model instances $|\mathcal{H}_i|$ is constant. L_2 fitting is an appropriate choice, since combined with the labeling step, it can be considered as truncated L_2 norm.

The overall energy $\widehat{E}$ can only decrease or stay constant during this step since it consists of three terms: (1) E_d – the sum of the assignment costs minimized, (2) E_g – a function of the labeling L_i, fixed in this step and (3) $\widehat{E}_c$ – which depends on $|H_i|$ so $\widehat{E}_c$ remains the same. Thus

$$\widehat{E}(L_i, \mathcal{H}_{i+1}) \leq \widehat{E}(L_i, \mathcal{H}_i). \tag{5.23}$$

[20] Details on the choice of model representation are provided in the supplementary material.

5. Model Validation considers that a group of outliers may form spatially coherent structures in the data. We propose a post-processing step to remove statistically insignificant models using cross-validation. The algorithm, summarized in Alg. 11, selects $aminimalsubsett$ times from the inlier points I. In each iteration, an instance is estimated from the selected points and its distance to each point is computed. The original instance is considered stable if the mean of the distances is lower than threshold γ. Note that γ is the outlier threshold used in the previous sections.

Algorithm 11 Model Validation.

Input: I – inlier points, t – trial number,
γ – outlier threshold ▷ default $t = 100$
Output: $R \in \{\text{true}, \text{false}\}$ – response

1: $\widehat{D} := 0$
2: **for** $i := 1$ to t **do**
3: MSS := SelectMinimalSubset(I)
4: H := ModelEstimation(MSS)
5: $\widehat{D} := \widehat{D} +$ MeanDistanceFromPoints(H, I) $/t$
6: $R := \widehat{D} < \gamma$

Automatic parameter setting is crucial for Multi-X to be applicable to various real world tasks without requiring the user to set most of the parameters manually. To avoid manual bandwidth selection for **mode-seeking**, we adopted the automatic procedure proposed in [183] which sets the bandwidth ϵ_i of the ith instance to the distance of the instance and its kth neighbor. Thus each instance has its own bandwidth set automatically on the basis of the input.

Label cost w_c is set automatically using the approach proposed in [15] as follows: $w_c = m \log(|\mathcal{P}|)/h_{\max}$, where m is the size of the minimal sample to estimate the current model, $|\mathcal{P}|$ is the point number and $h_{\max}$ is the maximum expected number of instances in the data. Note that this cost is not required to be high since mode-seeking successfully suppresses instances having similar parameters. The objective of introducing a label cost is to remove model instances with weak supports. In practice, this means that the choice of $h_{\max}$ is not restrictive.

Experiments show that the choice of the **number of initial instances** does not affect the outcome of Multi-X significantly. In our experiments, the number of instances generated was twice the number of the input points.

Spatial coherence weight w_g value 0.3 performed well in the experiments. The common problem-specific outlier thresholds which led to the most accurate results was: homographies (2.4 pixels), fundamental matrices (2.0 pixels), lines and circles (2.0 pixels), rigid motions (2.5), planes and cylinders (10 cm).

5.6.4 Experimental Results

First we compare Multi-X with PEARL [13] combined with the label cost of Delong et al. [180]. Then the performance of Multi-X applied to the following Computer Vision problems is reported: line and circle fitting, 3D plane and cylinder fitting to LIDAR point clouds, multiple homography fitting, two-view and video motion segmentation.

	(1)		(2)		(3)	
	FP	FN	FP	FN	FP	FN
PEARL [13]	1	**0**	3	**0**	5	3
T-Linkage [84]	**0**	1	1	3	**0**	6
RPA [177]	**0**	1	**0**	2	**0**	5
Multi-X	**0**	**0**	**0**	**0**	**0**	**1**

TABLE 5.14: The number of false positive (FP) and false negative (FN) instances for simultaneous line and circle fitting.

Comparison of PEARL and Multi-X. In a test designed to show the effect of the proposed label move, model validation was not applied and both methods used the same algorithmic components described in the previous section. A synthetic environment consisting of three 2D lines, each sampled at 100 random locations, was created. Then 200 outliers, i.e. random points, were added.

Fig. 5.19 shows the probability of returning an instance number for Multi-X (top-left) and PEARL (bottom-left). The numbers next to the vertical axis are the number of returned instances. The curve on their right shows the probability ($\in [0, 1]$) of returning them. For instance, the red curve for PEARL on the right of number 3 is close to the 0.1 probability, while for Multi-X, it is approximately 0.6. Therefore, Multi-X more likely returns the desired number of instances. The processing times (top-right), and convergence energies (bottom-right) are also reported. Values are plotted as the function of the initially generated instance number (horizontal axis; ratio w.r.t. to the input point number). The standard deviation of the zero-mean Gaussian-noise added to the point coordinates is 20 pixels. Reflecting the fact that the noise σ is usually not known in real applications, we set the outlier threshold to 6.0 pixels. The maximum model number of the label cost was set to the ground truth value, $h_{\max} = 3$, to demonstrate that suppressing instances exclusively with label cost penalties is not sufficient even with the proper parameters. It can be seen that Multi-X more likely returns the ground truth number of models, both its processing time and convergence energy are superior to that of PEARL.

For Fig. 5.20, the number of the generated instances was set to twice the point number, the threshold was set to 3 pixels. Each reported property is plotted as the function of the noise σ added to the point coordinates. The same trend can be seen as in Fig. 5.19: Multi-X is less sensitive to the noise than PEARL. It more often returns the desired instances, its processing time and convergence energy are lower.

Simultaneous Line and Circle Fitting is evaluated on 2D edges of banknotes and coins. Edges are detected by Canny edge detector and assigned to circles and lines manually to create a ground truth segmentation.[21]

Each method started with the same number of initial model instances: twice the data point (e.g. edge) number. The evaluated methods are PEARL [13], [178], T-Linkage [84][22] and RPA [177][23] since they can be considered as the state-of-the-art and their implementations are available. PEARL and Multi-X fits circles and lines simultaneously, while T-Linkage and RPA sequentially. Table 5.14 reports the number of false negative and false positive models. Multi-X achieved the lowest error for all test cases.

[21] Submitted as supplementary material.

[22] http://www.diegm.uniud.it/fusiello/demo/jlk/

[23] http://www.diegm.uniud.it/fusiello/demo/rpa/

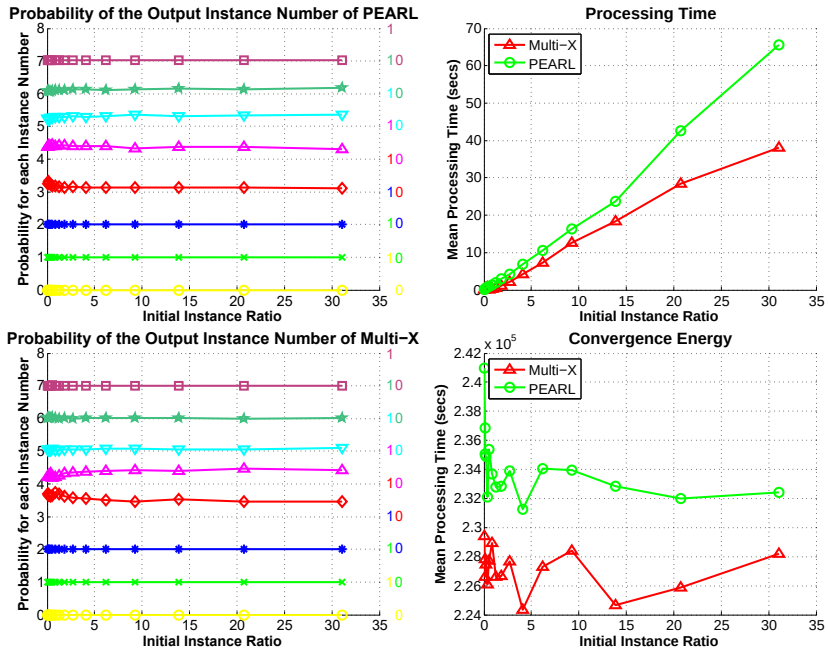


FIGURE 5.19: *Increasing instance number.* Comparison of PEARL and Multi-X. Three random lines sampled at 100 locations, plus 200 outliers. Parameters of both methods are: $h_{\max} = 3$, and the outlier threshold is (a) 6 and (b) 3 pixels. Zero-mean Gaussian noise with $\sigma = 20$ pixels added to the point coordinates. (**Left**) the probability of returning 0, ..., 7 instances (vertical axis) for PEARL (top) and Multi-X (bottom) plotted as the function of the ratio of the initial instance number and the point number (horizonal axis). (**Right**): the processing time in seconds and convergence energy.

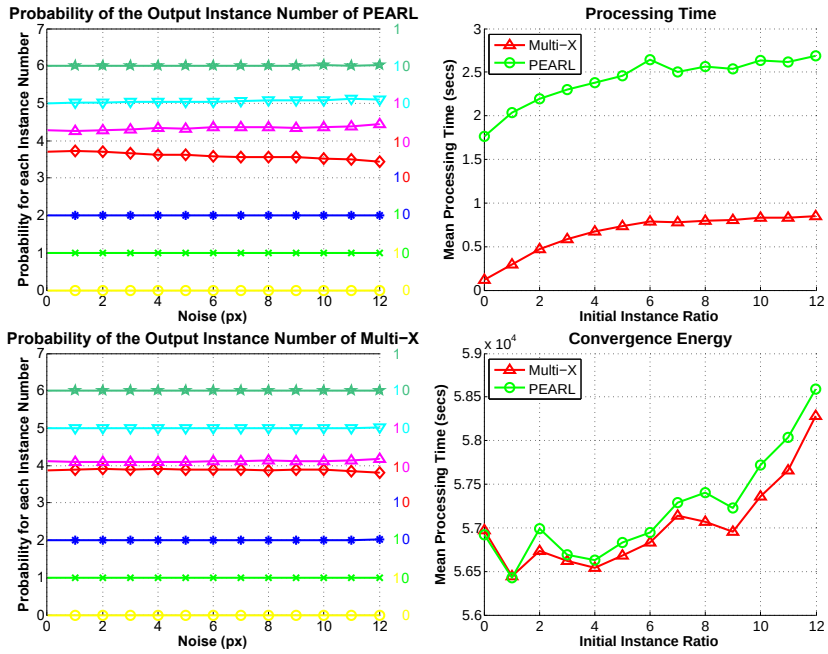


FIGURE 5.20: *Increasing noise.* Comparison of PEARL and Multi-X. Three random lines sampled at 100 locations, plus 200 outliers. Parameters of both methods are: $h_{\max} = 3$, and the outlier threshold is (a) 6 and (b) 3 pixels. The number of initial instances generated is twice the point number. (**Left**): the probability of returning instance numbers 0, ..., 7 (vertical axis) for PEARL (top) and Multi-X (bottom) plotted as the function of the noise σ (horizonal axis). (**Right**): the processing time in seconds and convergence energy.

Multiple Homography Fitting is evaluated on the AdelaideRMF homography dataset [98] used in most recent publications (see Fig. 5.21 for examples). AdelaideRMF consists of 19 image pairs of different resolutions with ground truth point correspondences assigned to planes (homographies). To generate initial model instances the technique proposed by Barath et al. [27] is applied: a single homography is estimated for each correspondence using the point locations together with the related local affine transformations. Table 5.15 reports the results of PEARL [140], FLOSS [169], T-Linkage [84], ARJMC [167], RCMSA [170], J-Linkage [83], and Multi-X. To allow comparison with the state-of-the-art, all methods, including Multi-X, are tuned separately for each test and only the same 6 image pairs are used as in [84].

Results using a fixed parameter setting are reported in Table 5.16 (results, except that of Multi-X, copied from [177]). Multi-X achieves the lowest errors. Compared to results in Table 5.15 for parameters hand-tuned for each problem, the errors are significantly higher, but automatic parameter setting is the only possibility in many applications. Moreover, per-image-tuning leads to overfitting.

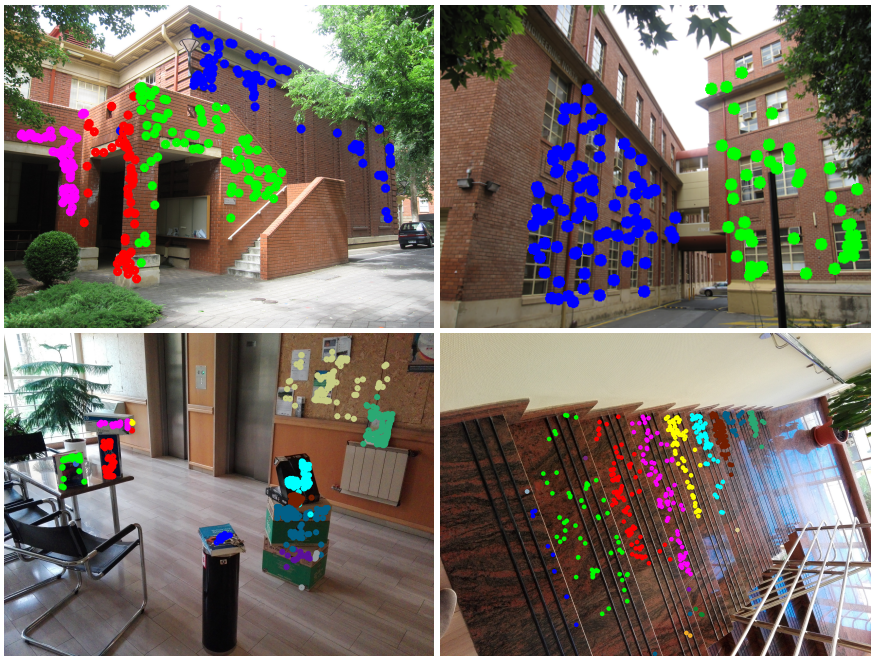


FIGURE 5.21: AdelaideRMF (top) and Multi-H (bot.) examples. Color indicates the plane Multi-X assigned a point to.

Two-view Motion Segmentation is evaluated on the AdelaideRMF motion dataset consisting of 21 image pairs of different sizes and the ground truth – correspondences assigned to their motion clusters.

Fig. 5.22 presents example image pairs from the AdelaideRMF motion datasets partitioned by Multi-X. Different motion clusters are denoted by color. Table 5.17 shows comparison with state-of-the-art methods when all methods are tuned separately for each test case. Results are the average and minimum misclassification errors (in percentage) of ten runs. All results except that of Multi-X are copied from [179]. For Table 5.18, all methods use fixed parameters. For both test types, Multi-X achieved higher accuracy than the other methods.

	# of planes	PEARL [13]	FLOSS [169]	T-Lnkg [84]	ARJMC [167]	RCMSA [170]	J-Lnkg [83]	**Multi-X**
(1)	4	4.02	4.16	4.02	6.48	5.90	5.07	**3.75**
(2)	6	18.18	18.18	18.17	21.49	17.95	18.33	**4.46**
(3)	2	5.49	5.91	5.06	5.91	7.17	9.25	**0.00**
(4)	3	5.39	5.39	3.73	8.81	5.81	3.73	**0.00**
(5)	2	1.58	1.85	0.26	1.85	2.11	0.27	**0.00**
(6)	2	0.80	0.80	0.40	0.80	0.80	0.84	**0.00**
Avg.		5.91	6.05	5.30	7.56	6.62	6.25	**1.37**
Med.		4.71	4.78	3.87	6.20	5.86	4.40	**0.00**

TABLE 5.15: Misclassification error (%) for the two-view plane segmentation on AdelaideRMF test pairs: (1) `johnsonna`, (2) `johnsonnb`, (3) `ladysymon`, (4) `neem`, (5) `oldclassicswing`, (6) `sene`.

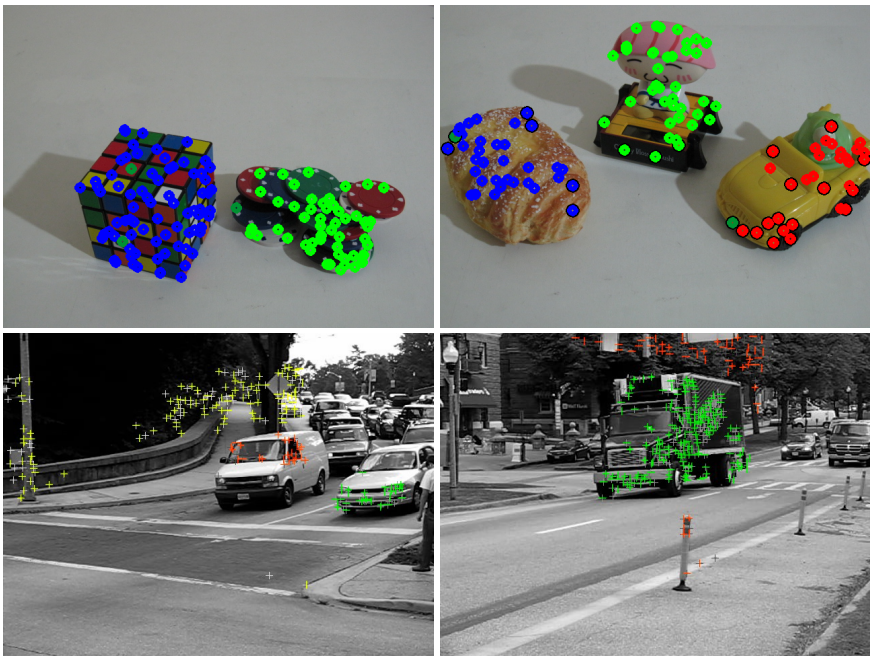


FIGURE 5.22: AdelaideRMF (top) and Hopkins (bot.) examples. Color indicates the motion Multi-X assigned a point to.

	T-Lnkg [84]	RCMSA [170]	RPA [177]	Multi-H [27]	**Multi-X**
Avg.	44.68	23.17	15.71	14.35	**9.72**
Med.	44.49	24.53	15.89	9.56	**2.49**

TABLE 5.16: Misclassification errors (%, average and median) for two-view plane segmentation on all the 19 pairs from AdelaideRMF test pairs using fixed parameters.

	KF [184]		RCG [185]		T-Lnkg [84]		AKSWH [186]		MSH [179]		**Multi-X**	
	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.
(1)	8.42	4.23	13.43	9.52	5.63	2.46	4.72	2.11	3.80	2.11	**3.45**	**1.41**
(2)	12.53	2.81	13.35	10.92	5.62	4.82	7.23	4.02	3.21	1.61	**2.27**	**0.40**
(3)	14.83	4.13	12.60	8.07	4.96	1.32	5.45	1.42	2.69	0.83	**1.45**	**0.41**
(4)	13.78	5.10	9.94	3.96	7.32	3.54	7.01	5.18	3.72	1.22	**0.61**	**0.30**
(5)	16.87	14.55	26.51	19.54	**4.42**	4.00	9.04	8.43	6.63	4.55	5.24	**1.80**
(6)	16.06	14.29	16.87	14.36	1.93	1.16	8.54	4.99	1.54	1.16	**0.62**	**0.00**
(7)	33.43	21.30	26.39	20.43	**1.06**	0.86	7.39	3.41	1.74	0.43	5.32	**0.00**
(8)	31.07	22.94	37.95	20.80	3.11	3.00	14.95	13.15	4.28	3.57	**2.63**	**1.52**

TABLE 5.17: Misclassification errors (%) for two-view motion segmentation on the AdelaideRMF dataset. All the methods were tuned separately for each video by the authors. Tested image pairs: (1) `cubechips`, (2) `cubetoy`, (3) `breadcube`, (4) `gamebiscuit`, (5) `breadtoycar`, (6) `biscuitbookbox`, (7) `breadcubechips`, (8) `cubebreadtoychips`.

	RPA [177]	RCMSA [170]	T-Lnkg [84]	AKSWH [186]	**Multi-X**
Avg.	5.62	9.71	43.83	12.59	**2.97**
Med.	4.58	8.48	39.42	11.57	**0.00**

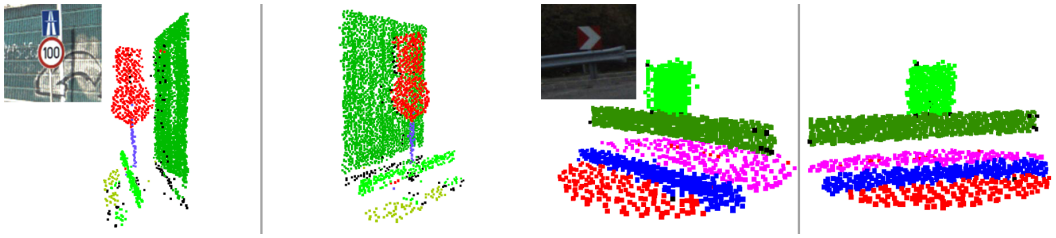
TABLE 5.18: Misclassification errors (%, average and median) for two-view motion segmentation on all the 21 pairs from the AdelaideRMF dataset using fixed parameters.

Simultaneous Plane and Cylinder Fitting is evaluated on LIDAR point cloud data (see Fig. 5.23). The annotated database consists of traffic signs, balusters and the neighboring point clouds truncated by a 3-meter-radius cylinder parallel to the vertical axis. Points were manually assigned to signs (planes) and balusters (cylinders).

Multi-X is compared with the same methods as in the line and circle fitting section. PEARL and Multi-X fit cylinders and planes simultaneously while T-Linkage and RPA sequentially. Table 5.19 reports that Multi-X is the most accurate in all test cases except one.

Video Motion Segmentation is evaluated on 51 videos of the Hopkins dataset [187]. Motion segmentation in video sequences is the retrieval of sets of points undergoing rigid motions contained in a dynamic scene captured by a moving camera. It can be seen as a subspace segmentation under the assumption of affine cameras. For affine cameras, all feature trajectories associated with a single moving object lie in a 4D linear subspace in $\mathbb{R}^{2f}$, where f is the number of frames [187].

Table 5.20 shows that the proposed method outperforms the state-of-the-art: SSC [188], T-Linkage [84], RPA [177], Grdy-RansaCov [14], ILP-RansaCov [14], and J-Linkage [83]. Results, except for Multi-X, are copied from [14]. Fig. 5.22 shows two frames of the tested videos.



(a) (b)

FIGURE 5.23: Results of simultaneous plane and cylinder fitting to LIDAR point cloud in two scenes. Segmented scenes visualized from different viewpoints. There is only one cylinder on the two scenes: the pole of the traffic sign on the top. Color indicates the instance Multi-X assigned a point to.

	PEARL [13]	T-Lnkg [84]	RPA [177]	**Multi-X**
(1)	10.63	57.46	46.83	**8.89**
(2)	10.88	41.79	53.39	**4.72**
(3)	37.34	52.97	61.64	**2.84**
(4)	38.13	38.91	41.41	**19.38**
(5)	17.20	51.83	53.34	**16.83**
(6)	**17.35**	61.77	51.21	21.72
(7)	6.12	12.49	80.45	**5.72**

TABLE 5.19: Misclassification error (%) of simultaneous plane and cylinder fitting to LIDAR data. See Fig. 5.23 for examples.

		(1)	(2)	(3)	(4)	(5)
SSC [188]	Avg.	0.06	0.76	3.95	2.13	1.08
	Med.	**0.00**	**0.00**	**0.00**	2.13	**0.00**
T-Lnkg [84]	Avg.	1.31	0.48	6.47	5.32	2.47
	Med.	**0.00**	0.19	2.38	5.32	**0.00**
RPA [177]	Avg.	0.14	0.19	4.41	9.11	1.42
	Med.	**0.00**	**0.00**	2.44	9.11	**0.00**
Grdy-RC [14]	Avg.	7.48	28.65	8.75	14.89	10.91
	Med.	**0.00**	1.53	0.20	14.89	**0.00**
ILP-RC [14]	Avg.	0.54	0.35	2.40	2.13	0.98
	Med.	**0.00**	0.19	1.30	2.13	**0.00**
J-Lnkg [83]	Avg.	1.75	1.58	5.32	6.91	2.70
	Med.	**0.00**	0.34	1.30	6.91	**0.00**
Multi-X	Avg.	**0.05**	**0.09**	**0.32**	**1.06**	**0.16**
	Med.	**0.00**	**0.00**	**0.00**	**1.06**	**0.00**

TABLE 5.20: Misclassification errors (%, average and median) for multi-motion detection on 51 videos of Hopkins dataset: (1) `Traffic2` – 2 motions, 31 videos, (2) `Traffic3` – 3 motions, 7 videos, (3) `Others2` – 2 motions, 11 videos, (4) `Others3` – 3 motions, 2 videos, (5) `All` – 51 videos.

	(1)		(2)		(3)		(4)		(5)	
#	**M**	T	**M**	T	**M**	T	**M**	T	**M**	T
100	0.1	0.4	0.1	0.3	0.1	0.3	0.0	0.2	0.1	0.4
500	2.0	14.0	3.2	8.4	2.1	8.4	0.8	7.0	3.8	15.9
1000	5.1	102.8	-	-	-	-	-	-	7.5	120.9

TABLE 5.21: Processing times (sec) of Multi-X (M) and T-Linkage (T) for the problem of fitting (1) lines and circles, (2) homographies, (3) two-view motions, (4) video motions, and (5) planes and cylinders. The number of data points is shown in the first column.

Processing Time. Multi-X is orders of magnitude faster than currently available Matlab implementations of J-Linkage, T-Linkage and RPA. Attacking the fitting problem with a technique similar to PEARL and SA-RCM, it is significantly faster since it benefits from high reduction of the number of instances in the Median-Shift step (see Table 5.21).

5.6.5 Summary

A novel multi-class multi-instance model fitting method has been proposed. It extends an energy minimization approach with a new move in the label space: replacing a set of labels corresponding to model instances by the mode of the density in the model parameter domain. Most of its key parameters are set adaptively making it applicable as a black box on a range of problems. Multi-X outperforms the state-of-the-art in multiple homography, rigid motion, simultaneous plane and cylinder fitting; motion segmentation; and 2D edge interpretation (circle and line fitting). Multi-X runs in time approximately linear in the number of data points, it is an order of magnitude faster than available implementations of commonly used methods.

Chapter 6

Conclusion

During the last few decades, the correspondence problem between images taken from significantly different viewpoints have been approached successfully by considering the warp between image regions. This warp can be locally approximated as an affine transformation and obtained by state-of-the-art affine covariant feature detectors. As a consequence, additional information about the underlying scene geometry, i.e. the rotation, scales along the axes and the shear, become available. However, with a few exceptions, the information which these local affine transformations encode is ignored in most of the geometric model estimation problems. In general, solely the centers of the corresponding regions are exploited for the estimation. Therefore, one of the main contributions of this thesis is the deepening of the knowledge about the application of affine correspondences in projective geometry.

Even though surface normal estimation from affine correspondences is considered to be an already solved problem, we showed that an optimal method, in the least squares sense, exists and it is solvable algebraically. We also proposed an extension for the multi-view case which makes the approach applicable in structure-from-motion pipelines. Combining the method with the well-known Patch-based Multi-view Stereo algorithm [133], we got significant improvement in the accuracy of the reconstructed dense point clouds.

Investigating homography estimation, we advanced the state-of-the-art minimal solver which exploits two affine correspondences to estimate a homography. The proposed method assume a rigid scene, thus having a fundamental matrix interpreting the camera motion, and provides a homography for each affinity independently. As a theoretical consequence, there is a *one-to-one relationship* between homographies and local affine transformations, therefore, they are equivalent for known fundamental matrix. Most of the feature detectors available in the field, provides more information about the underlying affine correspondence than just the point coordinates. Thus we generalized the estimation problem to describe the relationship of each affine component, i.e. scales, rotation and shear, and the homography, independently. The proposed method is able to make estimates from two partially known affine correspondences, if the rotation, a scale and the point coordinates in the two images are known.

We then showed the direct relationship of affine correspondences and epipolar geometry which was unknown to the best of our knowledge. The approach considers the mapping of the epipolar lines and thus establishes constraints on the epipolar geometry estimation directly. Exploiting the proposed relationship, we proposed methods (i) to make a measured local affinity consistent with the epipolar geometry, (ii) to estimate the essential matrix and (iii) for solving the semi-calibrated case, i.e. obtaining the fundamental matrix and the common focal length.

The second major part of the thesis investigates robust single- and multi-model fitting which is a fundamental component of computer vision pipelines. Mentioning

just only one example, Random Sample Consensus (RANSAC) is used dominantly for two-view geometry estimation and it is a part of some of its most successful applications like 3D reconstruction, image matching and retrieval.

The first algorithm we proposed aims to remove the outliers, i.e. invalid point matches, from a set of feature correspondences without considering an underlying model. As it is demonstrated, the proposed approach is applicable to non-rigid scenes and for rigid ones, combining it with model estimators makes it superior to the traditional approaches in terms of outlier rejection rate. The method is applicable in real time for most of the problems.

Then we advanced locally optimized RANSAC (LO-RANSAC) by replacing its local optimization step with energy minimization. It runs iteratively the graph cut algorithm in the local optimization (LO) step which is applied after a *so-far-the-best* model is found. The proposed LO step is conceptually simple, easy to implement, globally optimal and efficient. We demonstrated experimentally that GC-RANSAC outperforms LO-RANSAC and its state-of-the-art variants in terms of both accuracy and the required number of iterations for line, homography and fundamental matrix estimation on standard public datasets.

The final part of the thesis aims to solve multi-model estimation, i.e. the problem of interpreting the input data as a mixture of noisy observations originating from a single or multiple model classes. First, we approached the multi-homography fitting problem in two views, and proposed a method which is superior to the state-of-the-art in terms of accuracy on publicly available datasets. Then a more general case is considered: multi-class multi-instance fitting. The *multiple instance multiple class case* considers fitting of instances that are not necessarily of the same class. This generalization has received much less attention than the single-class case. To the best of our knowledge, the last significant contribution was that of Stricker and Leonardis [18]. The proposed Multi-X method finds multiple instances of multiple model classes drawing on progress in energy minimization extended with a new move in the label space: replacement of a set of labels with the corresponding mode in the model parameter domain. As it is demonstrated, this new move makes it outperforming the state-of-the-art single-class algorithms for various problems.

Appendix A

Proof of the Linear Affine Constraints

It is trivial that an affine transformation $\mathbf{A}$ transforms the direction of the corresponding epipolar lines to each other as all affine transformations correctly modify the lines going through the corresponding point locations $[u \quad v]$ and $[u' \quad v']$. Therefore, $\mathbf{Av} \parallel \mathbf{v}'$, where $\mathbf{v}$ and $\mathbf{v}'$ are the directions of the epipolar lines on the first and second images.

As it is well-known in computer graphics [99], line normals are transformed as $\mathbf{A}^{-\mathrm{T}}\mathbf{n} = \beta\mathbf{n}'$, where $\mathbf{n} = (\mathbf{F}^{\mathrm{T}}\mathbf{p}')_{1:2}$ and $\mathbf{n}' = (\mathbf{Fp})_{1:2}$ are the normals of the epipolar lines ($\beta \neq 0$). Lower index $(1:2)$ denotes the first two elements of a vector. We prove here that

$$\mathbf{A}^{-\mathrm{T}}\mathbf{n} = -\mathbf{n}'. \tag{A.1}$$

Suppose that corresponding point pair $\mathbf{p} = [u \quad v \quad 1]^{\mathrm{T}}$ and $\mathbf{p}' = [u' \quad v' \quad 1]^{\mathrm{T}}$ are given. Let $\mathbf{n} = [n_u \quad n_v]^{\mathrm{T}}$ and $\mathbf{n}' = [n'_u \quad n'_v]^{\mathrm{T}}$ be the normal directions of epipolar lines

$$\mathbf{l}_1 = \mathbf{F}^{\mathrm{T}}\mathbf{p}' = [l_{1,a} \quad l_{1,b} \quad l_{1,c}]^{\mathrm{T}}, \tag{A.2}$$

and

$$\mathbf{l}'_1 = \mathbf{Fp} = [l'_{1,a} \quad l'_{1,b} \quad l'_{1,c}]^{\mathrm{T}}, \tag{A.3}$$

respectively. It is trivial that $\mathbf{A}^{-\mathrm{T}}\mathbf{n} = \beta\mathbf{n}'$ due to $\mathbf{Av} \parallel \mathbf{v}'$, where β is a scale factor. First, it is shown how affine transformation $\mathbf{A}$ transforms the length of $\mathbf{n}$ if it is a unit vector. To calculate this scale factor β, it is required to introduce a new point as close to $\mathbf{p}$ as possible determining epipolar lines on both images and β as the ratio of distances from these new lines. Let us introduce point $\mathbf{q} = \mathbf{p} + \delta \begin{bmatrix}\mathbf{n}^T & 0\end{bmatrix}^T$, where δ is a small scalar value. Point $\mathbf{q}$ determines an epipolar line $\mathbf{l}'_2 = [l'_{2,a} \quad l'_{2,b} \quad l'_{2,c}]^{\mathrm{T}}$ on the second image as

$$\mathbf{l}'_2 = \mathbf{Fq} = \mathbf{F}\left(\mathbf{p} + \delta \begin{bmatrix}\mathbf{n}^{\mathrm{T}} & 0\end{bmatrix}^{\mathrm{T}}\right) = [s_1 \quad s_2 \quad s_3]^{\mathrm{T}},$$

where

$$\begin{aligned} s_1 &= l'_{1,a} + \delta f_{11} n_u + \delta f_{12} n_v, \\ s_2 &= l'_{1,b} + \delta f_{21} n_u + \delta f_{22} n_v, \\ s_3 &= l'_{1,c} + \delta f_{31} n_u + \delta f_{32} n_v. \end{aligned}$$

Then scale β is given by the distance d' between line $\mathbf{l}'_2$ and point $\mathbf{p}'$. The setup is visualized in Fig. 4.1(b). The calculation of distance d' is given by the well-known

formula as follows:

$$d' = \frac{|s_1 u' + s_2 v' + s_3|}{\sqrt{s_1^2 + s_2^2}}. \tag{A.4}$$

It is known that point $\mathbf{p}'$ lies on $\mathbf{l}'_1$, which can be written as $l'_{1,a}u' + l'_{1,b}v' + l'_{1,c} = 0$. This fact reduces Eq. A.4 to

$$\begin{aligned} d' &= \frac{|\hat{s}_1 u' + \hat{s}_2 v^2 + \hat{s}_3|}{\sqrt{s_1^2 + s_2^2}}, \\ \hat{s}_1 &= \delta f_{11} n_u + \delta f_{12} n_v, \\ \hat{s}_2 &= \delta f_{21} n_u + \delta f_{22} n_v, \\ \hat{s}_3 &= \delta f_{31} n_u + \delta f_{32} n_v. \end{aligned} \tag{A.5}$$

To determine β, the introduced point $\mathbf{q}$ has to be moved infinitesimally close to the location of $\mathbf{p}$. In other words, $\delta \to 0$. β is the ratio of the length of vector $(\mathbf{p} - \mathbf{q})$ and the distance between point $\mathbf{p}'$ and line $\mathbf{l}'_2$. The latter is δ, while the former has just calculated in Eq. A.5. Therefore the square of β is written as

$$\beta^2 = \lim_{\delta \to 0} \frac{\delta^2}{d'^2} = \lim_{\delta \to 0} \frac{s_1^2 + s_2^2}{|\hat{s}_1 u' + \hat{s}_2 v' + \hat{s}_3|^2}. \tag{A.6}$$

After elementary modifications, the final formula for scale β is given as

$$\begin{aligned} \beta &= \frac{\sqrt{l'_{1,a} l'_{1,a} + l'_{1,b} l'_{1,b}}}{|\widetilde{s}_1 u' + \widetilde{s}_2 v' + \widetilde{s}_3|}, \\ \widetilde{s}_i &= f_{i1} n_u + f_{i2} n_v, \quad i \in \{1, 2, 3\}. \end{aligned} \tag{A.7}$$

The epipolar line corresponding to point $\mathbf{p}$ is parameterized as $\begin{bmatrix} l'_{1,a} & l'_{1,b} & l'_{1,c} \end{bmatrix} = \mathbf{F}[u \quad v \quad 1]^{\mathrm{T}}$. Therefore, the normal of the line is as $\mathbf{n}' = \begin{bmatrix} l'_{1,a} & l'_{1,b} \end{bmatrix}^{\mathrm{T}} = (\mathbf{F} \begin{bmatrix} u' & v' & 1 \end{bmatrix}^{\mathrm{T}})_{(1:2)}$. Similarly, $\mathbf{n} = (\mathbf{F}^{\mathrm{T}} \begin{bmatrix} u' & v' & 1 \end{bmatrix}^{\mathrm{T}})_{(1:2)}$. The numerator in Eq. A.7 can be rewritten as $|\mathbf{n}| = \sqrt{l_{1,a}^2 + l_{1,b}^2}$, while the denominator is as follows:

$$\begin{aligned} \widetilde{s}_1 u' + \widetilde{s}_2 v' + \widetilde{s}_3 = n_u (f_{11} u' + f_{21} v' + f_{31}) + \\ n_v (f_{12} u' + f_{22} v' + f_{32}) = n_u^2 + n_v^2 = |\mathbf{n}|^2. \end{aligned}$$

Thus

$$\beta = \pm \frac{|\mathbf{n}|}{|\mathbf{n}|^2} = \pm \frac{1}{|\mathbf{n}|}.$$

The length of normal $\mathbf{n}$ is one, thus $\beta = 1$, and Eq. 4.2.2 is modified as $\mathbf{A}^{-\mathrm{T}}\mathbf{n} = \pm\mathbf{n}'$. Since the direction of the epipolar lines on the two images must be the opposite of each other, the positive solution can be omitted. The final formula is as follows: $\mathbf{A}^{-\mathrm{T}}\mathbf{n} = -\mathbf{n}'$.

Appendix B

Surface Normals and General Camera Model

Given the projections $\mathbf{p}_1 = [x_1 \quad y_1]$ and $\mathbf{p}_2 = [x_2 \quad y_2]$ of a 3D surface point $[X \quad Y \quad Z]^{\mathrm{T}}$ and the local affine transformation $\mathbf{A}$ mapping the infinitesimally neighborhoods from the first to the second images and the intrinsic parameters of both cameras $\mathbf{K}_1$ and $\mathbf{K}_2$. The goal is to show how the related surface normal $\mathbf{n}$ can be estimated (see Figure B.1). The coordinates of a projection is calculated using the projections

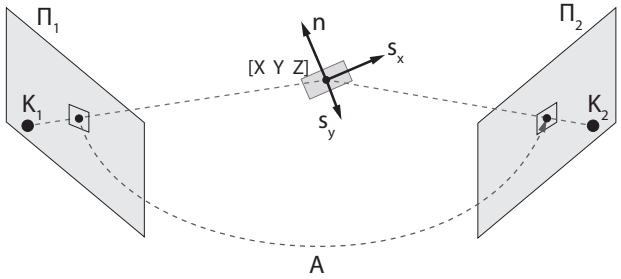


FIGURE B.1: 3D patch perspectively projected to stereo images.

functions Π_x and Π_y as follows:

$$x = \Pi_x(X, Y, Z), \qquad y = \Pi_y(X, Y, Z).$$

The surface point $[X \quad Y \quad Z]^{\mathrm{T}}$ is written in parametric form

$$X = X(u, v), \qquad Y = Y(u, v), \qquad Z = Z(u, v).$$

As it is well-known in differential geometry [189], the tangent vectors of the plane are written by the partial derivatives of the spatial coordinates, while the surface normal is given as the cross product of the tangent vectors: $\mathbf{n} = \mathbf{s}_u \times \mathbf{s}_v$, where

$$\mathbf{s}_u = \left[\begin{array}{ccc} \frac{\partial X(u,v)}{\partial u} & \frac{\partial Y(u,v)}{\partial u} & \frac{\partial Z(u,v)}{\partial u} \end{array} \right], \quad \mathbf{s}_v = \left[\begin{array}{ccc} \frac{\partial X(u,v)}{\partial v} & \frac{\partial Y(u,v)}{\partial v} & \frac{\partial Z(u,v)}{\partial v} \end{array} \right].$$

Point $[X \quad Y \quad Z]^{\mathrm{T}}$, and tangent vectors $\mathbf{s}_u$ and $\mathbf{s}_v$ determine the tangent plane which approximates the surface locally. Assuming a continuous surface, its points close to $[X \quad Y \quad Z]^{\mathrm{T}}$ are approximated by the first order Taylor-series:

$$\left[\begin{array}{c} x + \Delta x \\ y + \Delta y \end{array} \right] \approx \left[\begin{array}{c} \Pi_x(X, Y, Z) \\ \Pi_y(X, Y, Z) \end{array} \right] + \left[\begin{array}{cc} \frac{\partial \Pi_x(X,Y,Z)}{\partial u} & \frac{\partial \Pi_x(X,Y,Z)}{\partial v} \\ \frac{\partial \Pi_y(X,Y,Z)}{\partial u} & \frac{\partial \Pi_y(X,Y,Z)}{\partial v} \end{array} \right] \left[\begin{array}{c} \Delta u \\ \Delta v \end{array} \right].$$

Let us see that the partial derivatives of the projection functions give the affine transformation between 3D and 2D surface patches as follows:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \approx \mathbf{A} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \frac{\partial \Pi_x(X,Y,Z)}{\partial u} & \frac{\partial \Pi_x(X,Y,Z)}{\partial v} \\ \frac{\partial \Pi_y(X,Y,Z)}{\partial u} & \frac{\partial \Pi_y(X,Y,Z)}{\partial v} \end{bmatrix}.$$

The partial derivatives can be reformulated using the chain rule. For instance,

$$\begin{aligned} \frac{\partial \Pi_x(X,Y,Z)}{\partial u} = \frac{\partial \Pi_x(X,Y,Z)}{\partial u}\frac{X}{\partial u} + \frac{\partial \Pi_x(X,Y,Z)}{\partial v}\frac{Y}{\partial u} \\ + \frac{\partial \Pi_x(X,Y,Z)}{\partial Z}\frac{Z}{\partial u} = \nabla \Pi_x^{\mathrm{T}} \mathbf{s}_u, \end{aligned}$$

where $\nabla \Pi_x$ is the gradient vector of the projection function w.r.t. spatial coordinates X, Y and Z of the surface patch. Similarly,

$$\frac{\partial \Pi_x}{\partial v} = \nabla \Pi_x^{\mathrm{T}} \mathbf{s}_v \qquad \frac{\partial \Pi_y}{\partial u} = \nabla \Pi_y^{\mathrm{T}} \mathbf{s}_u \qquad \frac{\partial \Pi_y}{\partial v} = \nabla \Pi_y^{\mathrm{T}} \mathbf{s}_v.$$

Therefore, the affine matrix is written as

$$\mathbf{A} = \begin{bmatrix} \nabla \Pi_x^{\mathrm{T}} \\ \nabla \Pi_y^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{s}_u & \mathbf{s}_v \end{bmatrix}.$$

Considering the case when two images are given, the affine transformation between the image patches is obtained by multiplying the inverse of affine transformation $\mathbf{A}_1$ (between the patch of the 1st image and the 3D one), and the affine transformation $\mathbf{A}_2$ (between 3D patch and that in the 2nd image). Formally, it can be written as

$$\begin{bmatrix} \Delta x_2 & \Delta y_2 \end{bmatrix}^{\mathrm{T}} = \mathbf{A}_2 \mathbf{A}_1^{-1} \begin{bmatrix} \Delta x_1 & \Delta y_1 \end{bmatrix}^{\mathrm{T}}.$$

$\mathbf{A}_2\mathbf{A}_1^{-1}$ is the cumulated affine transformation between the images. The inverse of the affine matrix $\mathbf{A}$ can be written as

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \begin{bmatrix} \Pi_x^{\mathrm{T}} \mathbf{s}_u & -\Pi_y^{\mathrm{T}} \mathbf{s}_u \\ -\Pi_x^{\mathrm{T}} \mathbf{s}_v & \Pi_y^{\mathrm{T}} \mathbf{s}_v \end{bmatrix},$$

where $\det(\mathbf{A}) = \Pi_x^{\mathrm{T}} \mathbf{s}_u \Pi_y^{\mathrm{T}} \mathbf{s}_v - \Pi_x^{\mathrm{T}} \mathbf{s}_v \Pi_y^{\mathrm{T}} \mathbf{s}_u$. Exploiting the fact that $\mathbf{s}_v \mathbf{s}_u^{\mathrm{T}} - \mathbf{s}_u \mathbf{s}_v^{\mathrm{T}} = [\mathbf{n}]_\times$, the transformation $\mathbf{A}_2\mathbf{A}_1^{-1}$ can be written as

$$\mathbf{A}_1^{-1}\mathbf{A}_2 = \frac{1}{{\Pi_x^1}^{\mathrm{T}} [\mathbf{n}]_\times \Pi_y^1} \begin{bmatrix} {\Pi_x^2}^{\mathrm{T}} [\mathbf{n}]_\times \Pi_y^1 & {\Pi_x^1}^{\mathrm{T}} [\mathbf{n}]_\times \Pi_x^2 \\ {\Pi_y^2}^{\mathrm{T}} [\mathbf{n}]_\times \Pi_y^1 & {\Pi_x^1}^{\mathrm{T}} [\mathbf{n}]_\times \Pi_y^2 \end{bmatrix}.$$

Note that the scale of the normal is arbitrary since both the determinant and the matrix elements are multiplied by $[\mathbf{n}]_\times$. The expression $\mathbf{a}^{\mathrm{T}} [\mathbf{n}]_\times \mathbf{b}$ is also called the scalar triple product. Remark that $\mathbf{a}^{\mathrm{T}} [\mathbf{n}]_\times \mathbf{b}$ equals to $\mathbf{n}^{\mathrm{T}}(\mathbf{b} \times \mathbf{a})$. Therefore, the final equation of the affine transformation is written as

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \mathbf{A}_1^{-1}\mathbf{A}_2 = \frac{1}{\mathbf{n}^{\mathrm{T}}\mathbf{w}_5} \begin{bmatrix} \mathbf{n}^{\mathrm{T}}\mathbf{w}_1 & \mathbf{n}^{\mathrm{T}}\mathbf{w}_2 \\ \mathbf{n}^{\mathrm{T}}\mathbf{w}_3 & \mathbf{n}^{\mathrm{T}}\mathbf{w}_4 \end{bmatrix}, \tag{B.1}$$

where $\mathbf{w}_1 = \nabla \Pi_y^1 \times \nabla \Pi_x^2$, $\mathbf{w}_2 = \nabla \Pi_x^2 \times \nabla \Pi_x^1$, $\mathbf{w}_3 = \nabla \Pi_y^1 \times \nabla \Pi_y^2$, $\mathbf{w}_4 = \nabla \Pi_y^2 \times \nabla \Pi_x^1$, and $\mathbf{w}_5 = \nabla \Pi_y^1 \times \nabla \Pi_x^1$. Equation B.1 shows the relationship of surface normals and local affinities for arbitrary camera model.

Appendix C

Affine Parameters from a Homography

Affine transformation $\mathbf{A}$ comes from the first-order Taylor serie of homography $\mathbf{H}$, where $\mathbf{H}$ is a plane-plane perspective transformation between stereo images. The correspondence between the coordinates in the first (x_1 and y_1) and second (x_2 and y_2) images is

$$x_2 = \frac{\mathbf{h}_1^{\mathrm{T}} [x_1 \quad y_1 \quad 1]^{\mathrm{T}}}{\mathbf{h}_3^{\mathrm{T}} [x_1 \quad y_1 \quad 1]^{\mathrm{T}}}, \quad y_2 = \frac{\mathbf{h}_2^{\mathrm{T}} [x_1 \quad y_1 \quad 1]^{\mathrm{T}}}{\mathbf{h}_3^{\mathrm{T}} [x_1 \quad y_1 \quad 1]^{\mathrm{T}}},$$

where 3×3 homography matrix $\mathbf{H}$ is written as

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^{\mathrm{T}} \\ \mathbf{h}_2^{\mathrm{T}} \\ \mathbf{h}_3^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}.$$

The affine parameters equal to the partial derivatives of the homography. For example, the top left element a_{11} of the affine transformation is as follows:

$$a_{11} = \frac{\partial x_2}{\partial x_1} = \frac{h_{11}\mathbf{h}_3^{\mathrm{T}} [x_1 \quad y_1 \quad 1]^{\mathrm{T}} - h_{31}\mathbf{h}_1^{\mathrm{T}} [x_1 \quad y_1 \quad 1]^{\mathrm{T}}}{\left(\mathbf{h}_3^{\mathrm{T}} [x_1 \quad y_1 \quad 1]^{\mathrm{T}}\right)^2} = \frac{h_{11} - h_{31}x_2}{s},$$

where $s = \mathbf{h}_3^{\mathrm{T}}[x_1 \quad y_1 \quad 1]^{\mathrm{T}}$ is called the projective depth of the point. The other components are obtained in the same way:

$$a_{11} = \frac{\partial x_1}{\partial y_1} = \frac{h_{11} - h_{31}x_2}{s}, \quad a_{12} = \frac{\partial x_2}{\partial y_1} = \frac{h_{12} - h_{32}x_2}{s},$$
$$a_{21} = \frac{\partial y_2}{\partial x_1} = \frac{h_{21} - h_{31}y_2}{s}, \quad a_{22} = \frac{\partial y_2}{\partial y_1} = \frac{h_{22} - h_{32}y_2}{s}.$$

Appendix D

Is LSQ Minimization of the Affine Parameters Correct

It is shown in this section that the minimization of the Frobenious-norm has both algebraic and geometric interpretations for local affine transformations.

Matrix $\mathbf{A}$ without the translation is a 2×2 linear transformation, therefore, it is determined by two points. (The projection of the origin remains the same.) Let us choose points $\begin{bmatrix}1 & 0\end{bmatrix}^{\mathrm{T}}$ and $\begin{bmatrix}0 & 1\end{bmatrix}^{\mathrm{T}}$. Then the minimizing formula for the former one is as follows:

$$\begin{aligned}\left\|\mathbf{A}\begin{bmatrix}1\\0\end{bmatrix}-\mathbf{A}'\begin{bmatrix}1\\0\end{bmatrix}\right\|_2^2=\left\|(\mathbf{A}-\mathbf{A}')\begin{bmatrix}1\\0\end{bmatrix}\right\|_2^2=\\ \left\|\begin{bmatrix}a_{11}-a'_{11} & a_{12}-a'_{12}\\ a_{21}-a'_{21} & a_{22}-a'_{22}\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\right\|_2^2=\left\|\begin{bmatrix}a_{11}-a'_{11}\\ a_{21}-a'_{21}\end{bmatrix}\right\|_2^2=\\ (a_{11}-a'_{11})^2+(a_{21}-a'_{21})^2=0.\end{aligned} \tag{D.1}$$

The minimization for the second point is fairly similar as

$$\left\|\mathbf{A}\begin{bmatrix}0\\1\end{bmatrix}-\mathbf{A}'\begin{bmatrix}0\\1\end{bmatrix}\right\|_2^2=\left\|\begin{bmatrix}a_{12}-a'_{12}\\ a_{22}-a'_{22}\end{bmatrix}\right\|_2^2=(a_{12}-a'_{12})^2+(a_{22}-a'_{22})^2=0. \tag{D.2}$$

By combining both Eqs. D.1, D.2 the Frobenious-norm of difference matrix $\mathbf{A}-\mathbf{A}'$ is obtained. As a consequence, *minimizing the Frobenious-norm of the difference matrix is equivalent to the optimization of its effect on points.* Therefore, the squared differences of the parameters have both algebraic and geometric interpretations.

Summary

In English

In this thesis, we focused on two major fields of computer vision, that are geometric model estimation from affine correspondences and robust (multi-)model estimation. Some of the papers which inspired this work were published at the most prestigious computer vision forums, such as, Conference on Computer Vision and Pattern Recognition (CVPR) or European Conference on Computer Vision (ECCV).

Solving problems in geometric model estimation, we proposed methods for estimating surface normals from multiple views; homographies from a single affine correspondence; fundamental and essential matrices from three and two correspondences; and solved the semi-calibrated case as well, i.e. when the intrinsic camera parameters are given but a common focal length. Moreover, we showed what is the direct relationship between affine correspondences and epipolar geometry. The proposed methods were proven to be accurate both in our synthesized test environment and on publicly available real world data and they are compared with the state-of-the-art algorithms of the field.

Approaching robust model estimation, we proposed a method for rejecting outliers from a set of point correspondences without assuming an underlying geometric model which interprets the scene; a locally optimized RANSAC was proposed outperforming the state-of-the-art on various robust model fitting problems. The method is built on the assumption that close points are more likely belong to the same model and, thus, it is beneficial to take the spatial coherence into consideration. Also, a method is proposed for fitting multiple homographies in two views. Generalizing this problem, we proposed an approach for estimating multiple geometric models which are not necessarily of the same model class. The performance of these methods were also evaluated both in our synthesized test environment and on publicly available real world data.

In Hungarian

Ezen disszertációban a számítógépes látás két területére fókuszáltunk, amik a geometriai modell becslés affin megfeleltetések felhasználásával, illetve a robusztus (multi-)modell illesztés. Néhány, e munkát ihlető cikk a számógépes látás legnagyobb presztízsű fórumain jelent meg, mint például a Conference on Computer Vision and Pattern Recognition (CVPR), vagy a European Conference on Computer Vision (ECCV).

A geometriai modell becslés számos problémáját oldottuk meg affin megfeleltetések felhasználásával. Ezek a problémák: felületi normálisok becslése több nézet felhasználásával; homográfia becslése egyetlen megfeleltetésből; fundamentális és esszenciális mátrixok becslése három, illetve két megfeleltetésből; és megoldottuk az úgy nevezett félig-kalibrált esetet is affin jellemzőpontokkal. A félig-kalibrált eset alapfeltevése, hogy a kamerák belső paraméterei egy közös fókusztáv kivételével mind ismertek. Ezen felül megmutattuk, hogy mi a közvetlen kapcsolat affin megfeleltetések és az epipoláris geometria között. A javasolt algoritmusokat mind szintetikus tesztkörnyezetben, mind publikusan elérhető valós adatbázisokon teszteltük és hasonlítottuk össze a state-of-the-arttal.

A robusztus modell illesztés területén javasoltunk egy módszer outlierek szűrésére pont-megfeleltetések egy halmazából. A javasolt módszer nem feltételezi egy, a színteret magyarázó geometriai modell létezését. Javasoltunk egy új lokálisan optimalizált RANSAC algoritmust is, mely arra a feltételezésre épít, hogy az egymáshoz közel elhelyezkedő pontok nagy valószínűséggel tartoznak ugyanahhoz a modellhez. Tehát a pontok térbeli relációira építve nagyobb pontosság és korábbi terminá-ció érhető el. Javaslunk egy módszert multi-homográfia illesztésre is két kép között, majd ezt a módszert általánosítva eljutunk ahhoz a problémához, amikor ismeretlen számú és tetszőleges típusú modellt szeretnénk egyidőben megtalálni. Ezen módszerek hatékonyságát is kiértékeltük mind szintetikus, mind publikusan elérhető valós adatbázisokon.

Bibliography

[1] 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*, vol. 24, no. 6, pp. 381–395, 1981.

[2] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison", *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[3] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector", in *European Conference on Computer Vision*, Springer, 2002, pp. 128–142.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] K. Köser, *Geometric Estimation with Local Affine Frames and Free-form Surfaces*. Shaker, 2009.

[6] M. Perdoch, J. Matas, and O. Chum, "Epipolar geometry from two correspondences", in *International Conference on Pattern Recognition*, 2006, pp. 215–219.

[7] O. Chum, J. Matas, and S. Obdrzálek, "Epipolar geometry from three correspondences", *Computer Vision Winter Workshop*, 2003.

[8] J. Bentolila and J. M. Francos, "Conic epipolar constraints from affine correspondences", *Computer Vision and Image Understanding*, 2014.

[9] C. Raposo and J. P. Barreto, "Theory and practice of structure-from-motion using affine correspondences", in *Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5470–5478.

[10] O. Chum, J. Matas, and J. Kittler, "Locally optimized RANSAC", in *Joint Pattern Recognition Symposium*, Springer, 2003, pp. 236–243.

[11] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, "USAC: A universal framework for random sample consensus", *Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 2022–2038, 2013.

[12] P. V. C. Hough, *Method and means for recognizing complex patterns*, 1962.

[13] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting", *International Journal of Computer Vision*, vol. 97, no. 2, pp. 123–147, 2012.

[14] L. Magri and A. Fusiello, "Multiple model fitting as a set coverage problem", in *Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3318–3326.

[15] T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter, "Interacting geometric priors for robust multimodel fitting", *Transactions on Image Processing*, vol. 23, no. 10, pp. 4601–4610, 2014.

[16] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision", *Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[17] W. Zhang and J. Kŏsecká, "Nonparametric estimation of multiple structures with outliers", in *Dynamical Vision*, 2007.

[18] M. Stricker and A. Leonardis, "ExSel++: A general framework to extract parametric models", in *International Conference on Computer Analysis of Images and Patterns*, 1995.

[19] D. Barath, "Efficient energy-based topological outlier rejection", 2018.

[20] D. Barath and L. Hajder, "Efficient recovery of essential matrix from two affine correspondences", 2018.

[21] D. Baráth and L. Hajder, "A theory of point-wise homography estimation", *Pattern Recognition Letters*, vol. 94, no. Supplement C, pp. 7–14, 2017.

[22] D. Barath and J. Matas, "Multi-class model fitting by energy minimization and mode-seeking", 2018.

[23] D. Baráth and J. Matas, "Graph-cut RANSAC", *Conference on Computer Vision and Pattern Recognition*, 2018.

[24] D. Barath, "Five-point fundamental matrix estimation for uncalibrated cameras", *Conference on Computer Vision and Pattern Recognition*, 2018.

[25] D. Barath, T. Toth, and L. Hajder, "A minimal solution for two-view focal-length estimation using two affine correspondences", in *Conference on Computer Vision and Pattern Recognition*, 2017.

[26] D. Barath, "P-HAF: Homography estimation using partial local affine frames", in *International Joint Conference of Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2017, pp. 227–235.

[27] D. Baráth, J. Matas, and L. Hajder, "Multi-H: Efficient recovery of tangent planes in stereo images", in *British Machine Vision Conference*, vol. 28, 2016, p. 32.

[28] D. Barath, J. Matas, and L. Hajder, "Accurate closed-form estimation of local affine transformations consistent with the epipolar geometry", in *British Machine Vision Conference*, 2016.

[29] D. Barath and L. Hajder, "Energy-based topological outlier filtering", in *International Conference on Pattern Recognition*, IEEE, 2016, pp. 1237–1242.

[30] D. Baráth and L. Hajder, "Novel ways to estimate homography from local affine transformations", in *Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2016, pp. 432–443.

[31] D. Barath and I. Eichhardt, "A novel technique for point-wise surface normal estimation", 2016.

[32] D. Barath, J. Molnar, and L. Hajder, "Novel methods for estimating surface normals from affine transformations", in *Computer Vision, Imaging and Computer Graphics Theory and Applications*, Springer International Publishing, 2016, pp. 316–337.

[33] J. Molnár, D. Csetverikov, Z. Kató, and D. Baráth, "A theory of camera-independent correspondence", 2015.

[34] D. Barath, J. Molnar, and L. Hajder, "Optimal surface normal from affine transformation", SciTePress, 2015, pp. 305–316.

[35] D. Barath and J. Matas, "MAGSAC: Marginalizing sample consensus", *Conference on Computer Vision and Pattern Recognition*, 2019.

[36] D. Barath, J. Matas, and L. Hajder, "Epipoláris geometriával konzisztens, legközelebbi affin transzformáció optimális becslése", in *Képfeldolgozók és Alakfelismerők Társaságának 11. konferenciája*, 2017.

[37] D. Baráth, J. Matas, and L. Hajder, "Multi-H: Érintősíkok hatékony kinyerése képpárokból", in *Képfeldolgozók és Alakfelismerők Társaságának 11. konferenciája*, 2017.

[38] I. Eichhardt and D. Barath, "Felületi normális becslése egyetlen pont-megfeleltetés alapján", in *Képfeldolgozók és Alakfelismerők Társaságának 11. konferenciája*, 2017.

[39] D. Barath and L. Hajder, "Homográfia becslése részlegesen ismert affin transzformációból", in *GRAFGEO*, 2016.

[40] D. Baráth and L. Hajder, "Normálvektorok optimális becslése affin transzformációkból", in *GRAFGEO*, 2015.

[41] D. Barath, "Homográfiabecslés affin transzformációból", in *Képfeldolgozók és Alakfelismerők Társaságának 10. konferenciája*, 2015.

[42] D. Barath, J. Molnar, and L. Hajder, "Novel methods for estimating surface normals from affine transformations", in *International Joint Conference of Computer Vision, Imaging and Computer Graphics Theory and Applications, Revised Selected Papers*. Springer International Publishing, 2016, pp. 316–337.

[43] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[44] J. Molnár, R. Huang, and Z. Kato, "3d reconstruction of planar surface patches: A direct solution", *Asian Conference on Computer Vision – Big Data in 3D Vision Workshop*, 2014.

[45] J. Matas, O. Chum, M. Urban, and P. T., "Robust wide baseline stereo from maximally stable extremal regions", in *British Machine Vision Conference*, 2002.

[46] D. Mishkin, J. Matas, and M. Perdoch, "MODS: Fast and robust method for two-view matching", *Computer Vision and Image Understanding*, vol. 141, pp. 81–93, 2015.

[47] R. I. Hartley, "In defense of the eight-point algorithm", *Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[48] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction", in *Eurographics Symposium on Geometry processing*, Eurographics Association, 2006, pp. 61–70.

[49] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction", *Transactions on Graphics*, vol. 32, no. 3, p. 29, 2013.

[50] R. J. Woodham, "Photometric method for determining surface orientation from multiple images", *Optical engineering*, vol. 19, no. 1, pp. 139–144, 1980.

[51] F. E. Nicodemus, "Directional reflectance and emissivity of an opaque surface", *AO*, vol. 4, no. 7, pp. 767–775, 1965.

[52] C. H. Esteban, G. Vogiatzis, and R. Cipolla, "Multiview photometric stereo", *Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 548–554, 2008.

[53] Y. Quéau, R. Mecca, J.-D. Durou, and X. Descombes, "Photometric stereo with only two images: A theoretical study and numerical resolution", *Image and Vision Computing*, vol. 57, pp. 175–191, 2017.

[54] O. D. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, no. 03, pp. 485–508, 1988.

[55] E. Malis and M. Vargas, "Deeper understanding of the homography decomposition for vision-based control", PhD thesis, INRIA, 2007.

[56] H. Liu, "Deeper understanding on solution ambiguity in estimating 3d motion parameters by homography decompositionand its improvement", 2012.

[57] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors", *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[58] Y. Furukawa and J. Ponce, "Accurate camera calibration from multi-view stereo and bundle adjustment", in *Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.

[59] Y. Furukawa and J. Ponce, *Patch-based multi-view stereo software*, 2010. [Online]. Available: http://www.di.ens.fr/pmvs.

[60] P. F. Georgel, S. Benhimane, and N. Navab, "A unified approach combining photometric and geometric information for pose estimation.", in *British Machine Vision Conference*, 2008, pp. 1–10.

[61] M. Clegg, J. Edmonds, and R. Impagliazzo, "Using the groebner basis algorithm to find proofs of unsatisfiability", in *Symposium on Theory of computing*, ACM, 1996, pp. 174–183.

[62] B. Åke, *Numerical methods for least squares problems*. SIAM, 1996.

[63] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery", in *Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[64] Z. Pusztai and L. Hajder, "Ground-truth tracking data generation using rotating real-world objects", in *Computer Vision, Imaging and Computer Graphics Theory and Applications - Revised Selected Papers*, Springer, 2017, pp. 395–417.

[65] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, P. M., and A. S. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos", in *Conference on Computer Vision and Pattern Recognition*, 2017.

[66] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis", *Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[67] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "OpenMVG: Open multiple view geometry", in *International Workshop on Reproducible Research in Pattern Recognition*, Springer, 2016, pp. 60–74.

[68] Z. Zhang and A. R. Hanson, "3d reconstruction based on homography mapping", *Proc. ARPA96*, pp. 1007–1012, 1996.

[69] T. Werner and A. Zisserman, "New techniques for automated architectural reconstruction from photographs", in *European Conference on Computer Vision*, Springer, 2002, pp. 541–555.

[70] Z. Z., "A flexible new technique for camera calibration", *Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[71] Z. Chuan, T. D. Long, Z. Feng, and D. Z. Li, "A planar homography estimation method for camera calibration", in *International Symposium on Computational Intelligence in Robotics and Automation*, IEEE, vol. 1, 2003, pp. 424–429.

[72] T. Ueshiba and F. Tomita, "Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices", in *International Conference on Computer Vision*, IEEE, 2003, pp. 966–973.

[73] S. J. D. Prince, K. Xu, and A. D. Cheok, "Augmented reality camera tracking with homographies", *Computer Graphics and Applications*, vol. 22, no. 6, pp. 39–45, 2002.

[74] J. Zhou and B. Li, "Robust ground plane detection with normalized homography in monocular sequences from a robot platform", in *International Conference on Image Processing*, IEEE, 2006, pp. 3017–3020.

[75] S. Mann, J. Huang, R. Janzen, R. Lo, V. Rampersad, A. Chen, and T. Doha, "Blind navigation with a wearable range camera and vibrotactile helmet", in *International Conference on Multimedia*, ACM, 2011, pp. 1325–1328.

[76] A. Agarwal, C. V. Jawahar, and P. J. Narayanan, "A survey of planar homography estimation techniques", *Centre for Visual Information Technology*, 2005.

[77] J. Nemeth, C. Domokos, and Z. Kato, "Recovering planar homographies between 2d shapes", in *International Conference on Computer Vision*, IEEE, 2009, pp. 2170–2176.

[78] P. K. Jain and C. V. Jawahar, "Homography estimation from planar contours", in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, IEEE, 2006, pp. 877–884.

[79] P. K. Mudigonda, C. V. Jawahar, and P. J. Narayanan, "Geometric structure computation from conics", in *Indian Conference on Computer Vision, Graphics and Image Processing*, 2004, pp. 9–14.

[80] A. Sugimoto, "A linear algorithm for computing the homography from conics in correspondence", *Journal of Mathematical Imaging and Vision*, vol. 13, no. 2, pp. 115–130, 2000.

[81] S. Choi, T. Kim, and W. Yu, "Performance evaluation of RANSAC family", *Journal of Computer Vision*, vol. 24, no. 3, pp. 271–300, 1997.

[82] P. J. Rousseeuw, "Least median of squares regression", *Journal of the American statistical association*, vol. 79, no. 388, pp. 871–880, 1984.

[83] R. Toldo and A. Fusiello, "Robust multiple structures estimation with J-linkage", in *European Conference on Computer Vision*, Springer, 2008, pp. 537–547.

[84] L. Magri and A. Fusiello, "T-linkage: A continuous relaxation of J-linkage for multi-model fitting", in *Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3954–3961.

[85] J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory", in *Numerical analysis*, Springer, 1978, pp. 105–116.

[86] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions", *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[87] J. Chen, W. E. Dixon, D. M. Dawson, and M. McIntyre, "Homography-based visual servo tracking control of a wheeled mobile robot", *Transactions on Robotics*, vol. 22, no. 2, pp. 406–415, 2006.

[88] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[89] A. Tanacs, A. Majdik, J. Molnar, A. Rai, and Z. Kato, "Establishing correspondences between planar image patches", in *Digital lmage Computing: Techniques and Applications*, IEEE, 2014, pp. 1–7.

[90] O. Chum and J. Matas, "Homography estimation from correspondences of local elliptical features", in *International Conference on Pattern Recognition*, IEEE, 2012, pp. 3236–3239.

[91] J. Matas, S. Obdrzálek, and O. Chum, "Local affine frames for wide-baseline stereo", in *International Conference on Pattern Recognition*, 2002, pp. 363–366.

[92] K. Köser and R. Koch, "Differential spatial resection - pose estimation using a single local image feature", in *European Conference on Computer Vision*, 2008, pp. 312–325.

[93] A. Bódis-Szomorú, H. Riemenschneider, and L. Van Gool, "Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels", in *Conference on Computer Vision and Pattern Recognition*, 2014.

[94] J. Bentolila and J. M. Francos, "Conic epipolar constraints from affine correspondences", *Computer Vision and Image Understanding*, vol. 122, pp. 105–114, 2014.

[95] D. G. Lowe, "Object recognition from local scale-invariant features", in *Proceedings of the International Conference on Computer Vision*, ser. International Conference on Computer Vision, 1999, pp. 1150–1157.

[96] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features", in *European Conference on Computer Vision*, Springer, 2006, pp. 404–417.

[97] P. Courrieu, "Fast computation of moore-penrose inverse matrices", *arXiv preprint arXiv:0804.4809*, 2008.

[98] H. S. Wong, T.-J. Chin, J. Yu, and D. Suter, "Dynamic and hierarchical multi-structure geometric model fitting", in *International Conference on Computer Vision*, 2011.

[99] K. Turkowski, "Transformations of surface normal vectors", in *Tech. Rep. 22, Apple Computer*, 1990.

[100] R. I. Hartley and P. Sturm, "Triangulation", *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[101] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors", *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[102] A. Baumberg, "Reliable feature matching across widely separated views", in *Computer Vision and Pattern Recognition*, IEEE, vol. 1, 2000, pp. 774–781.

[103] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to sift or surf", in *International Conference on Computer Vision*, IEEE, 2011, pp. 2564–2571.

[104] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints", in *International Conference on Computer Vision*, 2011, pp. 2548–2555.

[105] Q.-T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis", *International Journal of Computer Vision*, 1996.

[106] H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky, "A minimal solution for relative pose with unknown focal length", *Image and Vision Computing*, 2008.

[107] H. Li, "A simple solution to the six-point two-view focal-length problem", in *European Conference on Computer Vision*, Springer, 2006.

[108] W. Wang and C. Wu, "Six-point synthetic method to estimate fundamental matrix", *Science in China Series E: Technological Sciences*, 1997.

[109] R. I. Hartley and H. Li, "An efficient hidden variable approach to minimal-case camera motion estimation", *Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[110] J. Philip, "A non-iterative algorithm for determining all essential matrices corresponding to five point pairs", *The Photogrammetric Record*, 1996.

[111] D. Nistér, "An efficient solution to the five-point relative pose problem", *Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[112] H. Li and R. I. Hartley, "Five-point motion estimation made easy", in *International Conference on Pattern Recognition*, IEEE, 2006.

[113] D. Batra, B. Nabbe, and M. Hebert, "An alternative formulation for five point relative pose problem", in *Workshop on Motion and Video Computing*, IEEE, 2007.

[114] M. Perd'och, J. Matas, and O. Chum, "Epipolar geometry from two correspondences", in *International Conference on Pattern Recognition*, IEEE.

[115] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide baseline stereo", *Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[116] O. Chum and J. Matas, "Matching with PROSAC-progressive sample consensus", in *Conference on Computer Vision and Pattern Recognition*, IEEE, 2005.

[117] V. Codreanu, F. Dong, B. Liu, J. B. Roerdink, D. Williams, P. Yang, and B. Yasar, "GPU-ASIFT: A fast fully affine-invariant feature extraction algorithm", in *International Conference on High Performance Computing and Simulation*, IEEE, 2013.

[118] A. Torii, Z. Kukelova, M. Bujnak, and T. Pajdla, "The six point algorithm revisited", in *Proceedings of the Asian Conference on Computer Vision*, 2010.

[119] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day", *Communications ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[120] F. Jan-Michael, F. G. Pierre, G. David, J. Tim, R. Rahul, W. Changchang, J. Yi-Hung, D. Enrique, C. Brian, and L. Svetlana, "Building rome on a cloudless day", in *European Conference on Computer Vision*, 2010, pp. 368–381.

[121] M. Pierre, M. Pascal, and M. Renaud, "Global fusion of relative motions for robust, accurate and scalable structure from motion", in *International Conference on Computer Vision*, 2013, pp. 3248–3255.

[122] D. A. Cox, J. Little, and D. O'shea, *Using algebraic geometry*. 2006.

[123] Z. Kukelova, M. Bujnak, and T. Pajdla, "Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems", in *British Machine Vision Conference*, 2008.

[124] Á. Pernek and L. Hajder, "Automatic focal length estimation as an eigenvalue problem", *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1108–1117, 2013.

[125] H. Li and R. Hartley, "A non-iterative method for correcting lens distortion from nine-point correspondences", in *International Conference on Computer Vision*, 2005.

[126] Z. Kukelova, T. Pajdla, and M. Bujnak, "Algebraic methods in computer vision", PhD thesis, Center for Machine Perception, Czech Technical University, Prague, Czech republic, 2012.

[127] K. J. Anil, N. M. Murty, and J. F. Patrick, "Data clustering: A review", *Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[128] L. Shapira, S. Avidan, and A. Shamir, "Mode-detection via median-shift", in *Proceedings of the International Conference on Computer Vision*, 2009.

[129] J. W. Tukey, "Mathematics and the picturing of data", *Proceedings of the International Congress of Mathematicians*, vol. 2, pp. 523–531, 1975.

[130] E. Weiszfeld, "Sur le point pour lequel la somme des distances de n points donnés est minimum", *Tohoku Mathematical Journal, First Series*, 1937.

[131] M. Perdoch, J. Matas, and O. Chum, "Epipolar geometry from two correspondences", in *International Conference on Pattern Recognition*, 2006.

[132] M. Bujnak, Z. Kukelova, and T. Pajdla, "Robust focal length estimation by voting in multi-view scene reconstruction", *Asian Conference on Computer Vision*, pp. 13–24, 2010.

[133] Y. Furukawa and J. Ponce, *PMVS*, 2007.

[134] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Clustering views for multi-view stereo", in *Conference on Computer Vision and Pattern Recognition*, vol. 13, 2010, e18.

[135] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes", *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[136] P. H. S. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix", *International Journal of Computer Vision*, vol. 24, no. 3, pp. 271–300, 1997.

[137] E. Vincent and R. Laganiére, "Detecting planar homographies in an image pair", in *International Symposium on Image and Signal Processing and Analysis*, 2001.

[138] Y. Kanazawa and H. Kawakami, "Detection of planar regions with uncalibrated stereo using distributions of feature points.", in *British Machine Vision Conference*, 2004.

[139] M. Zuliani, C. S. Kenney, and B. S. Manjunath, "The multiransac algorithm and its application to detect planar homographies", in *International Conference on Image Processing*, IEEE, vol. 3, 2005, pp. III–153.

[140] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts", *Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[141] L. Magri and A. Fusiello, "Robust multiple model fitting with preference analysis and low-rank approximation", 2016.

[142] P. Bhattacharya and M. Gavrilova, "DT-RANSAC: A delaunay triangulation based scheme for improved RANSAC feature matching", in *Transactions on Computational Science XX*, Springer, 2013, pp. 5–21.

[143] D.-T. Lee and B. J. Schachter, "Two algorithms for constructing a delaunay triangulation", *International Journal of Computer & Information Sciences*, vol. 9, no. 3, pp. 219–242, 1980.

[144] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts", in *Transactions on Graphics*, ACM, vol. 23, 2004, pp. 309–314.

[145] O. Chum, T. Werner, and J. Matas, "Two-view geometry estimation unaffected by a dominant plane", in *Conference on Computer Vision and Pattern Recognition*, IEEE, vol. 1, 2005, pp. 772–779.

[146] V. Kolmogorov and R. Zabin, "What energy functions can be minimized via graph cuts?", *Pattern Analysis and Machine Intelligence*, 2004.

[147] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration.", *Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 2, no. 331-340, p. 2, 2009.

[148] P. H. S. Torr and D. W. Murray, "Outlier detection and motion segmentation", in *Optical Tools for Manufacturing and Advanced Automation*, International Society for Optics and Photonics, 1993.

[149] P. H. S. Torr, A. Zisserman, and S. J. Maybank, "Robust detection of degenerate configurations while estimating the fundamental matrix", *Computer Vision and Image Understanding*, 1998.

[150] P. Pritchett and A. Zisserman, "Wide baseline stereo matching", in *International Conference on Computer Vision*, IEEE, 1998.

[151] D. Ghosh and N. Kaabouch, "A survey on image mosaicing techniques", *Journal of Visual Communication and Image Representation*, 2016.

[152] C. Sminchisescu, D. Metaxas, and S. Dickinson, "Incremental model-based estimation using geometric constraints", *Pattern Analysis and Machine Intelligence*, 2005.

[153] D. Nasuto and J. M. B. R. Craddock, "NAPSAC: High noise, high dimensional robust estimation-it's in the bag", 2002.

[154] V. Fragoso, P. Sen, S. Rodriguez, and M. Turk, "EVSAC: Accelerating hypotheses generation by modeling matching scores with extreme value theory", in *International Conference on Computer Vision*, 2013.

[155] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry", *Computer Vision and Image Understanding*, 2000.

[156] K. Lebeda, J. Matas, and O. Chum, "Fixing the locally optimized RANSAC – full experimental evaluation", in *British Machine Vision Conference*, Citeseer, 2012.

[157] Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations", in *Computer Vision and Pattern Recognition*, IEEE, 1998.

[158] R. Zabih and V. Kolmogorov, "Spatially coherent clustering using graph cuts", in *Conference on Computer Vision and Pattern Recognition*, IEEE, vol. 2, 2004, pp. II–II.

[159] H. Le, T.-J. Chin, and D. Suter, "An exact penalty method for locally convergent maximum consensus", in *Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2017.

[160] O. Chum, T. Werner, and J. Matas, "Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint", in *International Conference on Pattern Recognition*, 2004.

[161] C. Strecha, R. Fransens, and L. Van Gool, "Wide-baseline stereo from multiple views: A probabilistic account", in *Conference on Computer Vision and Pattern Recognition*, IEEE, vol. 1, 2004, pp. I–I.

[162] C. Benedek and T. Szirányi, "Change detection in optical aerial images by a multilayer conditional mixed markov model", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 10, pp. 3416–3430, 2009.

[163] G. Simon, A. W. Fitzgibbon, and A. Zisserman, "Markerless tracking using planar structures in the scene", in *International Symposium on Augmented Reality*, 2000, pp. 120–128.

[164] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis", *Transactions on Pattern Analysis and Machine Intelligence*, pp. 603–619, 2002.

[165] D. Barath, J. Matas, and L. Hajder, "Accurate closed-form estimation of local affine transformations consistent with the epipolar geometry", in *British Machine Vision Conference*, 2016.

[166] M. Marius and G. D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration", in *International Conference on Computer Vision Theory and Application*, 2009, pp. 331–340.

[167] T. T. Pham, T.-J. Chin, J. Yu, and D. Suter, "Simultaneous sampling and multi-structure fitting with adaptive reversible jump mcmc", in *Advances in Neural Information Processing Systems*, 2011, pp. 540–548.

[168] J. Yu, T.-J. Chin, and D. Suter, "A global optimization approach to robust multi-model fitting", in *Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2041–2048.

[169] N. Lazic, I. Givoni, B. Frey, and P. Aarabi, "Floss: Facility location for subspace segmentation", in *International Conference on Computer Vision*, 2009, pp. 825–832.

[170] T.-T. Pham, T.-J. Chin, J. Yu, and D. Suter, "The random cluster model for robust geometric fitting", *Pattern Analysis and Machine Intelligence*, pp. 1658–1671, 2014.

[171] V. Vineet and P. J. Narayanan, "Solving multilabel mrfs using incremental alpha-expansion on the gpus", in *Asian Conference on Computer Vision*, 2009, pp. 633–643.

[172] J. Illingworth and J. Kittler, "A survey of the hough transform", *Computer Vision, Graphics, and Image Processing*, 1988.

[173] N. Guil and E. L. Zapata, "Lower order circle and ellipse hough transform", *Pattern Recognition*, 1997.

[174] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic hough transform", *Computer Vision and Image Understanding*, 2000.

[175] P. L. Rosin, "Ellipse fitting by accumulating five-point fits", *Pattern Recognition Letters*, 1993.

[176] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized hough transform (rht)", *Pattern Recognition Letters*, 1990.

[177] L. Magri and A. Fusiello, "Robust multiple model fitting with preference analysis and low-rank approximation", in *British Machine Vision Conference*, 2015.

[178] A. Delong, L. Gorelick, O. Veksler, and Y. Boykov, "Minimizing energies with hierarchical costs", *International Journal of Computer Vision*, 2012.

[179] H. Wang, G. Xiao, Y. Yan, and D. Suter, "Mode-seeking on hypergraphs for robust geometric model fitting", in *International Conference on Computer Vision*, 2015.

[180] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs", *International Journal of Computer Vision*, 2012.

[181] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions", in *International Symposium on Computational Geometry*, 2004.

[182] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009.

[183] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: A texture classification example", in *International Conference on Computer Vision*, 2003.

[184] T.-J. Chin, H. Wang, and D. Suter, "Robust fitting of multiple structures: The statistical learning approach", in *International Conference on Computer Vision*, 2009.

[185] H. Liu and S. Yan, "Efficient structure detection via random consensus graph", in *Conference on Computer Vision and Pattern Recognition*, 2012.

[186] J.-P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection", in *International Conference on Computer Vision*, 2009.

[187] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms", in *Conference on Computer Vision and Pattern Recognition*, 2007.

[188] E. Elhamifar and R. Vidal, "Sparse subspace clustering", in *Conference on Computer Vision and Pattern Recognition*, 2009.

[189] E. Kreyszig, *Differential geometry*. Dover Publications, 1991, pp. I–XIV, 1–352.

[1]ADATLAP
a doktori értekezés nyilvánosságra hozatalához

I. A doktori értekezés adatai

A szerző neve: Baráth Dániel Béla
MTMT-azonosító: 10048611
A doktori értekezés címe és alcíme: Affine Correspondences and their Applications for Model Estimation
DOI-azonosító[2]:……..
A doktori iskola neve: Eötvös Loránd Tudományegyetem
A doktori iskolán belüli doktori program neve: Az informatika alapjai és módszertana
A témavezető neve és tudományos fokozata: Hajder Levente, PhD
A témavezető munkahelye: Eötvös Loránd Tudományegyetem

II. Nyilatkozatok

1. A doktori értekezés szerzőjeként[3]

a) hozzájárulok, hogy a doktori fokozat megszerzését követően a doktori értekezésem és a tézisek nyilvánosságra kerüljenek az ELTE Digitális Intézményi Tudástárban. Felhatalmazom az Informatika Doktori Iskola hivatalának ügyintézőjét, Kulcsár Adinát, hogy az értekezést és a téziseket feltöltse az ELTE Digitális Intézményi Tudástárba, és ennek során kitöltse a feltöltéshez szükséges nyilatkozatokat.

b) kérem, hogy a mellékelt kérelemben részletezett szabadalmi, illetőleg oltalmi bejelentés közzétételéig a doktori értekezést ne bocsássák nyilvánosságra az Egyetemi Könyvtárban és az ELTE Digitális Intézményi Tudástárban;[4]

c) kérem, hogy a nemzetbiztonsági okból minősített adatot tartalmazó doktori értekezést a minősítés (*dátum*)-ig tartó időtartama alatt ne bocsássák nyilvánosságra az Egyetemi Könyvtárban és az ELTE Digitális Intézményi Tudástárban;[5]

d) kérem, hogy a mű kiadására vonatkozó mellékelt kiadó szerződésre tekintettel a doktori értekezést a könyv megjelenéséig ne bocsássák nyilvánosságra az Egyetemi Könyvtárban, és az ELTE Digitális Intézményi Tudástárban csak a könyv bibliográfiai adatait tegyék közzé. Ha a könyv a fokozatszerzést követőn egy évig nem jelenik meg, hozzájárulok, hogy a doktori értekezésem és a tézisek nyilvánosságra kerüljenek az Egyetemi Könyvtárban és az ELTE Digitális Intézményi Tudástárban.[6]

2. A doktori értekezés szerzőjeként kijelentem, hogy

a) az ELTE Digitális Intézményi Tudástárba feltöltendő doktori értekezés és a tézisek saját eredeti, önálló szellemi munkám és legjobb tudomásom szerint nem sértem vele senki szerzői jogait;

b) a doktori értekezés és a tézisek nyomtatott változatai és az elektronikus adathordozón benyújtott tartalmak (szöveg és ábrák) mindenben megegyeznek.

3. A doktori értekezés szerzőjeként hozzájárulok a doktori értekezés és a tézisek szövegének plágiumkereső adatbázisba helyezéséhez és plágiumellenőrző vizsgálatok lefuttatásához.

Kelt: 2019.5.28.

a doktori értekezés szerzőjének aláírása

[1] Beiktatta az Egyetemi Doktori Szabályzat módosításáról szóló CXXXIX/2014. (VI. 30.) Szen. sz. határozat. Hatályos: 2014. VII.1. napjától.
[2] A kari hivatal ügyintézője tölti ki.
[3] A megfelelő szöveg aláhúzandó.
[4] A doktori értekezés benyújtásával egyidejűleg be kell adni a tudományági doktori tanácshoz a szabadalmi, illetőleg oltalmi bejelentést tanúsító okiratot és a nyilvánosságra hozatal elhalasztása iránti kérelmet.
[5] A doktori értekezés benyújtásával egyidejűleg be kell nyújtani a minősített adatra vonatkozó közokiratot.
[6] A doktori értekezés benyújtásával egyidejűleg be kell nyújtani a mű kiadásáról szóló kiadói szerződést.