DGK Deutsche Geodätische Kommission
der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 712

Timo Dickscheid

Robust Wide-Baseline Stereo Matching for Sparsely Textured Scenes

München 2013

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission beim Verlag C. H. Beck

ISSN 0065-5325 ISBN 978-3-7696-5124-9

Diese Arbeit ist gleichzeitig veröffentlicht in:
Schriftenreihe des Instituts für Geodäsie und Geoinformation
der Rheinischen Friedrich-Wilhelms Universität Bonn
ISSN 1864-1113, Nr. 36, Bonn 2009

DGK Deutsche Geodätische Kommission
der Bayerischen Akademie der Wissenschaften

Reihe C Dissertationen Heft Nr. 712

Robust Wide-Baseline Stereo Matching for Sparsely Textured Scenes

Inaugural-Dissertation zur
Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
der Hohen Landwirtschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms Universität
zu Bonn

vorgelegt

am 20.12.2010 von

Timo Dickscheid

aus Koblenz

München 2013

Verlag der Bayerischen Akademie der Wissenschaften
in Kommission beim Verlag C. H. Beck

ISSN 0065-5325 ISBN 978-3-7696-5124-9

Diese Arbeit ist gleichzeitig veröffentlicht in:
Schriftenreihe des Instituts für Geodäsie und Geoinformation
der Rheinischen Friedrich-Wilhelms Universität Bonn
ISSN 1864-1113, Nr. 36, Bonn 2009

Adresse der Deutschen Geodätischen Kommission:

DGK

Deutsche Geodätische Kommission

Alfons-Goppel-Straße 11 ● D – 80 539 München

Telefon +49 – 89 – 23 031 1113 ● Telefax +49 – 89 – 23 031 - 1283 / - 1100
e-mail hornik@dgfi.badw.de ● http://www.dgk.badw.de

Diese Publikation ist als pdf-Dokument veröffentlicht im Internet unter der Adresse /
This volume is published in the internet

<http://dgk.badw.de> ● <http://hss.ulb.uni-bonn.de/diss_online>

Prüfungskommission

Referent:Prof. Dr.-Ing. Dr.-Ing. h.c. Wolfgang Förstner
1. Korreferent:Prof. Dr.rer.nat. Lutz Plümer

Tag der mündlichen Prüfung:15.07.2011

ISSN 0065-5325 ISBN 978-3-7696-5124-9

Zusammenfassung

Robuste Merkmalszuordnung für Bildpaare schwach texturierter Szenen mit deutlicher Stereobasis

Die Aufgabe von Wide Baseline Stereo Matching Algorithmen besteht darin, korrespondierende Elemente in Paaren überlappender Bilder mit deutlich verschiedenen Kamerapositionen zu bestimmen. Solche Algorithmen sind ein grundlegender Baustein für zahlreiche Computer Vision Anwendungen wie Objekterkennung, automatische Kameraorientierung, 3D Rekonstruktion und Bildregistrierung. Die heute etablierten Verfahren für Wide Baseline Stereo Matching funktionieren in typischen Anwendungsszenarien sehr zuverlässig. Sie setzen jedoch Eigenschaften der Bilddaten voraus, die nicht immer gegeben sind, wie beispielsweise einen hohen Anteil markanter Textur. Für solche Fälle wurden sehr komplexe Verfahren entwickelt, die jedoch oft nur auf sehr spezifische Probleme anwendbar sind, einen hohen Implementierungsaufwand erfordern, und sich zudem nur schwer auf neue Matchingprobleme übertragen lassen.

Die Motivation für diese Arbeit entstand aus der Überzeugung, dass es eine möglichst allgemein anwendbare Formulierung für robustes Wide Baseline Stereo Matching geben muß, die sich zur Lösung schwieriger Zuordnungsprobleme eignet und dennoch leicht auf verschiedenartige Anwendungen angepasst werden kann. Sie sollte leicht implementierbar sein und eine hohe semantische Interpretierbarkeit aufweisen.

Unser Hauptbeitrag besteht daher in der Entwicklung eines allgemeinen statistischen Modells für Wide Baseline Stereo Matching, das verschiedene Typen von Bildmerkmalen, Ähnlichkeitsmaßen und räumlichen Beziehungen nahtlos als Informationsquellen integriert. Es führt Ideen bestehender Lösungsansätze in einer Bayes'schen Formulierung zusammen, die eine klare Interpretation als MAP Schätzung eines binären Klassifikationsproblems hat. Das Modell nimmt letztlich die Form eines globalen Minimierungsproblems an, das mit herkömmlichen Optimierungsverfahren gelöst werden kann. Der konkrete Typ der verwendeten Bildmerkmale, Ähnlichkeitsmaße und räumlichen Beziehungen ist nicht explizit vorgeschrieben. Ein wichtiger Vorteil unseres Modells gegenüber vergleichbaren Verfahren ist seine Fähigkeit, Schwachpunkte einer Informationsquelle implizit durch die Stärken anderer Informationsquellen zu kompensieren.

In unseren Experimenten konzentrieren wir uns insbesondere auf Bilder schwach texturierter Szenen als ein Beispiel schwieriger Zuordnungsprobleme. Die Anzahl stabiler Bildmerkmale ist hier typischerweise gering, und die Unterscheidbarkeit der Merkmalsbeschreibungen schlecht. Anhand des vorgeschlagenen Modells implementieren wir einen konkreten Wide Baseline Stereo Matching Algorithmus, der besser mit schwacher Textur umgehen kann als herkömmliche Verfahren. Um die praktische Relevanz zu verdeutlichen, wenden wir den Algorithmus für die automatische Bildorientierung an. Hier besteht die Aufgabe darin, zu einer Menge überlappender Bilder die relativen 3D Kamerapositionen und Kameraorientierungen zu bestimmen. Wir zeigen, dass der Algorithmus im Fall schwach texturierter Szenen bessere Ergebnisse als etablierte Verfahren ermöglicht, und dennoch bei Standard-Datensätzen vergleichbare Ergebnisse liefert.

Summary

Robust Wide-Baseline Stereo Matching for Sparsely Textured Scenes

The task of wide baseline stereo matching algorithms is to identify corresponding elements in pairs of overlapping images taken from significantly different viewpoints. Such algorithms are a key ingredient to many computer vision applications, including object recognition, automatic camera orientation, 3D reconstruction and image registration. Although today's methods for wide baseline stereo matching produce reliable results for typical application scenarios, they assume properties of the image data that are not always granted, for example a significant amount of distinctive surface texture. For such problems, highly advanced algorithms have been proposed, which are often very problem specific, difficult to implement and hard to transfer to new matching problems.

The motivation for our work comes from the belief that we can find a generic formulation for robust wide baseline image matching that is able to solve difficult matching problems and at the same time applicable to a variety of applications. It should be easy to implement, and have good semantic interpretability.

Therefore our key contribution is the development of a generic statistical model for wide baseline stereo matching, which seamlessly integrates different types of image features, similarity measures and spatial feature relationships as information cues. It unifies the ideas of existing approaches into a Bayesian formulation, which has a clear statistical interpretation as the MAP estimate of a binary classification problem. The model ultimately takes the form of a global minimization problem that can be solved with standard optimization techniques. The particular type of features, measures, and spatial relationships however is not prescribed. A major advantage of our model over existing approaches is its ability to compensate weaknesses in one information cue implicitly by exploiting the strength of others.

In our experiments we concentrate on images of sparsely textured scenes as a specifically difficult matching problem. Here the amount of stable image features is typically rather small, and the distinctiveness of feature descriptions often low. We use the proposed framework to implement a wide baseline stereo matching algorithm that can deal better with poor texture than established methods. For demonstrating the practical relevance, we also apply this algorithm to a system for automatic image orientation. Here, the task is to reconstruct the relative 3D positions and orientations of the cameras corresponding to a set of overlapping images. We show that our implementation leads to more successful results in case of sparsely textured scenes, while still retaining state of the art performance on standard datasets.

Contents

Chapter 1

Introduction

1.1 Motivation

Wide-baseline stereo matching algorithms search for corresponding elements in pairs of overlapping images taken from significantly different viewpoints. To solve this task robustly, it is common practice to consider only stable image features, and use highly distinctive feature descriptions for correspondence analysis. Sparsely textured scenes, such as the empty room with mostly white walls depicted in Figure 1.1, cause two major problems for feature matching:

1. The amount of stable image features is rather small.
2. The distinctiveness of feature descriptions is low.

In this situation, state of the art methods are often not able to produce enough correspondences for solving a particular computer vision problem. For example, automatic image orientation systems often fail in such cases (Dickscheid and Förstner, 2009).

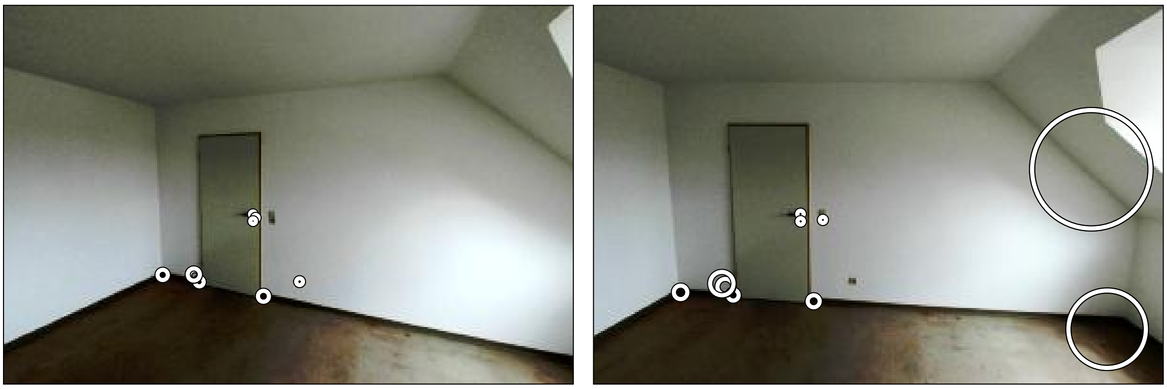


Figure 1.1: A pair of overlapping images showing a scene with sparsely textured surfaces, overlaid with a typical set of local features (Lowe, 2004). The number of features is critically low for applications like image orientation.

The amount of features can be efficiently increased when using multiple feature detectors with highly complementary properties (Dickscheid et al., 2010). To compensate for weak feature descriptions, spatial feature relationships are often used as an additional cue of information (Schmid and Mohr, 1997; Pilu and Lorusso, 1997; Tell and Carlsson, 2002; Schellewald and Schnörr, 2005; Bay et al., 2005; Torresani et al., 2008; Aguilar et al., 2009; Choi and Kweon, 2009). However, such methods usually provide very specific solutions for particular types of spatial relationships and feature operators. This makes it difficult to transfer them to new matching problems. Furthermore, interpretability and semantic correctness of the mathematical models often get lost in favor of computationally efficient implementations.

1.2 Goal and Achievements of the Thesis

The goal of this work is to provide a generic framework for wide baseline stereo matching that seamlessly integrates different types of features, descriptors, and spatial feature relationships as information cues. The key achievement is a sound statistical model for the matching problem, which unifies the ideas of existing approaches and brings them into a well-defined Bayesian formulation. The model has a clear statistical interpretation as the MAP estimate of a binary classification problem and strictly avoids uninterpretable external parameters. It takes the form of a global optimization problem that can be solved with standard optimization methods. A major advantage of the framework is its ability to compensate weaknesses in one information cue implicitly by exploiting the strength of others. To demonstrate its capabilities, we use the framework to implement a feature matching algorithm that can deal better with images of sparsely textured scenes than standard methods, while mostly retaining state of the art performance on regular datasets.

1.3 Applications of the Proposed Method

Applications of wide baseline stereo matching algorithms are numerous. In fact they are a key ingredient to many computer vision systems, and often the first critical computational step. For example, they help to recognize and localize known objects in an image (*object recognition*), to estimate the 3D geometry of cameras (*image orientation*) and scene objects (*3D reconstruction*), or to properly align and possibly fuse different images of the same scene (*image registration*). Although our method is applicable to each of these problems, we will focus on automatic image orientation, where the task is to reconstruct the relative 3D positions and orientations of the cameras corresponding to a set of overlapping images. As a specifically difficult problem, we direct our attention on images of sparsely textured scenes.

1.4 Organization of the Thesis

We start in Chapter 2 by introducing some of the popular feature detectors and descriptors, and describing a standard approach for wide baseline stereo matching based on similarity of feature descriptors. The chapter will also cover some important results about the performance of these techniques in the case of sparsely textured scenes. The relevance of spatial relationships between features is motivated in Chapter 3, where we also give an overview on existing methods for exploiting such relationships.

The core of this work is a framework for robust wide baseline stereo matching that seamlessly integrates a broad range of feature detectors and descriptors together with a variety of spatial relationships. Chapter 4 covers both the derivation of the statistical model and the optimization algorithm of the framework. It finishes with a particular statistical modelling of the observation cues that is especially suited for images of sparsely textured scenes.

We will use different datasets with ground-truth feature correspondences for deriving statistics of descriptor similarities and spatial relationships, and for running the experiments. For this purpose, we develop a novel scheme for automatic annotation of feature correspondences on real image datasets, which is described in Chapter 5.

Chapter 6 presents a number of experimental results that characterize the performance of the proposed framework and relate it to two other popular wide baseline stereo matching algorithms. We use different sets of detectors, descriptors and information cues on image datasets with varying properties for the evaluation.

We conclude in Chapter 7 with a summary of our results, and a discussion about possible extensions and future investigations referring to our work.

1.5 Mathematical Notation

A list of frequently used mathematical symbols is given in Table 1.1 on page 10. It covers the major part of symbols occurring in this document.

With a few exceptions, we will denote sets by calligraphic uppercase letters, vectors by bold lowercase letters, and matrices by bold uppercase letters. Elements of a set are usually represented by the same letter as the set itself, and carry their index as a lower right subscript. The first element in a set has index 1. For example, the set $\mathcal{V}$ of feature matches is $\{v_1, \cdots, v_n, \cdots, v_N\}$. We will generally represent the number of elements in a set by an uppercase letter, and the main index variable over the set by the same letter in lowercase.

As we will deal with two or more images at a time, we use upper right apostrophes to indicate the affiliation of elements to an image. For example, we will often work with two overlapping images $\mathcal{I}'$ and $\mathcal{I}''$, each representing a set of pixels. The set of features extracted from $\mathcal{I}''$ will consequently be denoted as $\mathcal{P}''$. If we use upper right numbers on vectors or sets, they usually indicate the dimensionality.

Finally, we denote continuous probability density functions by the letter p, and the probability of a discrete event by P. Estimated entities are sometimes explicitly marked by a hat, e.g. $\widehat{P}(x)$.

Symbol	**Type**	**Meaning**
$\mathcal{I}'$, $\mathcal{I}''$, ...	sets	input images (sets of pixels)
$\mathcal{P}'$, $\mathcal{P}''$, ...	sets	sets of image features extracted from $\mathcal{I}'$, $\mathcal{I}''$, ...
$\boldsymbol{x}_i$	tuple (x_i, y_i)	position of local feature $\boldsymbol{p}_i$ in an image, given in pixels
α_i	angle	characteristic orientation of local feature $\boldsymbol{p}_i$ in radians
$\boldsymbol{d}_i$	vector	descriptor of local feature $\boldsymbol{p}_i$
$\boldsymbol{p}_i$	tuple	complete local feature $(\boldsymbol{x}_i, \alpha_i, \boldsymbol{d}_i)$
$\mathbf{x}_i = \mathbf{x}(\boldsymbol{p}_i)$	3-vector	homogeneous 2d point representation of $\boldsymbol{p}_i$ (cf. Sec. 4.3.4)
$\mathit{\Sigma}_{xx}(\boldsymbol{p}_i)$	3×3-matrix	covariance matrix corresponding to $\mathbf{x}_i$ (cf. Sec. 4.3.4)
$\mathbf{l}_i = \mathbf{l}(\boldsymbol{p}_i)$	3-vector	homogeneous 2d line representation of $\boldsymbol{p}_i$ (cf. Sec. 4.3.4)
$\mathit{\Sigma}_{ll}(\boldsymbol{p}_i)$	3×3-matrix	covariance matrix corresponding to $\mathbf{l}_i$ (cf. Sec. 4.3.4)
$d(\boldsymbol{d}'_i, \boldsymbol{d}''_j)$	$d \in \mathbb{R}$ (metric)	dissimilarity of two particular feature descriptors, also denoted as s_n if $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$
s_n	$s_n \in \mathbb{R}$	dissimilarity of match $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$, given by $d(\boldsymbol{d}'_i, \boldsymbol{d}''_j)$
$\mathcal{V}$	$\mathcal{V} \subseteq \mathcal{P}' \times \mathcal{P}''$	putative feature matches between $\mathcal{I}'$ and $\mathcal{I}''$
$\mathcal{N}$	set	set of indices $\{1, \cdots, n, \cdots, N\}$ over $\mathcal{V}$
$\mathcal{U}_k$	set	set of all possible k-ary groups of non-redundant putative matches, $\mathcal{U}_k \subseteq \mathcal{V}^k$ (cf. Sec. 4.1.2 and Sec. 4.3.8).
v_n	2-tuple	one (putative) match in $\mathcal{V}$
$\boldsymbol{v}$	N-vector	vector $\boldsymbol{v} = [v_n]$ of all elements in $\mathcal{V}$
λ_n^F	tuple	feature type of match v_n
λ_n^D	tuple	descriptor type of match v_n
λ_n^M	tuple	dissimilarity measure used for match v_n
$\boldsymbol{\lambda}_n$	tuple	$\boldsymbol{\lambda}_n = (\lambda_n^F, \lambda_n^D, \lambda_n^M)$
$\boldsymbol{\lambda}_{\text{type}}$	tuple	a standard setting $(\lambda_n^F, \lambda_n^D, \lambda_n^M)$ for match v_n referring to our particular setup. "Type" can be one of Segment, Blob, AffineRegion or Junction (cf. Section 6.1).
$\mathcal{L}$	set	label domain for variables v_n. We always use $\mathcal{L} = \{0, 1\}$.
l_n	$l_n \in \mathcal{L}$	label assigned to match v_n. By $l_n = 1$ we denote that "match n is selected as an inlier".
$\boldsymbol{l} = f(\mathcal{V})$	vector	particular labeling of all putative matches $\mathcal{V}$, configuration of the corresponding Markov Random Field (cf. Sec. 4.1.2)
$\mathcal{C}_k$	set	set of cliques of order k in a Markov Random Field
$\boldsymbol{s}$	vector $[s_n]$	dissimilarities s_n for all n in $\mathcal{N}$
t_{nm}	$t_{nm} \in \mathbb{R}$	inconsistency measure for a spatial relationship between two matches v_n, v_m.
$\boldsymbol{t}_{nm}$	$\boldsymbol{t}_{nm} \in \mathbb{R}^G$	vector of multiple inconsistency measures t_{nm}.
$\mathcal{T}$	set	set of all observed geometric incompatibility measures $\boldsymbol{t}_{nm}$ related to a complete group of matches
$\mathcal{D}$	set $\mathcal{D} = \{\boldsymbol{s}, \mathcal{T}\}$	all observed data related to a group of feature matches
$\boldsymbol{\theta}$	set	set of all potentials of a Markov Random Field
$\boldsymbol{\theta}^k$	set, $\boldsymbol{\theta}^k \subseteq \boldsymbol{\theta}$	set of k-ary potentials of a Markov Random Field
$\boldsymbol{\theta}^1_{n;l_n}$	$\boldsymbol{\theta}^1_{n;l_n} \in \mathbb{R}$	coefficient related to the unary potential of variable v_n having label l_n
$\boldsymbol{\theta}^2_{nm;l_n l_m}$	$\boldsymbol{\theta}^2_{nm;l_n l_m} \in \mathbb{R}$	coefficient related to the binary potential of variables v_n, v_m having labels l_n, l_m

TABLE 1.1: Mathematical symbols and notation.

Chapter 2

Feature Detection and Description for Wide-Baseline Matching

Wide-baseline feature matching typically starts with a feature detection algorithm, or *feature detector*, which determines an initial set of image elements with desirable properties for correspondence analysis. Such elements are called *image features*. A feature description algorithm then assigns a distinctive description to each feature, which we call a *descriptor*. In this chapter, we will briefly describe some popular methods for wide baseline stereo matching which are solely based on such feature descriptors. The primary goal is to provide an impression for the variety of available image features and the descriptive power of local image intensities.

2.1 Representation of Features and Descriptors

Feature extraction and description is a *mid-level vision process*: It will take a set of images $\{\mathcal{I}', \mathcal{I}'', \mathcal{I}''', \ldots\}$, and return sets of features $\{\mathcal{P}', \mathcal{P}'', \mathcal{P}''', \ldots\}$, as depicted in Figure 2.1 for a minimalistic example. Note that we use apostrophes to indicate the affiliation of an element to a particular image only when necessary. When referring to general elements, we dismiss the apostrophes for simplicity.

In this work, we assume that each feature $\boldsymbol{p}_i \in \mathcal{P}$ can be represented by a location $\boldsymbol{x}_i = (x_i, y_i)$ in the image coordinate system, given in pixels, and a characteristic orientation α_i given in radians. We also assume that the shape of the image region associated to image features can be either represented by an ellipse or a straight line segment. The position $\boldsymbol{x}_i$ is then identified by the ellipse center or the midpoint of a segment. The technical interpretation of α_i can differ according to the particular type of feature: For elliptically shaped features, α_i is usually determined by the most dominant gradient orientation within the elliptical region (cf. Section 2.3). For line segments, the orientation is taken directly from their direction, and the 180 degree ambiguity is resolved by choosing the brighter image intensities to be on the right side, as proposed by Bay et al. (2005).

We will represent local feature descriptors by real-valued vectors $\boldsymbol{d}_i$ and associate them directly to the features themselves. A feature can hence be regarded as a set $\boldsymbol{p}_i = \{\boldsymbol{x}_i, \alpha_i, \boldsymbol{d}_i\}$. Two typical feature description algorithms are presented in Section 2.3.

2.2 Feature Detectors

A broad range of local feature detectors is available today for correspondence analysis under a wide variety of conditions, especially under scale changes, camera rotation and perspective

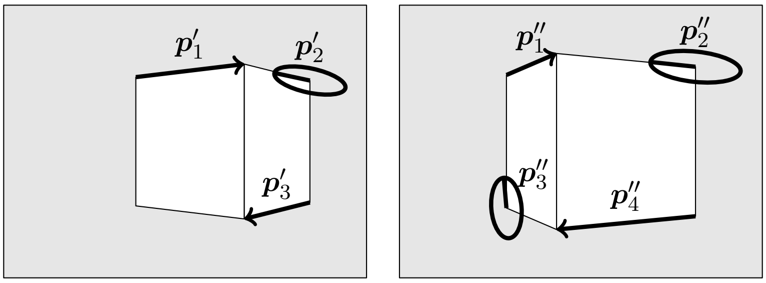


Image $\mathcal{I}'$ with features
$\mathcal{P}' = \{\boldsymbol{p}_1', \boldsymbol{p}_2', \boldsymbol{p}_3'\}$

Image $\mathcal{I}''$with features
$\mathcal{P}'' = \{\boldsymbol{p}_1'', \boldsymbol{p}_2'', \boldsymbol{p}_3'', \boldsymbol{p}_4''\}$

Figure 2.1: Two images $\mathcal{I}'$ and $\mathcal{I}''$ showing an object from different viewpoints, with sets $\mathcal{P}'$ and $\mathcal{P}''$ of local features depicted as arrows and ellipses. The feature sets include straight line segments and elliptically shaped regions. The task is to identify corresponding features. We assume each feature to have an orientation $\alpha \in [0, 2\pi]$. Note that $|\mathcal{P}'| \neq |\mathcal{P}''|$, and that the ordering of the features in the sets is arbitrary.

distortion. Among those algorithms, one usually distinguishes corner, blob, region and edge detectors. We will give an incomplete summary of the more prominent detectors here to illustrate the variability in the methods. A detailed description and categorization has been worked out by Tuytelaars and Mikolajczyk (2008).

The scale invariant blob detector proposed by Lowe (2004), here denoted as lowe, is by far the most prominent one. It is based on finding local extrema of the Laplacian of Gaussians (LoG) of the image, which has the well-known Mexican hat form and therefore aims at extracting dark and bright blobs on characteristic scales of an image. To gain speed, the LoG is approximated by Difference of Gaussians (DoG). The Hessian affine detector (hesaf) introduced by Mikolajczyk and Schmid (2004) is theoretically related to lowe, as it also relies on the second derivatives of the image function over scale space. However, hesaf evaluates both the determinant and the trace of the Hessian instead of taking maxima of the DoG.

The distinction between blobs and regions is not always clear. We will use the term "blob" for features attached to a particular pixel position, representing dark or bright areas around the pixel, while "regions" refer to image patches which are explicitly determined by their boundaries. A very prominent affine region detector is the Maximally Stable Extremal Region detector (mser) proposed by Matas et al. (2004). It computes a watershed-like segmentation with varying thresholds, and selects such regions that remain stable over a range of thresholds. The mser detector is known to have very good matching performance especially on objects with planar structures, and is widely used especially for object recognition. The direct output of the algorithm can be any closed boundary of a segmented region, but often an elliptical approximation of the regions is used. In that case, mser features can be technically used in the same manner as blob features.

Corner features have been used extensively in photogrammetry and computer vision since the works of Förstner and Gülch (1987) and Harris and Stephens (1988). Both of these methods are based on the structure tensor, or second moment matrix, which is computed from the dyadic products of the image gradients. They are known to provide rotation invariance and good localization accuracy. The sfop detector proposed by Förstner et al. (2009) is based on a scale space formulation that directly exploits the structure tensor and the general spiral feature model of Bigün (1990) to detect complementary scale-invariant features. It includes corner features as a special case, and generalizes the point detector in (Förstner, 1994). The Harris affine (haraf) detector (Mikolajczyk and Schmid, 2004) computes the structure tensor on multiple scales to detect 2D extrema within each scale, and then locates characteristic scales at these positions in the Laplacian image pyramid, similar to the HESAF and LOWE detectors.

Line feature detectors usually start with a pixelwise detection of strong gradients, e.g. using the structure tensor, followed by a grouping stage to obtain connected straight or curved components. The most widely known edge detector is the one of Canny (1986), while more advanced approaches are that of Bergholm (1987) and the straight line segment detector included in the framework of Förstner (1994), for example. Lindeberg (1998) proposed a method for detecting scale-invariant line segments.

2.3 Feature Descriptors

Descriptors for point features. The work on descriptors for point-like features is manifold. A survey and evaluation of many techniques is given in Mikolajczyk and Schmid (2005). It is not our intention to analyze the performance of different types of point descriptors again. We will therefore rely on the popular SIFT descriptor proposed by Lowe (2004) for all point-like features throughout our experiments. The SIFT descriptor can be constructed for scale-invariant features $\boldsymbol{p}_i$ with a specific location $\boldsymbol{x}_i = (x_i, y_i)$ in an image, having a characteristic scale σ_i. The scale σ_i identifies the level in a Gaussian scale space pyramid, and thereby defines both the level of blur and the effective size of a circular window that is used for computing the descriptor at position $\boldsymbol{x}_i$. Literally all scale-invariant blob and corner detectors provide such a scale parameter.

The SIFT algorithm starts by assigning a characteristic orientation α_i to each feature $\boldsymbol{p}_i$. This is achieved by searching for dominant peaks in a histogram of gradient orientations within the circular window. Although the histogram bins impose a quantization of 10 degrees, the orientation assignment typical has an empirical accuracy of about 2.5 to 4 degrees, which results from a bilinear interpolation that is applied when filling the bins (Lowe, 2004, Section 5). We will rely on this empirical accuracy later when modelling spatial interactions of features (Section 4.3). In case that multiple peaks are found in the orientation histogram, the feature is duplicated, so that each characteristic orientation induces a separate feature.

The final descriptor is constructed as the concatenation of sixteen weighted orientation histograms, each of which corresponds to a rectangular subregion of the circular patch defined by σ_i. While computing the histograms, gradient orientations are transformed according to the characteristic orientation α_i in order to gain rotation invariance. To achieve robustness against illumination changes, peaks of the histograms are trimmed down to a fixed threshold, and the final descriptor is scaled to unit length.

Descriptors for straight line segments. The work on distinctive descriptions for line features is less comprehensive than that for point- and region-like features. Meltzer and Soatto (2008) recently proposed a sophisticated descriptor suited for scale-invariant lines with mostly general shape, which preserves scale-invariance by exploiting similar concepts as the SIFT descriptors. They obtain impressive matching results especially suitable for object recognition. Bay et al. (2005) have proposed a descriptor for oriented straight line segments based on color histograms, which is very fast to compute but significantly less distinctive than the descriptor of Meltzer and Soatto (2008) or the SIFT descriptors for point features. We will use the descriptor of Bay et al. (2005) in our experiments, so we will give a brief description here.

Given a straight line segment, color intensity profiles are extracted at a distance of three pixels to the left and right from the line, and collected in two separate histograms, one for each

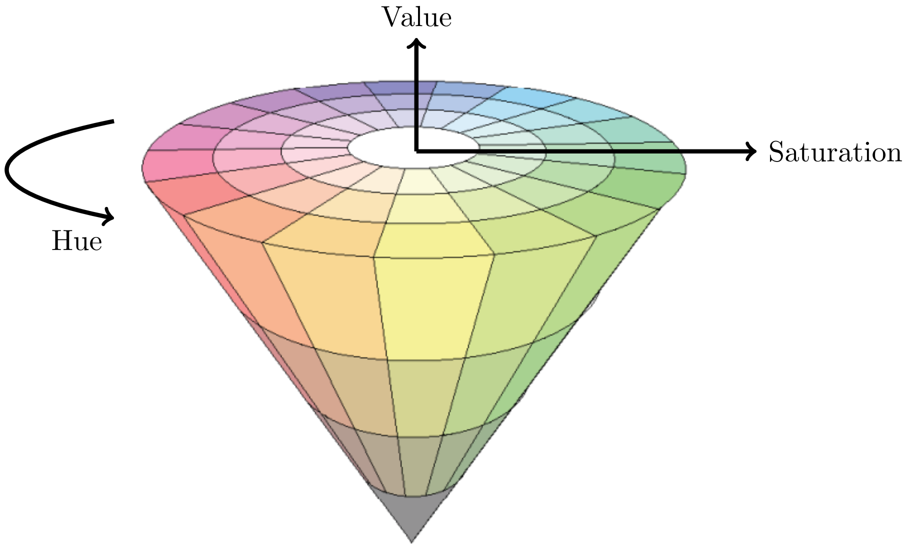


FIGURE 2.2: Conical representation of the HSV color space quantization used for building color histogram in Bay et al. (2005). It uses 18 subdivisions for Hue, three for Saturation, three for Value, and four additional bins for greyvalues in the center of the cone.

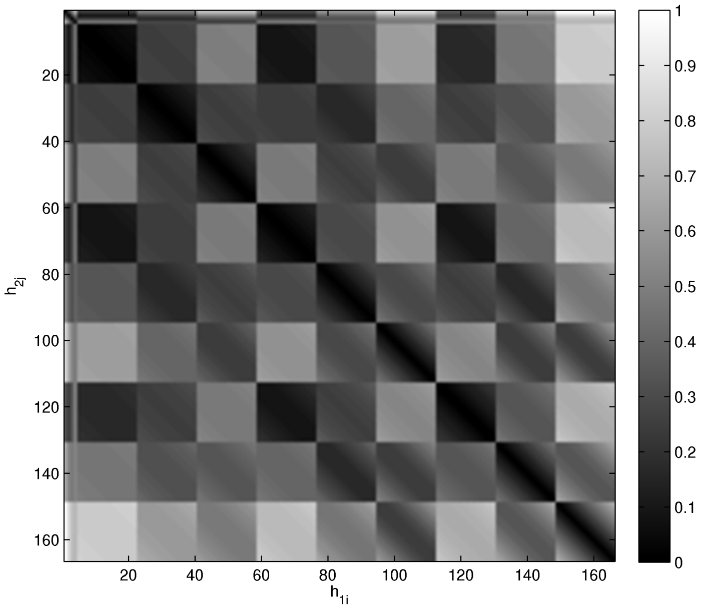


FIGURE 2.3: Plot of the coefficients in the 166 × 166 weight matrix A for computing weighted Euclidean distances of color histograms, as proposed for measuring the similarity of straight line segment descriptors by Bay et al. (2005). The matrix gives the weight of the difference between bin j in color profile h_2 and bin i in color profile h_1, which corresponds to the Euclidean distance of the two colors in the conical representation of the quantized HSV color space shown in Figure 2.2. Bright values denote high weights, dark values low weights, as indicated by the color bar on the right.

of the sides. The two histograms are denoted as h_1 and h_2, respectively. The histograms are based on a strong quantization of the HSV color space, as shown in Figure 2.3, and contain 166 bins each. The histogram values are normalized by the length of the line segment, restricting them to the range $(0, 1)$. The distance between two histograms

$$d_{1,2} = (\boldsymbol{h}_1 - \boldsymbol{h}_2)^{\mathsf{T}} \mathsf{A} (\boldsymbol{h}_1 - \boldsymbol{h}_2) \tag{2.1}$$

has the structure of a Mahalanobis distance. The coefficients of the weight matrix A are derived from the Euclidean distance of color bins in the quantized HSV conic (Bay et al., 2005, Eq. 3). The matrix A is therefore constant. Its structure, arising from a particular vectorization of the HSV cone, is shown in Figure 2.3.

The final descriptor of an oriented straight line segment is composed of the two histograms for the left and right side and represented by a matrix with 2×166 coefficients. The dissimilarity of two descriptors is defined as the square root of the mean of the distance $d_{1,2}$ for both sides, i.e. the corresponding histograms left and right of the two segments.

2.4 Feature Matching based on Descriptor Dissimilarity

Although we will advocate the use of spatial relationships (Section 3.1), the similarity of descriptors is often the most important cue of information. It can be highly effective to simply assign to each feature in the first image its nearest neighbor in the second image, expressed in terms of descriptor dissimilarity. We denote descriptor dissimilarity of two features $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ as $s_{ij} = d(\boldsymbol{d}_i, \boldsymbol{d}_j)$, where $\boldsymbol{d}_i$ is the descriptor for feature $\boldsymbol{p}_i$, and d is a suitable distance measure.

In practice, it may happen that a feature $\boldsymbol{p}_i$ in one image may not have a valid correspondence in the other image at all. To avoid selecting mismatches, Lowe (2004) proposed to compare the descriptor dissimilarity for its nearest and second nearest neighbor. A correspondence is only established if the dissimilarity to the nearest neighbor is significantly smaller than the dissimilarity to the second nearest neighbor. This test has shown to be much more reliable than putting a general threshold on the descriptor distance (Lowe, 2004, Sec. 7.1), and has become a de-facto standard for finding wide baseline stereo correspondences. We will denote it as BestMatch-2 in the following.

The above method only makes a decision about selecting the nearest neighbor or not. It hereby ignores that the second, third, or in general k-th nearest neighbor may also be the true match if the descriptors are not sufficiently distinctive. In order to capture such correspondences, the approach can be generalized as follows:

1. Given two overlapping images $\mathcal{I}', \mathcal{I}''$ with associated sets of features $\mathcal{P}', \mathcal{P}''$, determine the larger of the two feature sets. Let us assume here that $|\mathcal{P}'| > |\mathcal{P}''|$.

2. For every feature $\boldsymbol{p}'_i \in \mathcal{P}'$, determine its $k+1$ nearest neighbors in $\mathcal{P}''$ w.r.t. descriptor dissimilarity.

3. Assuming that the k-th and $k+1$-th nearest neighbors are $\boldsymbol{p}''_l$ and $\boldsymbol{p}''_m$, respectively, check whether $s_{il} < T s_{im}$. A typical value for T is 0.7. If this condition is satisfied, select all k-th nearest neighbors as candidates. Otherwise do not match $\boldsymbol{p}'_i$.

Depending on the value of k, we denote this procedure as BestMatch-k. In the special case of BestMatch-1, the nearest neighbor is always chosen.

We have performed an empirical investigation on the role of the best matching rank k for the combinations of detectors and descriptors described in Section 6.1.1. Figure 2.4 shows the

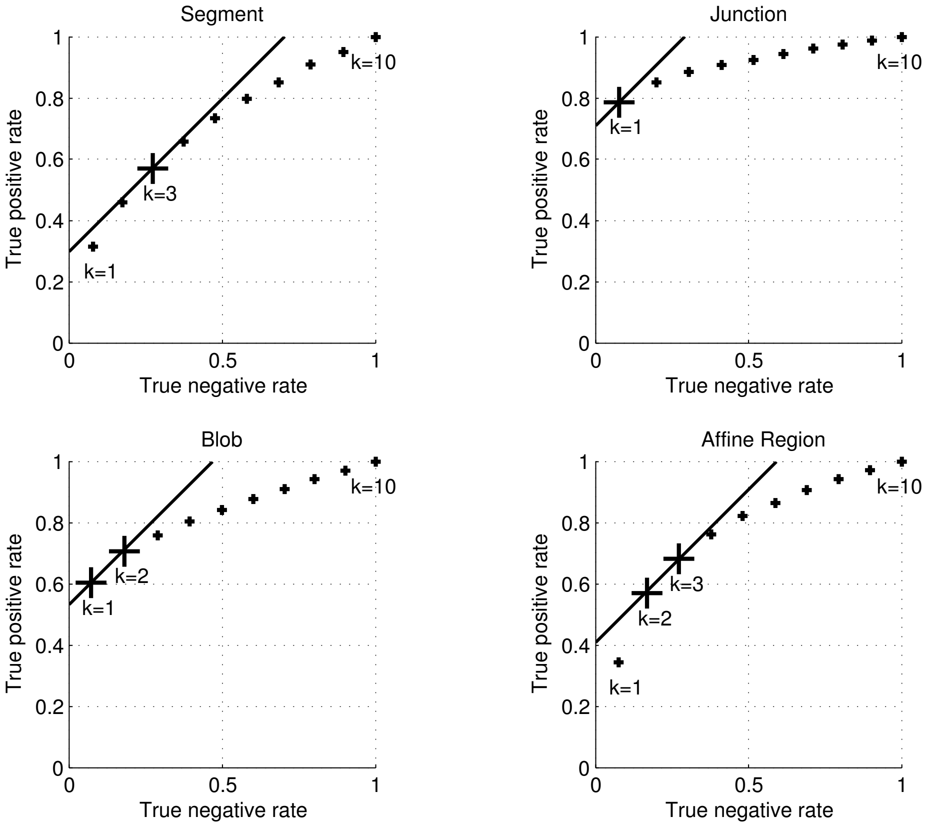


Figure 2.4: Receiver operating characteristic (ROC) statistics per feature type for the selection of putative matches using the BestMatch-k method described in Section 2.4. For increasing values of $k \leq 11$, we plot the true positive rate against the true negative rate when applying the BestMatch-k method. The true positive rate expresses the percentage of correctly selected correspondences among all possible correct correspondences. Accordingly, the true negative rate expresses the percentage of all correspondences that have been correctly discarded among the total number of possible invalid pairs. Hence, a true negative rate of 0.5 is obtained if the data contained 200 possible invalid assignments, 100 of which have been mistakingly selected as correspondences. Good values of k, indicated by the large crosses, were identified by constructing a line with gradient equal to one through each sample, and searching for the line which is nearest to $(0, 1)$. If multiple samples are on the best line, given a tolerance of 0.01, we prefer the one corresponding to the smaller k. The experimental setup used for these plots is described in Section 6.1.

receiver operation characteristic (ROC) coordinates for a large number of putative feature correspondences of a training dataset, using different values for k.[1]

For the chosen junction detector, we see that selecting the nearest neighbor based on the BestMatch-2 method will usually give us over 90 % of the inliers at a true negative rate between 70 and 80 %, which is a very good result. The best possible value would be $(0, 1)$, which is the upper left corner of the diagram. Switching to an affine region detector, here represented by the MSER detector of Matas et al. (2004), we see that the characteristic is different. The best ROC value is achieved when using BestMatch-3. Note that this does not necessarily indicate a weakness of the MSER detector itself, as we do not exploit its full power in our experimental setup (cf. Section 6.1.1). Our intention is to show that the best value of k depends on the particular detector and descriptor combination.

The result for the line segments is worst. Here the best ROC value is achieved using

[1] The annotation procedure used for obtaining the training datasets is explained in Chapter 5. The images of the training dataset are shown on page 88.

$k = 4$. This is not surprising, as we use the descriptors proposed by Bay et al. (2005), which are described in Section 2.3. These descriptors have lower distinctiveness than the SIFT descriptors used for the other features. In fact, the value $k = 4$ for the line segments coincides with the decision of Bay et al. (2005) to initially select the three best matches when using these descriptors.

To conclude, the BestMatch-k method is an effective way of selecting correspondences, but its reliability depends on the type of applied feature, descriptor, and similarity measure. We will use this method for selecting an initial set of putative correspondences for our method. While this principle is also applied by other authors, e.g. Bay et al. (2005) or Choi and Kweon (2009), we will use different k per feature type, hereby directly taking the empirical results of Figure 2.4 into account. Furthermore, we will *not* use a threshold T, in order to avoid unnecessary heuristics. In fact, we assume that a "soft" selection based on descriptor dissimilarity effectively reduces the size of the set of putative matches, but does not eliminate a significant number of true positives.

2.5 Relevance of Complementary Features

As stated before, sparsely textured scenes inhibit two major problems for wide baseline stereo matching: A low amount of detected features, and a possibly reduced distinctiveness of the descriptors.

An obvious and intuitive solution to the first problem is to use multiple feature detectors, which leads directly to a larger number of detected features. However, such a combined feature set may be highly redundant if the detectors have similar characteristics. In general, the amount of image information covered by the features will not increase proportional to the amount of features, except in case of highly complementary detectors. This idea has been the basis for the work in Dickscheid et al. (2010), where we developed a scheme for evaluating the completeness of a feature detector w.r.t. the image information covered by the features. The scheme ultimately allows to find sets of feature detectors with high complementarity.

The basic idea is to define a reference representation for the information contained in an image. Motivated by the coding scheme used in JPEG, this reference is built from an entropy density p_H computed over overlapping local image patches. This density is evaluated over different scales, i.e. different patch sizes.

The information covered by a particular set of features is also represented as a density p_c, based on a normalized sum of anisotropic Gaussians representing each feature. As features may appear on different scales, p_c is implicitly evaluated over scales, in a similar manner as the reference p_H. Based on these two densities, the completeness of a particular feature set w.r.t. an image is then defined by the Hellinger distance of p_c and p_H.

The basic workflow for comparing the completeness of two feature sets is illustrated in Figure 2.5. Smallest distances are obtained for complementary sets of multiple detectors. The work in Dickscheid et al. (2010) has shown empirically that the use of three or four detectors, including a blob and junction detector together with either an edge or a region detector, yields significantly higher completeness than using one or two detectors only. Furthermore, the use of theoretically related detectors, like the LOWE and HESAF detectors (Lowe, 2004; Mikolajczyk and Schmid, 2004), hardly increases the completeness at all compared to using only one of them.

The findings concerning completeness and complementarity of feature detectors refer to the amount of image information which is covered by the features. The question remains if increased coverage of image information does ultimately produce better results for a given application. Indeed, the empirical complementarity measures in Dickscheid et al. (2010) can

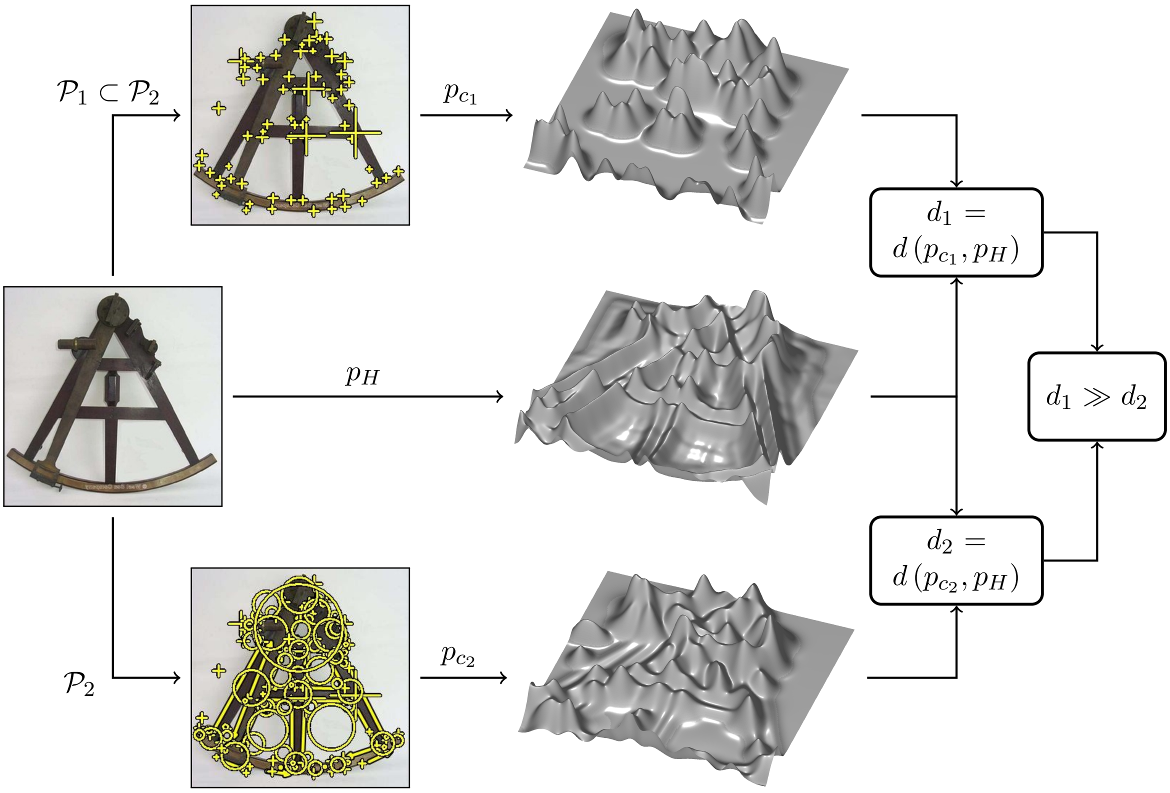


FIGURE 2.5: Principle for comparing the completeness of two different sets $\mathcal{P}_1$, $\mathcal{P}_2$ of local features w.r.t. a particular image proposed by Dickscheid et al. (2010). Incompleteness is defined by the Hellinger distance d of a feature coding density p_c, which is derived from each particular set of features, to an entropy distribution p_H. In the case depicted here we have $\mathcal{P}_1 \subset \mathcal{P}_2$, so we expect lower completeness for $\mathcal{P}_1$, resulting in a higher distance.

be mostly verified when compared to the results of an image orientation procedure that uses different detector combinations as an input (Dickscheid and Förstner, 2009).

The investigations in Dickscheid et al. (2010) have been a strong motivation for the wide baseline stereo matching framework presented in the next chapters. The framework allows for easy integration of feature detectors with different characteristics, and defines a straightforward procedure for “calibrating” the framework for a particular set of detectors. In our experiments, we will demonstrate how weaknesses of some detectors can be compensated seamlessly by exploiting the strengths of others.

2.6 Summary

In this chapter we presented some of the most popular feature detectors, and mentioned the problem of finding reasonable combinations of detectors. Here the complementarity of features plays an important role. We also described two methods for extracting distinctive descriptions of features, one suited for features with elliptical shape and one for straight line segments. An important observation is that the distinctiveness of such descriptors can differ enormously. The feature matching problem can be solved based on descriptors using the BESTMATCH-2 algorithm, but its reliability depends strongly on the type of feature and descriptor. We will therefore motivate the use of spatial relationships as an additional cue of information in the next chapter.

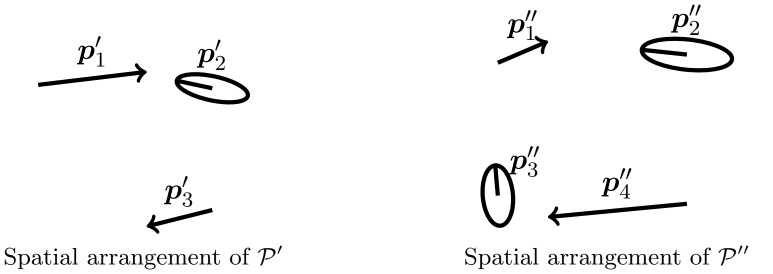
Chapter 3

Exploiting Spatial Feature Relationships

The BestMatch-k approach presented in Section 2.4 exploits the similarity of feature descriptors for finding correspondences, but ignores the spatial arrangement of the features. In this chapter, we will demonstrate that the consistency of spatial feature relationships across views can provide an important additional cue of information. We will also give an overview of existing methods that exploit such relationships, some of which have been a strong inspiration for our own approach.

3.1 Relevance of Spatial Feature Relationships

To motivate the importance of spatial relationships, let us consider the example in Figure 2.1 again, but this time ignore the image content:



Spatial arrangement of $\mathcal{P}'$ Spatial arrangement of $\mathcal{P}''$

For a human observer, it is still possible to find the correct matching. A typical reasoning might be as follows: First of all, we have a feeling that the relative placement of the line segments is consistent across the views: They are located right of each other in both images. As the length of $\boldsymbol{p}'_3$ and $\boldsymbol{p}''_1$ is almost identical, and the distances between $(\boldsymbol{p}'_1, \boldsymbol{p}'_2)$ and between $(\boldsymbol{p}''_4, \boldsymbol{p}''_3)$ are similar, one might argue at first that $\mathcal{P}'$ is roughly rotated by 180 degrees w.r.t. $\mathcal{P}''$. On the other hand, the orientation of $\boldsymbol{p}''_3$ is almost orthogonal to $\boldsymbol{p}''_4$, while $\boldsymbol{p}'_1$ and $\boldsymbol{p}'_2$ have a rather similar orientation, which contradicts the 180 degree rotation. In the end, we feel that the inconsistency in orientation constitutes a stronger violation of spatial arrangement than the inconsistency in distance. We therefore decide that $(\boldsymbol{p}'_1, \boldsymbol{p}'_2)$ on the left correspond to $(\boldsymbol{p}''_1, \boldsymbol{p}''_2)$ on the right.

For such reasoning, a few simple geometric relationships between pairs of features are observed in one view, and then verified in the other. Specifically, the following properties were useful:

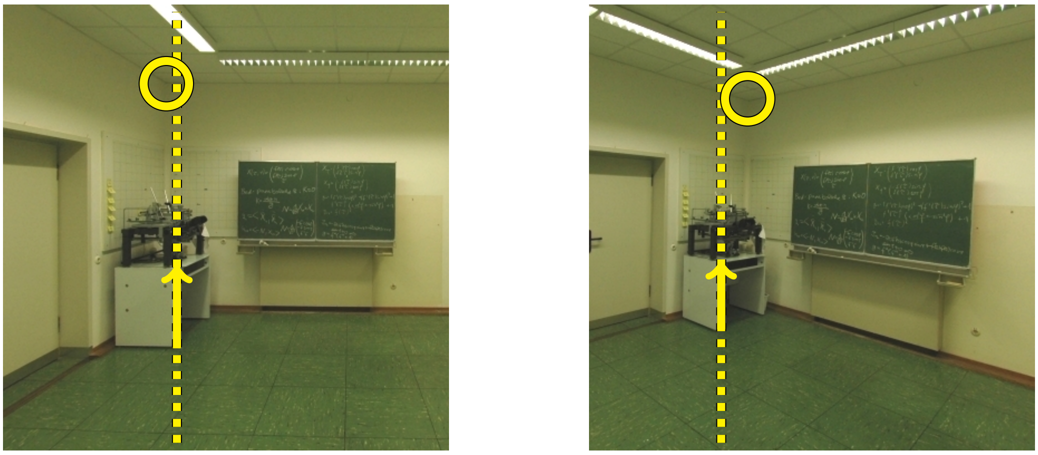


FIGURE 3.1: Typical limitation of the "sidedness" test: Even moderate 3D structure in the scene can cause the test to fail for valid features matches. Here, the upper left corner of the room changes its relative sidedness w.r.t. the border of the table stand after a typical viewpoint change of the camera. Obviously, such geometric relationships in the image domain should not be modelled as hard constraints.

1. *Sidedness.* If a feature is clearly located left or right of another feature in one view, we expect the same spatial relation for their correct correspondences in another view.

2. *Angle.* If two features have a similar orientation in one view, the angle spanned by their orientations is small. Then we also expect the angle spanned by the corresponding features in another view to be small. The same reasoning applies if the angle is large.

3. *Proximity.* If two features are located close to each other in one view, we also expect their correspondences in another view to be close. Here, we intuitively relate closeness to the overall image size.

Among these relationships, only the sidedness was strictly satisfied by the final assignment. The other two relationships were not exactly consistent across views. Instead we used our experience to grant a certain tolerance on the consistency, and put emphasis on the most stable relationship according to our prior experience.

It is important to note that none of the geometric relationships discussed above is generally preserved between feature groups. Figure 3.1 gives an example where even the sidedness between two correct matches is violated for a rather simple scene. However, the value of spatial feature relationships is obvious, especially when using many different observations and applying a soft reasoning that takes prior experience into account.

3.2 Existing Methods

Ullman (1979) identified the three criteria *similarity*, *proximity* and *exclusion* as a key to establishing a good visual mapping. In this section, we will present a number of existing techniques for wide baseline stereo matching, which consider both similarity of feature correspondences and geometric relationships. Most of these will indeed model geometric consistency using a measure of proximity, and employ Ullman's criterion of exclusion by enforcing unique feature correspondences between two views. In our own approach, we will relax the exclusion criterion for reasons explained in Section 4.3.8, and besides proximity use angle and sidedness between pairs of features as spatial relationships.

For the following summary of existing approaches, we need some formalization of the problem. We will denote an initial set of putative feature matches between two images $\mathcal{I}'$ and $\mathcal{I}''$ as $\mathcal{V} = \{v_1, \cdots, v_{\mathcal{N}}\}$. Each element is a pair of features $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$, where $\boldsymbol{p}'_i \in \mathcal{P}'$ and $\boldsymbol{p}''_j \in \mathcal{P}''$, and $\mathcal{P}', \mathcal{P}''$ are the sets of detected features. Thus, $\mathcal{V}$ can be seen as the set of edges in a bipartite Graph with nodes $\mathcal{P}'$ and $\mathcal{P}''$, as illustrated in Figure 4.2 (page 31). A feature itself is represented as described in Section 2.1. We will also use the set of indices $\mathcal{N} = \{1, \cdots, n, \cdots, N\}$ over $\mathcal{V}$, and work with pairs of correspondences defined by sets of index pairs $\mathcal{C}_2 \subseteq \mathcal{N}^2 = \mathcal{N} \times \mathcal{N}$. Accordingly, a set of triplet indices would be denoted as $\mathcal{C}_3$, and so on. The set $\mathcal{N}$ itself could be considered as $\mathcal{C}_1$. Note that we usually assume exactly $\mathcal{C}_k = \mathcal{N}^k$. In case that $\mathcal{C}_k \subset \mathcal{N}^k$, it will be clear from the context.

The descriptor dissimilarity of a putative feature correspondence v_n is computed using some distance measure d over the descriptors, and denoted as $s_n = s_{ij} = d(\boldsymbol{d}'_i, \boldsymbol{d}''_j)$. Descriptor dissimilarities for all elements of $\mathcal{V}$ are collected in the vector $\boldsymbol{s} = [s_n], n \in \mathcal{N}$. For pairs of correspondences $(v_n, v_m) \in \mathcal{C}_2$, we will observe measures of geometric inconsistency which we denote as t_{nm}, sometimes further distinguished by an upper right subscript denoting a particular type of spatial relationship. These are collected in sets $\mathcal{T}$ for each image pair.

Wide baseline stereo matching algorithms have to select a subset of $\mathcal{V}$, so we define a labelling state $l_n \in \{0, 1\}$ referring to each correspondence v_n. By $l_n = 1$ we denote the event that correspondence v_n is selected, and by $l_n = 0$ the event that it is discarded.

3.2.1 Methods Relying on Local Proximity

Most of the earlier works on feature matching with geometric relationships exploit the consistency of spatial relationships between features only within a region of local proximity. This is due to the fact that the instability induced by occlusions and partial visibility of objects increases for distant features. Local proximity is usually defined by a maximum spatial distance or by selecting a fixed number of closest neighbors. For example, Schmid and Mohr (1997) rely on the consistency of angles in a cyclic ordering of neighboring features as illustrated in Figure 3.2. These constraints are collected into a voting framework, which also incorporates information about similarity of features. The authors do not discuss the problem of relative weighting of the different cues of information.

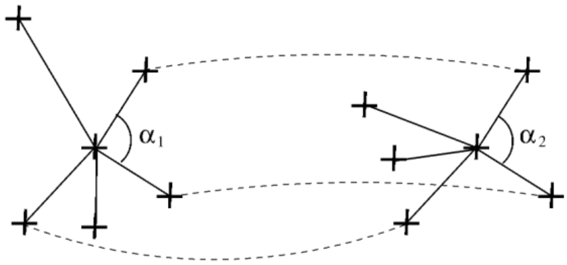


FIGURE 3.2: Semilocal spatial relationships used in Schmid and Mohr (1997). Crosses denote point features, straight lines denote local feature neighborhoods, and dashed lines denote putative correspondences between the two views. The angles $\alpha_{1/2}$ induced by a cyclic ordering of the nearest neighbor features in each view are assumed to be preserved across images. Image taken from Schmid and Mohr (1997).

Tell and Carlsson (2002) propose an algorithm that directly incorporates pairwise spatial relationships into the feature descriptor. It uses scale invariant descriptions based on color intensities along straight lines connecting pairs of corner features, as shown in Figure 3.3. For each feature, a description is computed with its K closest neighbors. The main idea is that the cyclic ordering of connecting lines emitted by a feature is robust under affine transformations in planar scenes. Therefore a feature signature can be constructed by concatenating all descriptions referring to one feature according to the cyclic ordering. Such a signature incorporates both appearance and geometric relationship within a region of local proximity. As the local feature neighborhood of a feature can differ across views, the matching algorithm

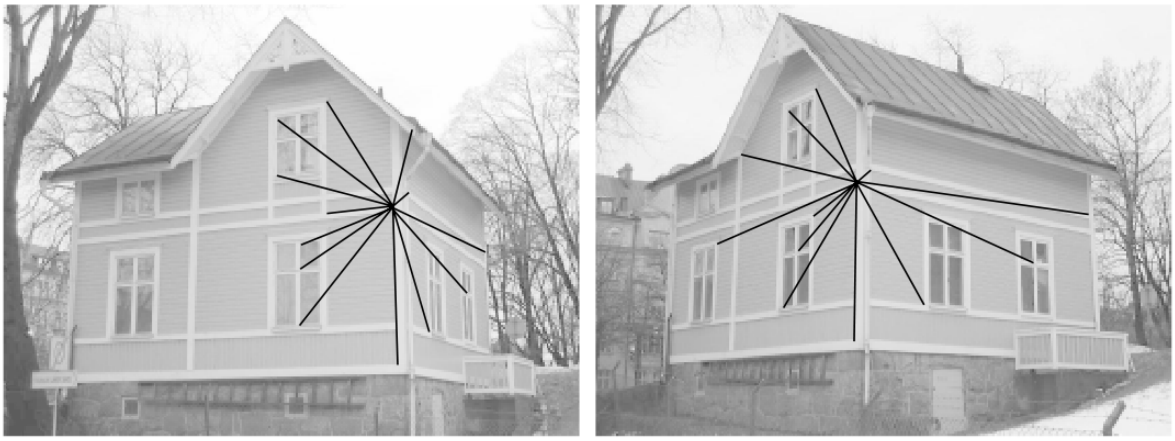


FIGURE 3.3: Illustration of corner feature descriptors used by Tell and Carlsson (2002): For each corner feature, color profile descriptors are computed along the connecting lines to other corner features in the view. The cyclically ordered concatenation of descriptors yields a signature which incorporates both appearance and geometric relationship to other features. The figure shows the connecting lines emitted by one corner feature in each view.

must take different lengths of signatures into account, corresponding to missing descriptions within the signature. Tell and Carlsson (2002) therefore interpret each particular description within the complete signature as a letter in a cyclically ordered string, and apply a technique called *cyclic string matching* for wide baseline stereo. For example, the strings "ABCDAABB" and "CDAB" have "ABCD" as their longest common cyclical substring. If cyclic invariance would not used, the longest common substrings would be "AB" and "CD", The complexity of the string matching algorithm is $O(m^2 \log m)$ for strings of length m, but is in practice reduced by applying a preselection of profiles based on their similarity.

Pilu and Lorusso (1997) propose an approach for wide baseline stereo matching which provides a global solution, covering proximity, exclusion and similarity simultaneously. The basic idea dates back to Scott and Longuet-Higgins (1991). It exploits the properties of a *Singular Value Decomposition (SVD)* for selecting unique correspondences between the sets of features $\mathcal{P}'$ and $\mathcal{P}''$ in images $\mathcal{I}'$ and $\mathcal{I}''$, respectively. To achieve this, a *proximity matrix* $\mathsf{G} \in \mathbb{R}^{|\mathcal{P}'|\times|\mathcal{P}''|}$ is constructed, where each matrix element G_{ij} refers to a possible feature correspondence $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$. In other words, the matrix defines the complete set $\mathcal{V} = \mathcal{P}' \times \mathcal{P}''$ of possible feature correspondences. Each element has the particular form

$$G_{ij} = G_n = \exp\left(-\frac{(C_{ij} - 1)^2}{2\gamma^2} - \frac{|\boldsymbol{x}'_i - \boldsymbol{x}''_j|^2}{2\sigma^2} \right) \tag{3.1}$$

The term $C_{ij} \in (-1, 1)$ is the normalized cross correlation between rectangular image patches of fixed size, centered around each feature location An extension where the cross correlation measure is replaced by SIFT descriptor dissimilarity has been proposed later by Delponte et al. (2006). When computing the Euclidean distance of the two features in the second fraction of (3.1), the two feature locations are treated as if they refer to the same image coordinate system. Both σ and γ are interpreted as Gaussian variances, smoothly restricting the influence of distant feature pairs, and thereby effectively realizing an evaluation of feature pairs within a region of local proximity. For obtaining a solution, the SVD $\mathsf{G} = \mathsf{USV}^\mathsf{T}$ is computed. Then all nonzero values in S are replaced by 1, yielding a new diagonal matrix S^* for calculating an updated proximity matrix $\mathsf{P} = \mathsf{US}^*\mathsf{V}^\mathsf{T}$. This transformation effectively maximizes tr$P^\mathsf{T} G$, and is shown to amplify matrix elements referring to nearby correspondences with high similarity. Because the final correspondences are selected as those elements in P which simultaneously constitute a maximum in the respective row and column, Ullman's exclusion criterion is explicitly fulfilled.

The notion of proximity implemented by Pilu and Lorusso (1997) makes sense for image pairs with small baseline, small affine distortions, and especially negligible rotations between the images. As soon as perspective distortions become significant, the weight of the distance measure needs to be decreased, which effectively shifts the algorithm's behavior towards a correlation-based matching. Most importantly, the structure provided by the proximity matrix is not suitable for evaluating geometric properties of groups of matches, which is one of our primary goals.

Other than the approaches presented above, we will not restrict to regions of local proximity for evaluating geometric relationships. To compensate for the problems caused by geometric distortions between distant features, we will treat all geometric consistency measures as statistically uncertain entities.

3.2.2 Methods Enforcing Global Geometric Consistency

Aguilar et al. (2009) have recently proposed an algorithm for iteratively removing outliers from an initial set of correspondences called *Graph Transformation Matching (GTM)*. An initial set $\mathcal{V}^0$ of putative correspondences is established using descriptor dissimilarities, which does not contain any redundant matches (cf. Section 4.3.8). This set is filtered by explicitly forcing consistency of the local neighborhood structure of groups of matches. The principle idea is to build one graph for each of the two images, which contains matched features as nodes. The vertices in the graph connect to each feature its K nearest neighbors within a fixed radius of local proximity. The GTM algorithm starts by identifying the feature match v_n which causes most inconsistencies in this neighborhood graph across the two views. Then the two graphs are updated using the new set $\mathcal{V}^1 = \{\mathcal{V}^0 \setminus v_n\}$. This two-step process is iterated until the two graphs are strictly isomorphic, hereby explicitly enforcing consistency of the local neighborhood structures. The detection of graph inconsistencies in each iteration is computed efficiently based on the corresponding adjacency matrices.

The GTM algorithm is very effective for eliminating outliers, but has a worst case complexity of $O(N^3 logN)$ in the number of initial correspondences. It focusses on extracting sets of correspondences with an outlier rate near zero, at the cost of possibly discarding a significant number of correct correspondences. Therefore it is not suitable for processing images of sparsely textured scenes. However, the graph-based model for spatial relationships is strongly related to *Relational Matching* (Shapiro and Haralick, 1987), which we will use as a basis for deriving our own model in Section 4.1.1.

The idea of starting with an initial matching based on descriptor similarity and then removing outliers until a level of full geometric consistency is reached has also been applied by Bay et al. (2005). Their approach is based on a weak descriptor-based matching of straight line segments, using the descriptors that we described in Section 2.3. In particular, an initial "softmatching" stage selects for each line segment in one view the three best correspondences in the second view, hereby at first allowing for a high amount of redundant matches. Independently, affine region feature correspondences are computed using classical descriptor-based matching with low redundancy. The iterative filtering stage relies on the *sidedness constraint* between both line segments and affine regions, and implements two tests:

1. For all triples of putative correspondences, the location of one feature w.r.t. to the line connecting the locations of the two other features is checked in the left and right image,

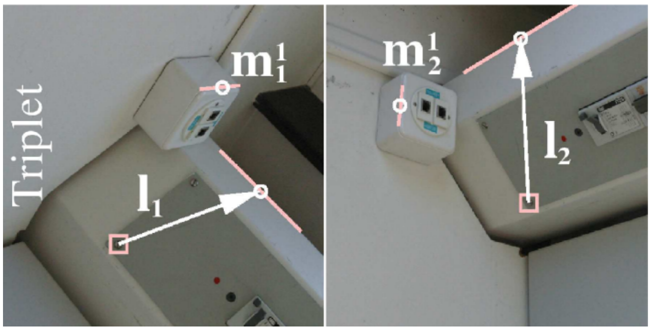


FIGURE 3.4: Illustration of the sidedness test between triples of putative correspondences in Bay et al. (2005): The location of one feature ($m^1_{1/2}$) w.r.t. to the line $l_{1/2}$ connecting the locations of the two other features is assumed to be identical in both images if all three correspondences are correct. Image from Bay et al. (2005).

as shown in Figure 3.4. The test considers whether the feature is located left or right. This relation is assumed to be preserved between the images if all three correspondences are correct.

2. For pairs containing at least one line segment correspondence, the location of one feature w.r.t. to the line (cf. Figure 4.20 left) is checked in both images. The principle is otherwise identical to the case for three correspondences. We will discuss this pairwise test in more detail in Section 4.3.5.

The evaluation of triplets (1.) in principle has cubic complexity in the number of putative correspondences, but an efficient $O(N^2 \log N)$ implementation is described in Ferrari (2004, p. 207f). As the pairwise filter (2.) can only exploit pairs containing at least one line segment, it is by far less powerful than the triplet test, which in turn increases the algorithm's complexity. After computing the sidedness tests over all possible groups of matches, Bay et al. (2005) iteratively determine the correspondence that is involved in the highest number of violations, and deselect it, in the same spirit as the GTM algorithm.

After achieving a level of high geometric consistency, the authors explicitly re-introduce previously unmatched or spuriously filtered correspondences into the final set of correspondences as long as they are geometrically consistent. This "boosting" stage is the most significant difference from the GTM approach, and makes it especially suitable for sparsely textured scenes, as it focusses not only on low outlier rates, but also on delivering a high amount of matches. This principle was a strong inspiration for our own method. However, the explicit treatment of line segment features opposed to the affine regions makes it difficult to use the procedure on arbitrary sets of combined feature types, and it is not clear how other relationships than the sidedness can be integrated smoothly into the setup. Furthermore, the two iteration stages lead to possibly high computation times and make a clear interpretation of the results difficult.

The filtering and boosting stage in Bay et al. (2005) do not take the descriptor similarity of individual softmatches into account, leading to a sequential and independent treatment of similarity and geometry. As we claim that a violation against a "strong" softmatch should have a higher impact than one against a "weak" softmatch, we are interested in a joint problem formulation which integrates arbitrary descriptor dissimilarities with consistency measures for different spatial relationships. Instead of an iterative solution, we aim at a global one.

3.2.3 Methods Based on Energy Minimization

Recently, a number of approaches that incorporate spatial relationships into a global optimization framework have appeared. Such methods are very similar in spirit to our approach,

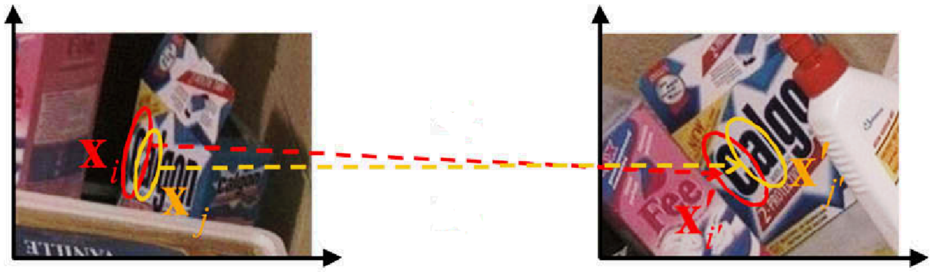


FIGURE 3.5: Local transformations induced by a pair of affine region feature correspondences. Each correspondence v_n defines a 2D affine transformation f_n, denoted by the dashed arrows. Each affine transformation maps an ellipse in one image to the corresponding ellipse in the other image. For correct feature correspondences which refer to the same flat surface in 3D, the transformations are expected to be similar. Image from Choi and Kweon (2009).

as they account for outliers in terms of geometric consistency implicitly by solving a global optimization problem.

Schellewald and Schnörr (2005) use a graph matching approach based on local features for recognizing rigid objects in images, which is formulated as a Quadratic Integer Program. Their approach considers all possible bipartite matchings between the feature sets $\mathcal{P}'$ and $\mathcal{P}''$ of two images, where image $\mathcal{I}'$ shows the object itself, and $\mathcal{I}''$ shows a scene containing an instance of this object. The set $\mathcal{V}$ of putative matches is exactly $\mathcal{P}' \times \mathcal{P}''$. The binary indicator vector $\boldsymbol{x} = \{0,1\}^N = [l_n], n \in \mathcal{N}$ represents all possible labelings corresponding to a bipartite matching. The objective function of the Quadratic Integer Program is

$$\min_{\boldsymbol{x}} \boldsymbol{s}^{\mathsf{T}} \boldsymbol{x} + \alpha \boldsymbol{x}^{\mathsf{T}} \boldsymbol{Q} \boldsymbol{x} \ . \tag{3.2}$$

The first part of the sum simply models the costs for selecting correspondences referring to the feature dissimilarities $\boldsymbol{s}$. The second part of the sum models the costs induced by violating the relational structure of the object, based on consistency of pairwise neighborhood relationships within $\mathcal{P}'$ and $\mathcal{P}''$. The parameter α fixes the relative influence between consistency of the relational structure and the similarity of descriptors. The model of geometric consistency is similar to that used in the GTM algorithm discussed above: The matrix $\boldsymbol{Q}$ is derived from the adjacency matrices corresponding to $\mathcal{P}'$ and $\mathcal{P}''$, respectively. For each pair of features in $\mathcal{P}'$ which is matched to $\mathcal{P}''$, $\boldsymbol{Q}$ will induce a cost of exactly 2 if the corresponding entries in the two adjacency matrices are different. As the objective function (3.2) models only costs for matched features, a trivial minimum is achieved by matching none of the features. Therefore, the Quadratic Program uses linear constraints to restrict the feasible set to solutions where every element of $\mathcal{P}'$ is matched uniquely to an element of $\mathcal{P}''$. As the solution is in general NP hard, the problem is relaxed to a semidefinite program which provides a good approximation of the original problem. The approach of Schellewald and Schnörr (2005) differs from ours in that it considers only neighborhood relationships, and models these as hard constraints. It is also very specific to object recognition, as it assumes that the features of the first image can be found in the second image, and that their neighborhood structure is consistently measured in the second image. However, the formulation as a standard optimization problem is very elegant.

Choi and Kweon (2009) propose a wide baseline stereo algorithm for affine region features, which selects an initial set $\mathcal{V}$ of putative correspondences based on Euclidean distances of SIFT descriptors. They use the local affine transformation H_n that is directly induced by the two affine regions related to each correspondence v_n, as illustrated in Figure 3.5. In particular, $\mathcal{H}_n$ is a 2×2 matrix representing the 2D affine transformation that maps the ellipses corresponding to the regions onto each other, and can be determined directly by the ellipse parameters. The authors exploit the fact that the transformations $\mathcal{H}_n$ and $\mathcal{H}_m$ of two correspondences v_n and v_m are very similar if the features refer to the same smooth and approximately planar surface in 3D. The backprojection error of the features related to v_m under the transformation $\mathcal{H}_n$ (and vice versa) will be small then. To obtain a good matching, an energy function of the form

$$E(l_n, l_m; \boldsymbol{s}, \mathcal{T}) = \sum_{n \in \mathcal{N}} s_n l_n + \sum_{(n,m) \in \mathcal{C}^2} t_{nm} l_n l_m \tag{3.3}$$

is minimized, where the first part of the sum models descriptor dissimilarities, and the second part models geometric consistency of the local affine transformations, with t_{nm} being based on the sum of backprojection errors induced by the affine transformations $\mathcal{H}_n$, $\mathcal{H}_m$, $\mathcal{H}_n^{-1}$ and $\mathcal{H}_m^{-1}$. The descriptor-based part can be considered as a sum over unary energies, as it refers to groups of single matches, and the geometry-related part as a sum over binary energies, as it refers to pairs of putative matches.

Although the method of Choi and Kweon (2009) is very elegant, it has a number of drawbacks. First of all, the unary energies are a linear function of the descriptor dissimilarities s_n. This is not a realistic model, as we will show by empirical distributions of dissimilarities in Section 4.3.3. Second, the relative weighting between the unary and binary energies has no clear semantics, so it is necessary to determine a balancing parameter. The fact that both s_n and t_{nm} are normalized is not sufficient. Third, the model will not work for features with circular shape and straight line segments. It also tends to eliminate possibly correct correspondences of features that do not sit on the same 3D plane as other features. We want to find a formulation that avoids discarding correct matches as far as possible.

Torresani et al. (2008) proposed an approach that is most similar to our work. They start with a set $\mathcal{V}$ of putative matches which contains all possible assignments between $\mathcal{P}'$ and $\mathcal{P}''$, in principle following a graph matching approach. The final correspondences are found by minimizing a complex energy function, which consists of four components that we will briefly discuss.

1. The first energy component, denoted as E^{app}, covers similarity of feature detectors, or “appearance”. It is identical to the sum over unary potentials in (3.3).

2. For explicitly imposing a penalty for unmatched features, the fraction of unmatched features in the smaller feature set is used as a cost. It can be written as a sum over unary energies, denoted as E^{occl}.

3. The component E^{geom} is a sum over binary potentials. It models the consistency of pairwise spatial relationships within a region of local proximity, referring to the orientation and length of the connecting line between two feature locations, which is compared across the two views. The principle is illustrated in Figure 3.6.

4. The energy component E^{coh} constitutes a classical smoothness term. It is the sum of neighboring correspondences v_n, v_m having different labellings, i.e. $l_n \neq l_m$.

These four components can be rewritten so that the final energy takes a very similar form as (3.3), and minimization of the final energy function gives the desired solution. The energy contains four balancing variables λ^{app}, λ^{occl}, λ^{geom} and λ^{coh} for weighting the different potentials, which have to be determined explicitly.

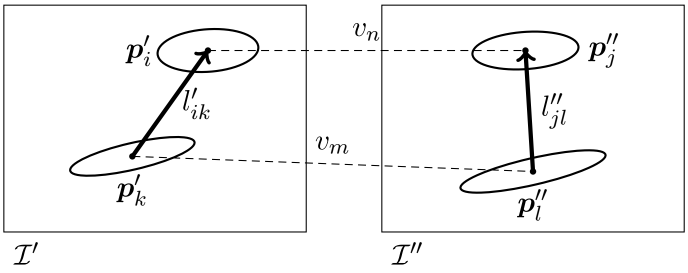


FIGURE 3.6: Model for the consistency of spatial relationships between two correspondences $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$ and $v_m = (\boldsymbol{p}'_k, \boldsymbol{p}''_l)$ used in Torresani et al. (2008). The idea is to test how well the line l'_{ik}, which connects the feature locations in image $\mathcal{I}'$, matches the line l''_{jl} connecting the corresponding locations in image $\mathcal{I}''$. The lines are compared in terms of length and direction, which assumes negligible scale and rotation differences between the views. The shape of the features, here denoted by the ellipses, is not used for the test.

The smoothness term E^{coh} shows that the approach of Torresani et al. (2008) is meant for object recognition and tracking of moving objects. In general wide baseline stereo matching, there is no reason to assume nearby features having the same labels. Also, the model E^{geom} for the binary spatial relationships assumes locally negligible differences in scale and rotation, which is not a typical assumption for wide baseline stereo matching. The most important difference to our approach is that Torresani et al. (2008) model all potentials as positive costs, which are mostly linear in the observations. This is a rather crude approximation of the true relationship between observations and labels, as we will show in Section 4.3. As a simple example, observe that referring to E^{app}, selecting a match with a very small descriptor dissimilarity induces a higher cost than not selecting it. This is not intuitive, because small dissimilarities indicate inliers. This semantic defect is common to most of the approaches discussed above. Torresani et al. (2008) compensate for this problem by introducing E^{occl}, a linear function that explicitly enforces the selection of some inliers. Our approach will not require an artificial term for selecting correspondences, as it uses a more realistic model which implicitly leads to a selection of good correspondences.

Chapter 4

A Generic Framework for Robust Wide-Baseline Stereo Matching

As motivated in Section 3.1, spatial relationships are a valuable cue of information for wide baseline stereo matching. In this chapter we will develop a generic framework that integrates both information about spatial relationships and information about similarity of feature descriptors. We will start by modeling the matching problem for a minimal feature configuration in a deterministic manner, and then introduce a statistical viewpoint. Then we will carefully extend the minimal model towards the general problem, and introduce some reasonable assumptions to make it computationally tractable. We will also describe how the problem can be solved efficiently, and develop a particular instance of the model with specific choices for descriptor dissimilarities and spatial relationships, which are suited well for processing images of sparsely textured scenes.

One important goal is to find a framework that not only leads to a filtering of bad correspondences, but also provokes the selection of geometrically consistent matches despite possibly low descriptor similarity. While Bay et al. (2005) have implemented an explicit boosting stage for producing such effects (cf. Section 3.2.2), we will achieve it as a natural behavior of a statistically motivated, more realistic problem formulation.

We suggested in Section 2.4 to use the rank of descriptor similarity as an initial filter for reducing the amount of putative correspondences from $|\mathcal{P}'||\mathcal{P}''|$ to about $k|\mathcal{P}'|$, with $k \ll 10$, where $\mathcal{P}'$ and $\mathcal{P}''$ are again the sets of features detected in images $\mathcal{I}'$ and $\mathcal{I}''$, respectively. We will use a different value of k per feature type, hereby taking the empirical observations in Figure 2.4 into account. Besides this, we will not apply a threshold on the ratio between the k and $k+1$ best assignments, as in the classical BestMatch-2 approach, following our intention to avoid thresholds wherever possible. Clearly, such preselection of putative matches is a heuristic filtering step, however motivated from our empirical observations on training data in Figure 2.4. In principle, the framework can also deal without a preselection, and start with the full set $\mathcal{V} = \mathcal{P}' \times \mathcal{P}''$. This would lead to significantly higher computation times.

The complete proposed workflow for wide baseline stereo matching of a pair of images is sketched in Figure 4.1, together with two other classical approaches. The illustration shows that the preselection of putative matches (Section 2.4) is directly related to the Softmatching step proposed in Bay et al. (2005). Our approach can be interpreted as a binary classification of the putative correspondences into inliers and outliers. The dataflow implemented by our method is very similar to other approaches relying on energy minimization methods (Schellewald and Schnörr, 2005; Torresani et al., 2008; Choi and Kweon, 2009). Other than these

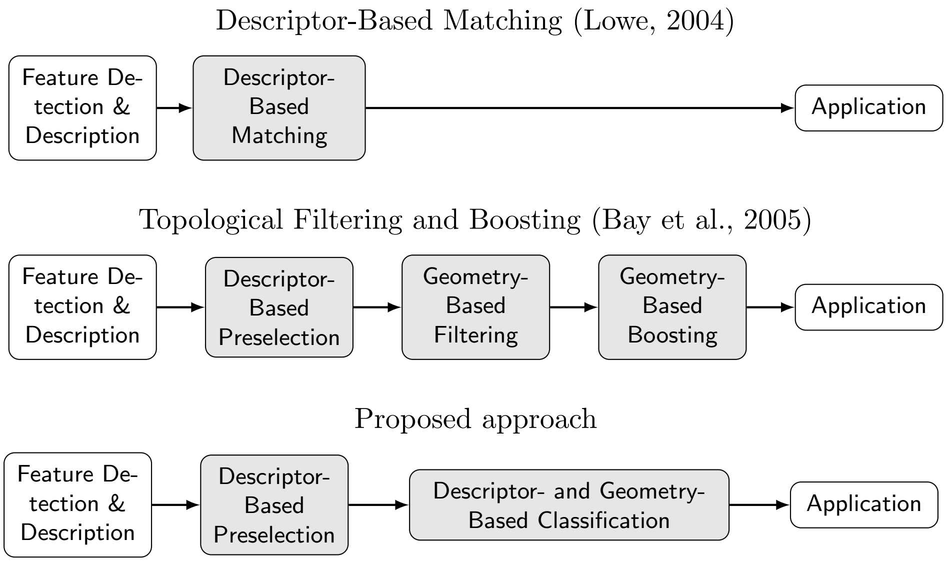

Descriptor-Based Matching (Lowe, 2004)
Feature Detection & Description
Descriptor-Based Matching
Application
Topological Filtering and Boosting (Bay et al., 2005)
Feature Detection & Description
Descriptor-Based Preselection
Geometry-Based Filtering
Geometry-Based Boosting
Application
Proposed approach
Feature Detection & Description
Descriptor-Based Preselection
Descriptor- and Geometry-Based Classification
Application

FIGURE 4.1: Comparison of the basic dataflow for wide baseline stereo matching of a pair of images for three different approaches. *Top:* Standard descriptor-based matching, as in Lowe (2004). *Middle:* Approach of Bay et al. (2005). *Bottom:* Proposed approach.

however, it combines the descriptor-based classification and the two geometrically inspired, iterative filter and boosting steps proposed by Bay et al. (2005) in a well-defined Bayesian treatment.

4.1 Statistical Model for the Matching Problem

In the following sections, we will derive a model for the wide baseline stereo matching problem that is similar in spirit to the ones of Choi and Kweon (2009) and Torresani et al. (2008) discussed in Section 3.2. We will also end up with an energy function that has a very similar form. The particular energy components and the derivation of the model differ significantly however, as we will rely on a Bayesian derivation. Before defining the statistical model, we revisit the problem of modeling spatial relationships for matching problems to get a deeper understanding of the underlying principles.

4.1.1 Representation as a Relational Matching Problem

Descriptor-based approaches for wide baseline stereo matching, as described in Section 2.4, consider each feature individually. Recall the simple matching problem with three and four features in Figure 2.1 on page 12. The standard descriptor-based algorithm would consider an initial set $\mathcal{V} \subseteq \mathcal{P}' \times \mathcal{P}''$ of putative correspondences, which is the set of all edges connecting features of the same type between $\mathcal{I}'$ and $\mathcal{I}''$. By observing six corresponding dissimilarities $\boldsymbol{s} = [s_1, s_2, s_3, s_4, s_5, s_6]$ of feature descriptors, it would select a subset of $\mathcal{V}$ as the solution.

Figure 4.2 illustrates the matching problem in Figure 2.1 as seen by such an algorithm: It is a bipartite graph $\mathcal{G} = (\mathcal{P}', \mathcal{P}'', \mathcal{E})$, where edges in $\mathcal{E}$ represent correspondences, connecting one vertex in $\mathcal{P}'$ with one vertex in $\mathcal{P}''$. The set of putative matches contains only six candidates $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, because under the assumption that good correspondences refer to features of the same type, pairs of line segments and blob features are not considered. The correct matching is denoted by thick edges. Representing the problem in this form visually classifies it as an inexact graph matching problem. However, our goal is to take

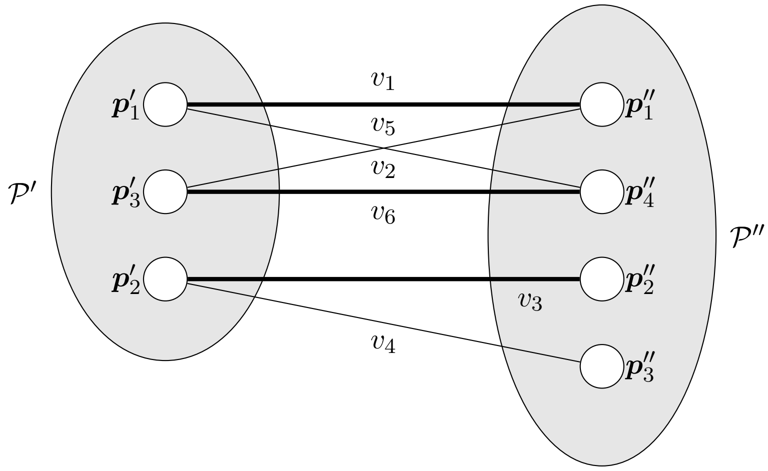


FIGURE 4.2: Descriptor-based matching of the features depicted in Figure 2.1, illustrated as a graph matching problem. The bipartite graph $\mathcal{G} = (\mathcal{P}', \mathcal{P}'', \mathcal{E})$ contains the features as nodes. The edges $\mathcal{E}$ are given by the set of putative matches $\mathcal{V} = \{v_1, \ldots, v_6\}$, connecting features of the same type. The algorithm has to select a good subset of $\mathcal{V}$, ideally the one denoted by the thick lines. The decision is made based on dissimilarities $\boldsymbol{s} = [s_1, \cdots, s_6]$ of feature descriptors.

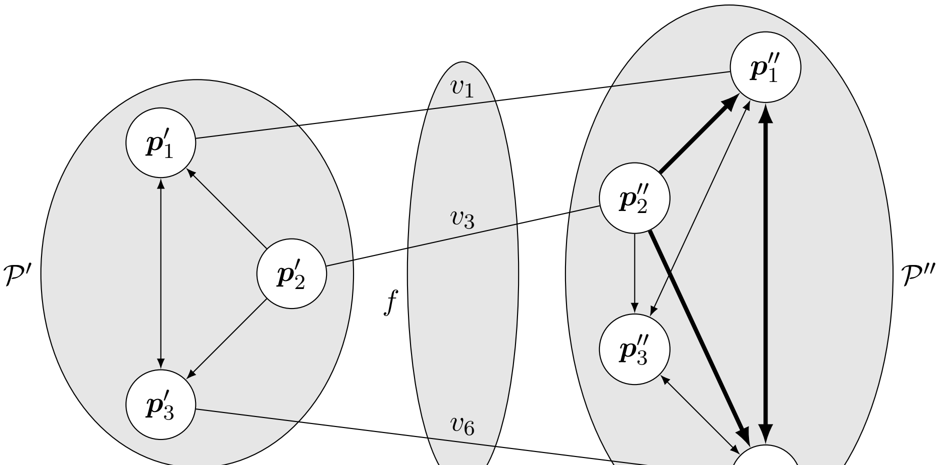


FIGURE 4.3: Relational matching representation of the problem depicted in Figure 2.1. The directed edges (arrows) connect elements within $\mathcal{P}'$ or $\mathcal{P}''$, and denote the binary relationship "is located right of". They are collected in the binary relation sets $\mathcal{R}'_2$ and $\mathcal{R}''_2$. The tuples $(\mathcal{P}', \mathcal{R}'_2)$ and $(\mathcal{P}'', \mathcal{R}''_2)$ each define a *relational description*. The correct matching $\{v_1, v_3, v_6\}$ defines a mapping of a subset of $\mathcal{P}'$ to a subset of $\mathcal{P}''$, which induces a *relational homomorphism*: The composition $\mathcal{R}'_2 \circ f$ is really observed on $\mathcal{P}''$, because $\mathcal{R}'_2 \circ f \subseteq \mathcal{R}''_2$. In other words: If we transfer the relationships in $\mathcal{P}'$ to $\mathcal{P}''$ by the mapping f, they are identical to the already existing relationships in $\mathcal{P}''$ between the mapped elements, as denoted by the thick arrows.

spatial relationships of groups of features into account (Section 3.1), so we need a problem representation that allows us to express structural relationships between the nodes in $\mathcal{P}'$ and $\mathcal{P}''$. Shapiro and Haralick (1987) introduced the concept of *relational descriptions* for finding objects in images, which provides such a representation. They describe an object by the set of its parts $\mathcal{A}$, and represent its structure by a set $\{\mathcal{R}_2, \mathcal{R}_3, \ldots\}$ of binary, ternary, and in general k-ary relations between these parts. For example, $\mathcal{R}_2 \subseteq A \times A$ is the set of binary relationships of object parts, which may contain the relationship "is part of" or "is connected to".

We can transfer the wide baseline matching problem to Shapiro and Haralick's representation by considering images as objects, and image features as object parts. For example, we may consider the two example images $\mathcal{I}'$ and $\mathcal{I}''$ in Figure 4.2 as objects with sets of parts $\mathcal{P}'$ and $\mathcal{P}''$, respectively. We can then extend the representation in Figure 4.2 by a set of binary relations $\mathcal{R}_2$ defined over the features in $\mathcal{P}'$ and $\mathcal{P}''$, for the moment using the spatial relationship "is right of" as an example (cf. Section 3.1). This yields a set of directed edges within the elements of $\mathcal{P}'$ and $\mathcal{P}''$, respectively, as illustrated in Figure 4.3.

Each possible matching of features between $\mathcal{I}'$ and $\mathcal{I}''$ is a mapping $f : \mathcal{P}' \to \mathcal{P}''$, and induces a *composition*

$$\begin{aligned}\mathcal{R}'_2 \circ f = \{(\boldsymbol{p}''_i, \boldsymbol{p}''_j, \cdots) \in \mathcal{R}''_2 \quad | \quad & \exists (\boldsymbol{p}'_m, \boldsymbol{p}'_n, \cdots) \in \mathcal{R}'_2 \\ & \text{with } f(\boldsymbol{p}'_m) = \boldsymbol{p}''_i, f(\boldsymbol{p}'_n) = \boldsymbol{p}''_j, \cdots \}\end{aligned} \tag{4.1}$$

for each set $\mathcal{R}'_2$ of binary relationships between the features in $\mathcal{P}'$. In other words, the composition of binary relations induced by f is the set of binary relations over $\mathcal{P}''$ produced by "transferring" $\mathcal{R}'_2$ according to the feature matching. Such a composition is defined for general sets of k-ary relations over $\mathcal{P}''$. Observe again the example in Figure 4.3: If we transfer the relations $\mathcal{R}'_2$ in $\mathcal{P}'$ to $\mathcal{P}''$ along the edges $\{v_1, v_3, v_6\}$, they constitute a subset of the already existing relations $\mathcal{R}''_2$ in $\mathcal{P}''$, i.e. $\mathcal{R}'_2 \circ f \subseteq \mathcal{R}''_2$. The correspondences represented by the edges are therefore likely to be correct. A mapping f which satisfies $\mathcal{R}'_k \circ f \subseteq \mathcal{R}''_k$ is called a *relational homomorphism.* In the special case where $\mathcal{R}'_k \circ f = \mathcal{R}''_k$, we call f a *relational isomorphism.* Then it represents a perfect symmetric match from $\mathcal{P}'$ to $\mathcal{P}''$.

In wide baseline stereo problems, we will hardly observe a relational isomorphism or homomorphism between the feature sets when using typical geometric relationships. It is thus reasonable to quantify the "error" induced by a set of correspondences. Shapiro and Haralick (1987) define a *structural error* $E^n_s(f)$ that measures both the number of n-tuples in $\mathcal{R}'_k$ which are not mapped to $\mathcal{R}''_k$ by f, and the number of n-tuples in $\mathcal{R}''_k$ which are not mapped to $\mathcal{R}'_k$ by f^{-1}:

$$E^n_s(f) = |\mathcal{R}'_k \circ f - \mathcal{R}''_k| + |\mathcal{R}''_k \circ f^{-1} - \mathcal{R}'_k| \tag{4.2}$$

A simple way to measure the quality of a matching is then to compute the sum of structural errors over all orders of relationships, which is the *relational distance*

$$E(f) = \sum_n E^n_s(f) \; . \tag{4.3}$$

By searching a mapping between $\mathcal{P}'$ and $\mathcal{P}''$ with minimal relational distance $E(f)$ according to a set of spatial relations, it is possible to search geometrically consistent matchings of features in an image pair.[1] This is similar to the methods which enforce geometric consistency

[1] Shapiro and Haralick (1987) use a backtracking search to accomplish this, which has exponential complexity in the number of features. In principle it builds a balanced tree, where each level contains the set

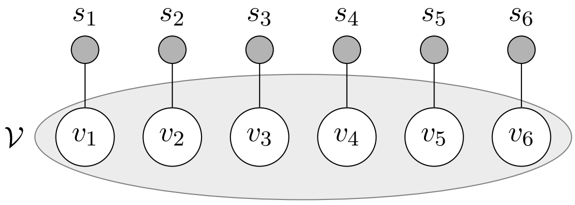


FIGURE 4.4: Graph representing the information used by a descriptor-based matching approach. Given a set of putative matches $\mathcal{V}$, the algorithm observes a descriptor dissimilarity s_n for each of them, and tries to select a good subset of $\mathcal{V}$ as a solution. Following the notation in Bishop (2006), observed values are represented by shaded nodes.

that we presented in Section 3.2.2. Shapiro and Haralick (1987) also present a way to add real-valued attributes to the relations, leading to an extension of the structural distance that incorporates distance measures over attributes of the involved relations.

As discussed in Section 3.1, we intend to model geometric consistency as soft constraints. This would require us to generalize the concept for relational matching explained above. More importantly, we aim at a statistical formulation of the problem. In the next section, we will therefore build on the ideas of relational matching, but transfer them to a different representation for the wide baseline stereo problem, which carefully integrates a preselection of putative matches into the framework, and allows to model descriptor similarity and geometric consistency in a Bayesian treatment.

4.1.2 Representation as a Binary Labeling problem

Let us recall that the classical descriptor-based matching approach observes only dissimilarities of feature descriptors. For each putative match $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$, the dissimilarity $s_n \in \mathbb{R}$ measures a distance $d(\boldsymbol{d}'_i, \boldsymbol{d}''_j)$ of the associated feature descriptors $\boldsymbol{d}'_i$ and $\boldsymbol{d}''_j$. We may collect all dissimilarities corresponding to $\mathcal{V} = \{v_1, \cdots, v_N\}$ in the vector $\boldsymbol{s} = [s_1, \ldots, s_N]$.

The descriptor-based matching approach in fact operates on the level of putative matches: Finding a good matching means to select a good subset of $\mathcal{V}$, given $\boldsymbol{s}$. In that sense, the simple graphical representation in Figure 4.4 contains all required information, despite being simpler than the graph in Figure 4.2, which does not show the observations. Selecting a subset means to assign a label from the set $\mathcal{L} = \{0, 1\}$ to each element in $\mathcal{V}$. This way each vertex v_n becomes a binary random variable defined on the set $\mathcal{L}$ of labels, with a *labeling function* $f : \{1, \ldots, n, \ldots, N\} \rightarrow \mathcal{L}$ assigning a particular label l_n to each variable v_n. If $l_n = 1$, we say that "match n is selected", otherwise "match n is discarded".[2] We will simply use the notation l_n for denoting the particular labeling event $v_n = l_n$. We call a labeling $\boldsymbol{l} = f(\mathcal{V})$ of all variables a *configuration.*

Are we able to transfer the ideas of relational matching into this concept? Consider a minimal example with two putative matches $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$ and $v_m = (\boldsymbol{p}'_k, \boldsymbol{p}''_l)$, having descriptor dissimilarities s_n, s_m, as shown in Figure 4.5. By taking the spatial relationship "is left of" into account, we get the relational matching graph depicted in Figure 4.6.

Obviously, the spatial relationships between features cannot be included directly in a model having the putative matches as its basic elements, as in Figure 4.4. However, according to the discussion in Section 3.1, the observations only have to reflect *consistency* of groups

of features $\mathcal{P}''$ as child nodes per parent. The number of child nodes on the first level is then $|\mathcal{P}''|$, and the number of leaves of the tree is $|\mathcal{P}''|^{|\mathcal{P}'|}$. If violations of spatial relationships are not allowed however, a subset of the branches can be ruled out on intermediate levels of the tree. Depending on the number of constraints, this may reduce the complexity noticeably. By consequently interpreting consistency of spatial relationships as hard constraints, the problem may also be solved with modern *constraint satisfaction methods*, which exploit constraints for reducing the search space more effectively than simple backtracking algorithms, cf. (Rossi et al., 2006).

[2] Observe that f_n only denotes the index of the label in the set $\mathcal{L}$. As the index and the label are identical in our case however, we will not make use of this distinction.

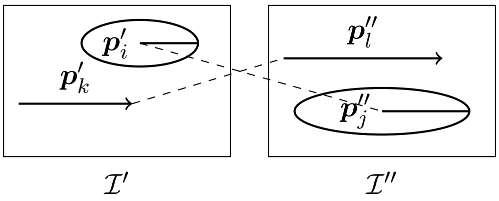


FIGURE 4.5: Minimal example of two images with two features each, having different feature type. Assuming that the line segments correspond to each other, as indicated by the dashed lines, the elliptical regions will most likely not correspond, because the geometric relationship "is left of" would is violated.

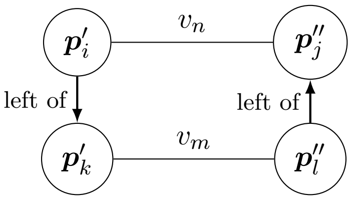


FIGURE 4.6: Relational matching graph for the minimal matching problem in Figure 4.5.

of putative matches w.r.t. geometric relationships, not the relationships themselves. Having putative matches as basic elements is therefore a sufficient model.

We can extend the representation introduced in Figure 4.4 to incorporate the geometric inconsistencies between two putative matches. Both variants are illustrated for the minimal example in Figure 4.7.

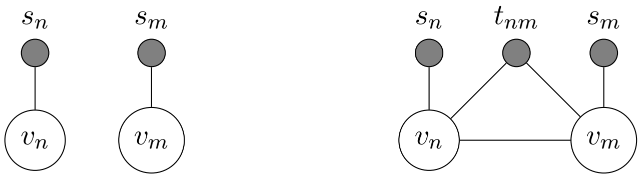


FIGURE 4.7: Graphical representations for the minimal matching problem depicted in Figure 4.5. *Left:* Model for descriptor-based matching, assuming independence between the variables, and using only descriptor dissimilarities s_n, s_m as observations. *Right:* Proposed model, including the pairwise dependency $(v_n, v_m) \in \mathcal{V} \times \mathcal{V}$ between the variables, and observing a geometric inconsistency t_{nm}.

We denote an observed inconsistency measure between two putative matches $(v_n, v_m) \in \mathcal{V} \times \mathcal{V}$, as $t_{nm} \in \mathbb{R}$. Note that this is effectively a function of four variables, reading

$$t_{nm} = t_{nm}(v_n, v_m) = t_{nm}\left((\boldsymbol{p}'_i, \boldsymbol{p}''_j), (\boldsymbol{p}'_k, \boldsymbol{p}''_l)\right) \ . \tag{4.4}$$

Furthermore, we must distinguish different types of geometric relationships, as motivated in Section 3.1. More precisely, we will have G different observations for each pair (v_n, v_m), referring to different types of geometric relationships, leading to a vector of observations $\boldsymbol{t}_{nm} = [t^1_{nm}, \cdots, t^g_{nm}, \cdots, t^G_{nm}]$. This leads us to the extended graph shown in Figure 4.8.

Usually we can only observe inconsistency measures for *non-redundant* – or "unique" – groups of matches. This property will be discussed in more detail in Section 4.3.8. Let us assume for now that we have a function f_u that gives us sets $\mathcal{U}$ of non-redundant groups of matches. For example, $f_u(\mathcal{V} \times \mathcal{V}) = \mathcal{U}_2$ gives us the set of non-redundant pairs of matches. Note the use of the lower subscript on $\mathcal{U}$ to denote the order of the groups. We can then collect all observed data in the set $\mathcal{D} = \{\boldsymbol{s}, \mathcal{T}_2\}$, where $\boldsymbol{s} = [s_n], v_n \in \mathcal{V}$, and

$$\mathcal{T}_2 = \{\boldsymbol{t}_{nm} \mid (v_n, v_m) \in \mathcal{U}_2\} \ . \tag{4.5}$$

Using this binary labeling representation, we will now describe the problem from a statistical viewpoint.

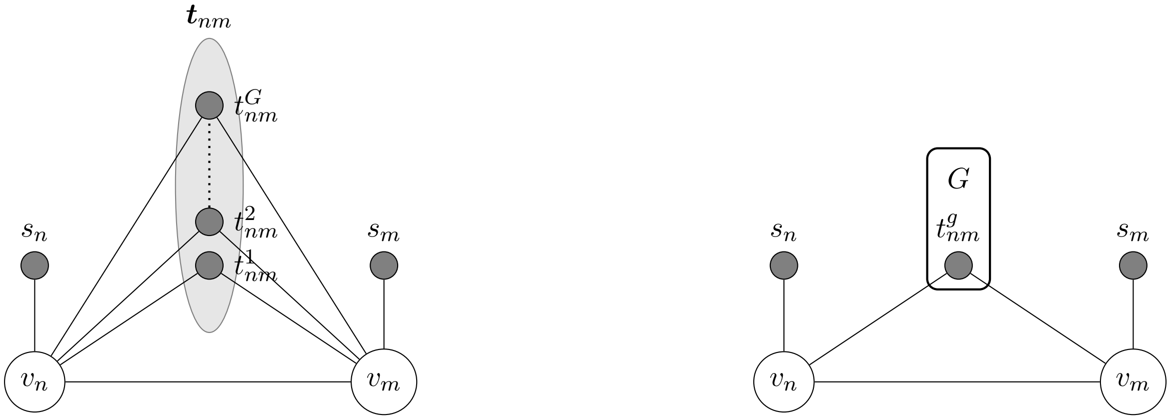


FIGURE 4.8: Extension of the graphical representation in Figure 4.7 right, which illustrates that we obtain G different kinds of observations related to geometric inconsistency of a pair of putative matches, collected in a vector $\boldsymbol{t}_{nm} = [t_{nm}^1, \cdots, t_{nm}^G]$. *Left:* Direct refinement of the graph in Figure 4.7 (right). *Right:* Compact representation of the same model, using the graphical notation of a *plate* (Bishop, 2006, p. 363).

4.1.3 Statistical Derivation of the Local Problem Structure

Independence Assumptions. The representation in Figure 4.7 left reflects the fact that the label of each site v_n is assigned *independently* of other labels by the descriptor based matching approach. Assuming that the descriptor dissimilarities s_n, s_m are both small, this approach would clearly select both matches as inliers. However, we are able to observe inconsistencies $\boldsymbol{t}_{nm}$ of binary geometric relationships, in this case a violation of the relationship "is left of". We must therefore expect that one of the putative matches is an outlier, even though the descriptors are similar! In other words, as we are able to observe the inconsistency, it would be naive to make independent decisions on the two putative matches.

In a statistical treatment, we would say that the random variable v_n is *conditionally dependent* on v_m, given a labeling $\boldsymbol{l} = f(\mathcal{V})$. More precisely, it also depends on the observations s_n and $\boldsymbol{t}_{nm}$. We have already modeled this information by the edges of the graph in Figure 4.8. Interpreted this way, the graph becomes an *undirected probabilistic graphical model.* The graph also models the following *conditional independence assumptions:*

1. All observations are mutually conditionally independent:

$$\begin{aligned} p(s_n, s_m, \boldsymbol{t}_{nm} \mid \boldsymbol{l}) &= p(s_n \mid \boldsymbol{l}) p(s_m \mid \boldsymbol{l}) p(\boldsymbol{t}_{nm} \mid \boldsymbol{l}) \\ &= p(s_n \mid \boldsymbol{l}) p(s_m \mid \boldsymbol{l}) \prod_{g=1}^{G} p(t_{nm}^g \mid \boldsymbol{l}) \end{aligned} \tag{4.6}$$

2. The label of one putative match does not depend on descriptor dissimilarities of other putative matches:

$$p(l_n, s_m) = P(l_n) p(s_m) \ , \quad n \neq m \tag{4.7}$$

Although these assumptions constitute a simplification of the real problem, we have two reasons for choosing them as a model for the local problem structure. First, we claim that it is a meaningful model and still provides enough simplicity to build a fairly fast algorithm from it. This has to be verified by the results that we present later. Second, we will see that the model nicely supports our practical setup, as its statistical dependencies can be estimated particularly well from data.

Maximum a Posteriori Estimate of the Model. The core idea of the approach presented here is to build an algorithm that gives use the *maximum a posteriori estimate (MAP)* of the variables in this local model, given the observed data. In particular, we want to maximize

$$\begin{aligned} & p(l_n, l_m, s_n, s_m, \boldsymbol{t}_{nm}) && (4.8)\\ =\; & p(\boldsymbol{t}_{nm} \mid l_n, l_m, s_n, s_m)\, p(s_m \mid l_n, l_m, s_n)\, p(s_n \mid l_n, l_m)\, P(l_m, l_n) && (4.9)\\ \doteq\; & p(\boldsymbol{t}_{nm} \mid l_n, l_m)\, p(s_m \mid l_m)\, p(s_n \mid l_n)\, P(l_m, l_n) && (4.10)\\ =\; & \left(\prod_{g=1}^{G} p(t^g_{nm} \mid l_n, l_m)\right) p(s_m \mid l_m)\, p(s_n \mid l_n)\, P(l_m, l_n) && (4.11) \end{aligned}$$

for the local structure.[3] Note again the use of l_n as a shorthand notation for the event $v_n = l_n$ here. The factorization in (4.9) results straightforward from repeated application of the product rule of probability. The simplification (4.10) exploits the conditional independence assumptions (4.6) and (4.7), using $p(a \mid b, c) = p(a \mid b)$ in case that a is conditionally independent on c. The expansion of $\boldsymbol{t}_{nm}$ in (4.11) also uses the assumption that all observations are mutually independent.

We will use this local statistical model for two putative matches to build a minimization function that solves the binary labeling problem over arbitrary numbers of putative matches (Section 4.2). We will see later that we can approximate the likelihood distribution components in (4.11) quite well by simple parametric distribution functions, the parameters of which we infer from training data (Section 4.3). For the joint prior $P(l_n, l_m)$, we will assume a uniform distribution that does not contradict the statistics of annotated data. Furthermore, we will use possibly different likelihoods and priors for each feature type, descriptor dissimilarity type, and type of geometric relationship.

Relation to Markov Random Field Theory. The density function (4.11) has a strong relationship to the theory of *Markov Random Fields (MRF)*, seen as the joint probability of the variables in the graphical model in Figure 4.7 (right). An MRF is generally defined by a set of random variables, represented as nodes, and a set of links between pairs of nodes, each of which denotes conditional dependence between the involved variables (Bishop, 2006, Sec. 8.3). We can therefore interpret both models in Figure 4.7 and the model in Figure 4.8 as an MRF.

One of the most important results of MRF theory is that the joint probability of a configuration of the field (or graph) can be factorized into a product of potential functions θ over the *maximum cliques* of the graph.[4] A clique is a set of mutually dependent variables.

[3] The equivalence of maximizing (4.8) and computing the MAP can be seen from the Bayes rule, which gives us the following equation for the posterior distribution:

$$\begin{aligned} P(l_n, l_m \mid s_n, s_m, t_{nm}) &= \frac{p(s_n, s_m, t_{nm} \mid l_n, l_m) P(l_n, l_m)}{p(s_n, s_m, t_{nm})} && (4.12)\\ &= \frac{p(s_n, s_m, t_{nm}, l_n, l_m)}{p(s_n, s_m, t_{nm})} && (4.13) \end{aligned}$$

As the entities s_n, s_m and t_{nm} are observed, the denominator in (4.13) becomes a constant, and can therefore be neglected. The posterior probability density is then equivalent to the joint probability density, which explains that maximizing (4.10) is equivalent to computing the MAP.

[4] The theoretical justification for this result is given by the *Hammersley-Clifford Theorem* (Hammersley and Clifford, 1971), which identifies the joint distribution of an MRF as a Gibbs distribution. We will not explain this equivalence here and refer to the textbooks by Bishop (2006) and Li (2009).

Referring to the graph, this means that the variables in a clique constitute a fully connected subgraph. We denote the set of cliques of two variables by $\mathcal{C}_2$, the set of cliques with three variables as $\mathcal{C}_3$, and in general the set of cliques with k variables by $\mathcal{C}_k$. A *maximum clique* is the special case of a clique where "it is not possible to include any other nodes from the graph in the set without it ceasing to be a clique" (Bishop, 2006, p. 385).

The graph in Figure 4.7 (right) obviously consists of two binary and G ternary maximum cliques. This enables us to write the joint probability for all variables as

$$p(l_n, l_m, s_n, s_m, \boldsymbol{t}_{nm}) = \frac{1}{Z}\,\theta(l_n, s_n)\,\theta(l_m, s_m) \prod_{g=1}^{G} \theta(l_n, l_m, t_{nm}^g) \tag{4.14}$$

where the *partition function* Z is a normalization term which ensures that the result is a valid density. The potential functions θ are usually required to be strictly positive, but need not have a particular interpretation as probability densities.

If our local statistical model is consistent with MRF theory, then (4.14) must be consistent with (4.11) referring to the graph. This can be verified easily if we use the definition

$$\overbrace{p(s_n \mid l_n)}^{a}\,\overbrace{p(s_m \mid l_m)}^{b} \prod_{g=G}^{1} \overbrace{\frac{P(l_m, l_n)}{G} p(t_{nm}^g \mid l_n, l_m)}^{c} \tag{4.15}$$

$$\doteq \frac{1}{Z}\,\underbrace{\theta(l_n, s_n)}_{a'}\,\underbrace{\theta(l_m, s_m)}_{b'} \prod_{g=G}^{1} \underbrace{\theta(l_n, l_m, t_{nm}^g)}_{c'} \tag{4.16}$$

and the fact that $Z = 1$, as in our case the complete term is a valid probability distribution which integrates to one.

4.1.4 Statistical Derivation of the Global Problem Structure

As we have seen in the previous section, we can interpret the local statistical structure for pairs of putative matches (Figure 4.7 right) as a Markov Random Field, which gives us a direct justification for interpreting the local energy potentials (4.14) as the joint probability of its variables. If we model these potentials from the likelihoods, given the observed data, the normalization term cancels out, and we get the compact form (4.11) for deriving a MAP estimate for the local structure containing a pair of putative matches. Now we want to investigate the *global problem structure* when using a larger set of putative matches, under similar conditional independence assumptions. In particular, we want to derive the joint probability of a model that follows directly from Figure 4.8, but uses more than two putative matches. We will also discuss if the resulting MAP estimate is consistent with MRF theory, given the independence assumptions.

To understand the problem structure for larger sets of putative matches, we extend the graph in Figure 4.8 to three matches, using the same conditional independence assumptions (Figure 4.9). It essentially contains three copies of the smaller graph for two matches. An important difference is that the mutual conditional dependence of the three variables v_1, v_2 and v_3 leads to the formation of the clique $(v_1, v_2, v_3) \in \mathcal{C}_3$. Exploiting the independence assumptions in the same manner as for the previous example, the joint probability of the

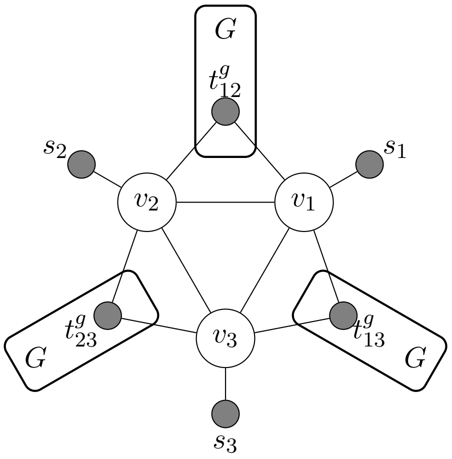


FIGURE 4.9: Graphical representation for a matching problem with three putative matches, extending the minimal model in Figure 4.7 right. The clique $(v_1, v_2, v_3) \in \mathcal{C}_3$ is constituted by the pairwise conditional dependencies of the variables.

graph reads

$$\begin{aligned}
&p(l_1, l_2, l_3, s_1, s_2, s_3, \boldsymbol{t}_{12}, \boldsymbol{t}_{13}, \boldsymbol{t}_{23}) && (4.17)\\
&\quad = p(\boldsymbol{t}_{23} \mid l_1, \cdots, \boldsymbol{t}_{13})\, p(\boldsymbol{t}_{13} \mid l_1, \cdots, \boldsymbol{t}_{12})\, p(\boldsymbol{t}_{12} \mid l_1, \cdots, s_3)\cdot && (4.18)\\
&\qquad \cdot p(s_3 \mid l_1, \cdots, s_2)\, p(s_2 \mid l_1, \cdots, s_1)\, p(s_1 \mid l_1, \cdots, l_3)\, P(l_3, l_2, l_1) &&\\
&\quad = p(\boldsymbol{t}_{23} \mid l_2, l_3)\, p(\boldsymbol{t}_{13} \mid l_1, l_3)\, p(\boldsymbol{t}_{12} \mid l_1, l_2)\cdot && (4.19)\\
&\qquad \cdot p(s_3 \mid l_3)\, p(s_2 \mid l_2)\, p(s_1 \mid l_1)\, P(l_3, l_2, l_1) &&\\
&\quad = P(l_3, l_2, l_1) \left[\prod_{n\in\mathcal{N}} p(s_n \mid l_n)\right] \prod_{(n,m)\in\mathcal{U}_2} p(\boldsymbol{t}_{nm} \mid l_n, l_m) && (4.20)\\
&\quad = P(l_3, l_2, l_1) \left[\prod_{n\in\mathcal{N}} p(s_n \mid l_n)\right] \prod_{(n,m)\in\mathcal{U}_2} \prod_{g=1}^{G} p(t^g_{nm} \mid l_n, l_m) && (4.21)
\end{aligned}$$

Here we use the set $\mathcal{N} = \{1, \cdots, n, \cdots, N\}$ of indices over the set $\mathcal{V}$, and the set $\mathcal{U}_2$ of non-redundant pairs of matches (cf. Section 4.1.2). Note that usually $|\mathcal{U}_2| < |\mathcal{V} \times \mathcal{V}|$ referring to Eq. (4.5).

For a general problem with $|\mathcal{V}| = N$ putative matches, we will obtain a graph having N binary cliques of the form (v_n, s_n) and $G|\mathcal{U}_2|$ ternary cliques of the form (v_n, v_m, t^g_{nm}). In case that no redundant matches are contained in $\mathcal{V}$, hence $\mathcal{U}_2 = \mathcal{V} \times \mathcal{V}$, one obtains exactly one higher order clique of order N. The joint probability of the variables then reads

$$\begin{aligned}
p(l_1, \cdots, l_N, \boldsymbol{s}, \mathcal{T}_2) &= p(l_1, \cdots, l_N, s_1, \cdots, s_N, \boldsymbol{t}_{12}, \cdots, \boldsymbol{t}_{(N-1)N})\\
&= \underbrace{P(l_N, \cdots, l_1)}_{a} \left[\prod_{n\in\mathcal{N}} p(s_n \mid l_n)\right] \prod_{(n,m)\in\mathcal{U}_2} p(\boldsymbol{t}_{nm} \mid l_n, l_m)\,. \qquad (4.22)
\end{aligned}$$

However, the amount of redundant matches is significant in practice, which leads to $|\mathcal{U}_2| \ll |\mathcal{V} \times \mathcal{V}|$. This causes the formation of multiple higher order cliques with sizes larger than three, but significantly smaller than N, depending on the particular situation. These higher order cliques arise only between nodes v_n, due to our independence assumption among the observations. Therefore, they only have an impact on the factor a in (4.22), changing its structure according to the standard rules of conditional independence. This is a serious problem in practice, as it makes the evaluation of a very difficult: It requires us to find an unknown number of maximum cliques with unknown size for each particular matching problem that we want to solve.

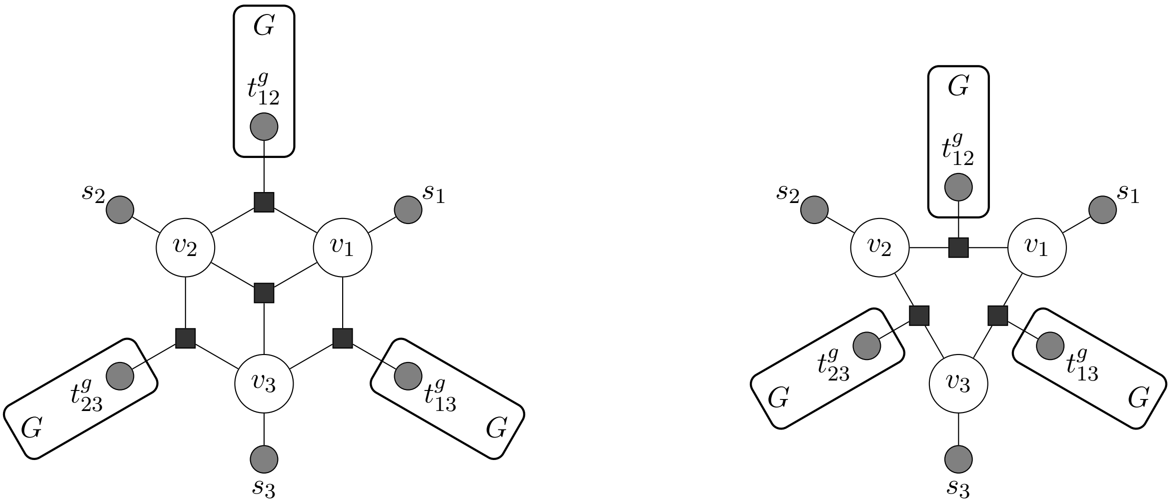


FIGURE 4.10: Two stochastic models corresponding to the three-node problem depicted in Figure 4.9. *Left:* General statistical model, *right:* restricted statistical model arising from the model assumption (4.24). The illustration uses *factor graphs* (Bishop, 2006, Sec. 8.4.3), where cliques are explicitly represented as rectangular nodes.

To make the formulation tractable for practical problems, we model the probabilities of higher order cliques by factors of pairwise cliques, assuming that the inclusion of higher order cliques would not change the results too much:

$$P(l_1, \cdots, l_N) \doteq \frac{1}{Z'} \prod_{(n,m)\in\mathcal{U}_2} \theta(l_n, l_m) \; . \tag{4.23}$$

Again, we want to identify these potentials with the pairwise probabilities. This time however we cannot guarantee that the factorization leads to properly normalized probabilities, so the normalization term does not cancel out. We therefore choose the following model for the joint probability of higher order cliques:

$$P(l_1, \cdots, l_N) \doteq \frac{1}{Z'} \prod_{(n,m)\in\mathcal{U}_2} P(l_n, l_m) \; . \tag{4.24}$$

It leads to a significant simplification of the model. Most importantly, we assume that (4.24) holds irrespective of the particular conditional dependencies among the v_n, which makes it independent of the size and number of the unknown higher order cliques. As we will see in Section 4.2, we do not require specific knowledge about the partition function Z', as it does not affect the final solution.

Just as for the previous independence assumptions, we will verify in our experiments that the model in (4.22) still leads to meaningful results when applying the assumption (4.24). It then reads

$$p(l_1, \cdots, l_N, \boldsymbol{s}, \mathcal{T}_2) = \frac{1}{Z'} \left[\prod_{n\in\mathcal{N}} p(s_n \mid l_n) \right] \prod_{(n,m)\in\mathcal{U}_2} p(\boldsymbol{t}_{nm} \mid l_n, l_m) P(l_n, l_m) \; . \tag{4.25}$$

Note that the factors in the righthand side product actually represent the probability density $p(\boldsymbol{t}_{nm})$. However, we will keep the above form, as it is the representation that we use for getting at a solution.

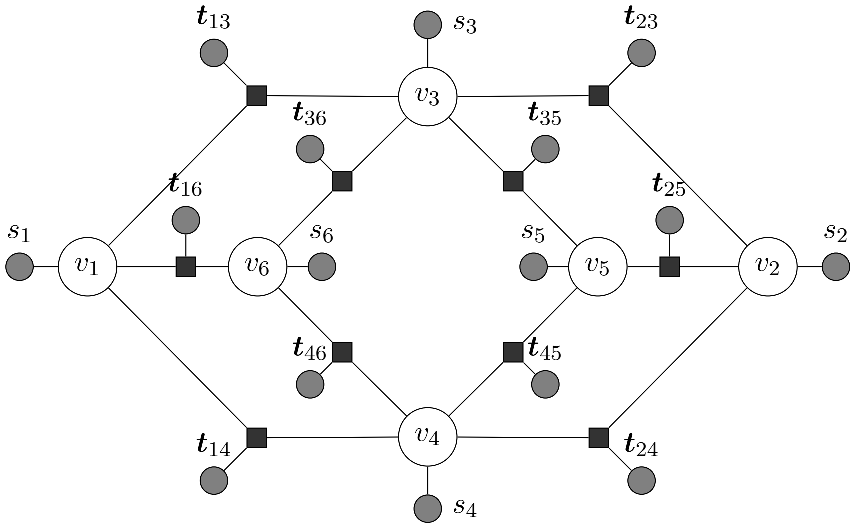


FIGURE 4.11: Stochastic model corresponding to the matching problem in Figure 2.1 on page 12, illustrated as a factor graph (Bishop, 2006, Sec. 8.4.3). For better readability, the G geometric inconsistency measures $\boldsymbol{t}_{nm}$ are drawn as a single node although they should represent a plate as in Figure 4.9. Observe that the maximum clique size between putative matches v_n in the graph is three, as in Figure 4.9, although six putative matches can theoretically lead to the formation of a 6-clique. This is due to the significant amount of redundant matches in $\mathcal{V}$ (cf. Figure 4.2), which causes the number of unique pairs of putative matches to be significantly smaller than the number of all possible pairs, i.e. $|\mathcal{U}_2| \ll |\mathcal{V} \times \mathcal{V}|$.

By going from (4.22) *to* (4.25)*, we make an explicit model assumption.* This leads to a restricted stochastic model which still corresponds to the original graphical model. To get a better understanding of the model assumption, observe the graphs in Figure 4.10. Here we illustrate the stochastic model arising from the assumption (4.24) together with the general statistical model in the form of *factor graphs* (Bishop, 2006, Sec. 8.4.3). These graphs make obvious that the model assumption effectively drops out the higher order cliques between putative matches v_n (i.e. the clique (v_1, v_2, v_3) in Figure 4.10 left) in favor of a change of the pairwise potential functions (rightmost factor of Eq. 4.25).

With increasing number of putative matches, the corresponding graphs become difficult to illustrate. For the introductory wide baseline problem with six putative matches (Figure 2.1), we obtain the graphical representation depicted in Figure 4.11. Here we really have $|\mathcal{U}_2| \ll |\mathcal{V} \times \mathcal{V}|$, as can be seen from the partially missing links among nodes v_n. Note how the graph consists of many substructures containing two putative matches v_n, v_m with observations $s_n, s_m, \boldsymbol{t}_{nm}$, each of which corresponds to the local model in Figure 4.8.

Relation to MRF Theory. As for the local model with two putative matches, the MAP estimate for a fully connected graph with three putative matches is consistent with MRF theory. This can again be seen by defining the factorization of the joint probability in (4.21) as a partition into cliquewise potentials according to Figure 4.9:

$$\begin{aligned} &p(l_n, l_m, l_o, s_n, s_m, s_o, \boldsymbol{t}_{nm}, \boldsymbol{t}_{no}, \boldsymbol{t}_{mo}) \\ &\quad = P(l_o, l_m, l_n) \left[\prod_{n \in \mathcal{N}} p(s_n \mid l_n) \right] \prod_{(n,m) \in \mathcal{U}_2} p(\boldsymbol{t}_{nm} \mid l_n, l_m) \qquad (4.26) \\ &\quad \doteq \frac{1}{Z} \theta(l_n, l_m, l_o) \left[\prod_{n \in \mathcal{N}} \theta(l_n, s_n) \right] \prod_{(n,m) \in \mathcal{U}_2} \theta(l_n, l_m, \boldsymbol{t}_{nm}) \; . \qquad (4.27) \end{aligned}$$

For the general fully connected graph, we can apply the same reasoning to obtain

$$\begin{aligned} & p(l_1, \cdots, l_N, s_1, \cdots, s_N, \boldsymbol{t}_{12}, \cdots, \boldsymbol{t}_{(N-1)N},) \\ & \quad = P(l_1, \cdots, l_N) \left[\prod_{n \in \mathcal{N}} p(s_n \mid l_n) \right] \prod_{(n,m) \in \mathcal{U}_2} p(\boldsymbol{t}_{nm} \mid l_n, l_m) \quad (4.28) \\ & \quad \doteq \frac{1}{Z} \theta(l_1, \cdots, l_N) \left[\prod_{n \in \mathcal{N}} \theta(l_n, s_n) \right] \prod_{(n,m) \in \mathcal{U}_2} \theta(l_n, l_m, \boldsymbol{t}_{nm}) \ . \quad (4.29) \end{aligned}$$

As we obtain a proper probability density, we can omit the normalization term $1/Z$.

Introducing the model assumption (4.24) is also admissible in MRF theory, because the Hammersley-Clifford-Theorem allows us to model the potentials of maximum cliques over arbitrary sub-cliques. In other words, in (4.29) we are free to choose

$$\theta(l_1, \cdots, l_N) \doteq \frac{1}{Z'} \prod_{(n,m) \in \mathcal{U}_2} \theta(l_n, l_m) \doteq \frac{1}{Z'} \prod_{(n,m) \in \mathcal{U}_2} P(l_n, l_m) \ . \quad (4.30)$$

This applies also to partially connected graphs, where the left hand side of (4.30) splits into a subdivided set of potentials of lower orders.

4.2 Finding a Solution

By maximizing the density function (4.25) we realize a MAP estimate of the involved variables. The density can also be written as

$$\begin{aligned} & p(l_1, \cdots, l_N, \boldsymbol{s}, \mathcal{T}_2) \quad (4.31) \\ = \ & \exp \left[\log \frac{1}{Z'} + \sum_{n \in \mathcal{N}} \log p(s_n \mid l_n) + \sum_{\substack{(n,m) \\ \in \mathcal{U}_2}} \left[\log P(l_n, l_m) + \log p(\boldsymbol{t}_{nm} \mid l_n, l_m) \right] \right] \ . \end{aligned}$$

Maximizing (4.31) is equivalent to minimizing the energy function

$$\begin{aligned} & E(l_1, \cdots, l_N, \boldsymbol{s}, \mathcal{T}_2) \quad (4.32) \\ = \ & - \sum_{n \in \mathcal{N}} \log p(s_n \mid l_n) - \sum_{\substack{(n,m) \\ \in \mathcal{U}_2}} \left(\log P(l_n, l_m) + \log p(\boldsymbol{t}_{nm} \mid l_n, l_m) \right) \ , \end{aligned}$$

where we omit the summand $(-\log 1/Z')$ of the partition function, as it does not affect the solution.

Remember that $(l_1, \cdots, l_N)$ denotes a particular labeling of all variables in $\mathcal{V}$. We can write it explicitly as a configuration $f(\mathcal{V})$ of the variables, using the labeling function $f : \mathcal{V} \to \mathcal{L}$. In order to find a good solution for the wide baseline stereo problem, given an initial set $\mathcal{V}$ of putative matches and observations $\mathcal{D} = \{\boldsymbol{s}, \mathcal{T}_2\}$, we finally search for a configuration with minimum energy (4.32). In other words, we look for an optimal solution

$$f^*(\mathcal{V}) = \underset{f(\mathcal{V})}{\operatorname{argmin}} \ E(f(\mathcal{V}), \boldsymbol{s}, \mathcal{T}_2) \ . \quad (4.33)$$

The energy (4.32) is essentially a sum over functions of unary and binary cliques $\mathcal{C}_1$, $\mathcal{C}_2$ of the variables $\mathcal{V}$. Expressing these unary and binary potentials in the form $\boldsymbol{\theta}^1_{n;l_n}$ and $\boldsymbol{\theta}^2_{nm;l_nl_m}$, respectively, we can write the energy as

$$Q(f(\mathcal{V}),\boldsymbol{s},\mathcal{T}_2;\boldsymbol{\theta}) = \sum_{n\in\mathcal{N}} \boldsymbol{\theta}^1_{n;l_n} + \sum_{\substack{(n,m)\\ \in\mathcal{U}_2}} \boldsymbol{\theta}^2_{nm;l_nl_m} \tag{4.34}$$

with

$$\begin{aligned} \boldsymbol{\theta}^1_{n;l_n} &= -\log p(s_n \mid l_n) && (4.35)\\ \boldsymbol{\theta}^2_{nm;l_nl_m} &= -\log p(l_n,l_m) - \log p(\boldsymbol{t}_{nm} \mid l_n,l_m) && (4.36) \end{aligned}$$

Minimization functions of the form (4.34) occur very frequently in computer vision problems. They can be considered as the energy of a general discrete pairwise MRF, meaning that the variables of the corresponding MRF take on discrete values and that the maximum clique size is two.

4.2.1 Solving the Discrete Minimization Problem

The commonness of the minimization problem (4.34) is particularly useful, as it has lead to the development of many algorithms for solving such problems. If defined on a general discrete set of labels, the problem is known to be NP hard. Solutions can be obtained using general-purpose solvers like *Simulated Annealing*, which have exponential time complexity and very slow practical runtimes (Kolmogorov and Zabih, 2004). More efficient minimization algorithms are available for restricted subclasses of the problem. In vision research, much of this work is in the context of pixel-labeling tasks, where *Iterative Conditional Modes (ICM)*, *Loopy Belief Propagation (LPB)* and *Graph Cut* algorithms are very popular, amongst others. Despite assuming restricted sets of labels, these algorithms usually require the neighborhood function, which defines the pairwise cliques $\mathcal{C}_2$, to have a special - usually regular - structure. This is difficult to ensure in our case, where $\mathcal{C}_2 \doteq \mathcal{U}_2$ (Section 4.1.2). Furthermore, most efficient discrete algorithms impose restrictions on the form of the pairwise clique potentials. For example, for some problems with binary labels, the global optimum can be computed in polynomial time with small constants using the swap-move graph cut algorithm, if the binary potentials satisfy the *submodularity constraint*

$$\boldsymbol{\theta}^2_{nm;00} + \boldsymbol{\theta}^2_{nm;11} \leq \boldsymbol{\theta}^2_{nm;01} + \boldsymbol{\theta}^2_{nm;10}\ , \tag{4.37}$$

which states that equal labels are preferred over different labels for neighboring sites, hereby encouraging smooth solutions. This constraint is not satisfied by our model.

It is not the focus of this work to classify discrete minimization methods exhaustively, and we refer the reader to the recent study of Szeliski et al. (2008). We also want to mention that some recent work focusses on relaxing the widely accepted restrictions referring to some minimization algorithms, e.g. the current work of Kolmogorov and Rother (2007).

4.2.2 Solution by Linear Programming Relaxation

It is our intention to provide a mostly generic framework, so we do not want to restrict the potentials further than being density functions. A popular technique for obtaining a very close approximate solution to (4.34) is to relax the combinatorial problem, allowing the variables to take real values in a restricted range, and thereby convert the original problem such that it can be solved using convex optimization methods.

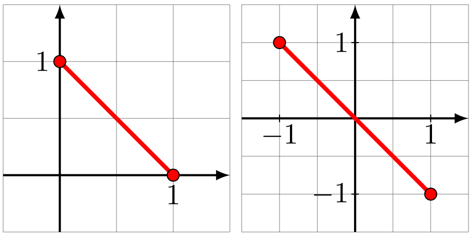


FIGURE 4.12: Feasible set for the relaxed problem of assigning a binary label to a site, shown in red. It is a simplex in $\mathbb{R}^2$. Left: Variable bounds are $[0,1]$, so the sum of variables over all labels is 1. Right: bounds are $[-1,1]$, so the sum is 0.

In particular, we will use the LP-S *Linear Programming Relaxation* which goes back to Schlesinger (1976). It has been shown by Kumar et al. (2009) to be a closer approximation to the original problem than a number of other popular relaxation methods. We will describe the basics of the LP-S relaxation in the following, but refer to the literature for more detailed explanations (Kumar et al., 2009; Li, 2009; Wainwright and Jordan, 2008).

Instead of solving the combinatorial problem with discrete labels, the labeling state for each site v_n is modelled as a real-valued vector $\boldsymbol{x}_n = [x_{n;i}]$ with $i \in \mathcal{L}$, which expands to $\boldsymbol{x}_n = [x_{n;0},\ x_{n;1}]$ for a problem with binary labels. In other words, we represent each labeling state by an individual variable. Similar to the set of putative matches $\mathcal{V} = \{v_n, \cdots, v_N\}$, this relaxation gives us a set of $2N$ relaxed variables

$$\begin{aligned} \mathcal{X} &= \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \cdots, \boldsymbol{x}_N\} && (4.38)\\ &= \{x_{1;0}, x_{1;1}, \cdots, x_{n;0}, x_{n;1}, \cdots, x_{N;0}, x_{N;1}\} && (4.39)\\ &= \{x_1, \cdots, x_q, \cdots, x_Q\}\ , && (4.40)\end{aligned}$$

using the new index range $\mathcal{Q} = \{1, \cdots, q, \cdots, Q\}$, $Q = 2N$, where each index q is directly related to a putative correspondence v_n with label l_n via $q \doteq 2n + l_n - 1$.

The variables x_q are restricted to the range $[0,1]$, and the sum of the two elements in each $\boldsymbol{x}_n$ has to equal one. This reduces the feasible set of solutions for each labeling event to a simplex in the space $\mathbb{R}^2$, which is the straight line segment from $(1,0)$ to $(0,1)$, as illustrated in Figure 4.12 left. Naturally, the relaxation can also be formulated using other ranges: Using the range $[-1,1]$ instead, one obtains the equivalent simplex shown in Figure 4.12 right, where the sum of elements in $\boldsymbol{x}_n$ must equal zero.

Given that a solution for the relaxed variables has been determined, it must be converted back into a set of binary decisions. This requires a disambiguation of the real values into the space consisting of the corner points of the simplex. It is generally not sufficient to perform a rounding to integers or maximum selection – the quality of the approximation to the original problem depends on a proper rounding scheme. We apply the scheme described in Ravikumar and Lafferty (2006), which is also used in the experiments of Kumar et al. (2009).

How does the energy (4.34) *transfer to the new set* $\mathcal{X}$ *of unknown variables?* Obviously each variable v_n in (4.34) is only considered with one particular label l_n, while the set of relaxed variables $\mathcal{X}$ expands over both possible labelings. However, we can easily rewrite (4.34) in the equivalent form

$$Q(f(\mathcal{V}), \boldsymbol{s}, \mathcal{T}_2; \boldsymbol{\theta}) = \sum_{n \in \mathcal{N}} \sum_{u \in \mathcal{L}} x^+_{n;u} \boldsymbol{\theta}^1_{n;u} + \sum_{\substack{(n,m)\\ \in \mathcal{U}_2}} \sum_{u \in \mathcal{L}} \sum_{v \in \mathcal{L}} x^+_{n;u} x^+_{n;v} \boldsymbol{\theta}^2_{nm;uv}\ , \tag{4.41}$$

using the set $\mathcal{L} = \{0,1\}$ and discrete variables

$$x^+_{n;u} \doteq \left\{ \begin{array}{ll} 1, & \text{if } u = l_n \\ 0, & \text{otherwise} \end{array} \right. . \tag{4.42}$$

The variables $x^+_{n;u}$ select explicitly those potentials that refer to the particular label l_n defined by $f(\mathcal{V})$. Observe how the sums in (4.41) are now taken over both labels for each variable, although the energy is exactly identical to (4.34).

Obviously, the set of discrete variables $\{x^+_{1;0}, x^+_{1;1}, \cdots, x^+_{n;0}, x^+_{n;1}, \cdots, x^+_{N;0}, x^+_{N;1}\}$ in (4.41) is then directly related to the set $\mathcal{X}$ of relaxed variables. The energy for the relaxed variables therefore simply reads

$$Q(\mathcal{X}, \boldsymbol{s}, \mathcal{T}_2; \theta) = \sum_{q \in \mathcal{Q}} x_q \boldsymbol{\theta}^1_q + \sum_{(q,r) \in \mathcal{U}^L_2} x_q x_r \boldsymbol{\theta}^2_{q;r} \ , \tag{4.43}$$

where the sums over labels $\{0, 1\}$ are already captured by the index range $\{1, \cdots, Q\}$. Here we use the set

$$\mathcal{U}^L_2 = \{(2n + i - 1, 2m + j - 1) \mid (n, m) \in \mathcal{U}_2,\ i, j \in \{0, 1\}\} \ , \tag{4.44}$$

which collects those index pairs of the relaxed variables that refer to pairs of non-redundant putative matches.

We may now collect the variables $\mathcal{X}$ in a vector $\boldsymbol{x} = [x_1, \cdots, x_Q]^\mathsf{T}$, and the unary potentials in a vector $\boldsymbol{r} = \left[\boldsymbol{\theta}^1_1, \ldots, \boldsymbol{\theta}^1_Q\right]^\mathsf{T}$. In a similar manner, the $Q \times Q$ matrix

$$\mathsf{R} = \begin{bmatrix} R_{1;1} & \cdots & R_{1;Q} \\ \vdots & \ddots & \vdots \\ R_{Q;1} & \cdots & R_{Q;Q} \end{bmatrix} \tag{4.45}$$

with coefficients

$$R_{q;r} = \begin{cases} \boldsymbol{\theta}^2_{q;r} \ , & \text{if } (q, r) \in \mathcal{U}^L_2 \ , \\ 0 \ , & \text{otherwise} \end{cases} \tag{4.46}$$

captures the binary potentials.[5] This enables us to rewrite (4.43) in the form

$$Q(\mathcal{X}, \boldsymbol{s}, \mathcal{T}_2; \theta) = \boldsymbol{r}^\mathsf{T}\boldsymbol{x} + \boldsymbol{x}^\mathsf{T}\mathsf{R}\boldsymbol{x} \ , \tag{4.47}$$

which leads to the quadratic programming (QP) problem

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{r}^\mathsf{T}\boldsymbol{x} + \boldsymbol{x}^\mathsf{T}\mathsf{R}\boldsymbol{x} && (4.48) \\ \text{subject to} \quad & \sum_{i \in \{0,1\}} x_{2n+i} = 1, \quad \forall n \in \mathcal{N} && (4.49) \\ & \boldsymbol{x} \in [0, 1]^Q \ . && (4.50) \end{aligned}$$

Referring to Ravikumar and Lafferty (2006), minimization of this QP is equivalent to maximizing (4.33), which is the desired MAP estimate.

To arrive at a linear programming (LP) formulation, one exploits the fact that the right part $\boldsymbol{x}^\mathsf{T}\mathsf{R}\boldsymbol{x}$ in (4.48) is a scalar, so it equals its own trace when seen as a 1×1 matrix. Therefore we can write $\boldsymbol{x}^\mathsf{T}\mathsf{R}\boldsymbol{x} = \operatorname{tr}(\mathsf{R}\boldsymbol{x}\boldsymbol{x}^\mathsf{T})$, now using the outer instead of the inner product, and the fact that $\operatorname{tr}(AB) = \operatorname{tr}(BA)$. Replacing the nonconvex term $\boldsymbol{x}\boldsymbol{x}^\mathsf{T}$ by a general matrix X, at first neglecting its outer product structure, we obtain the linear objective (4.51), where $\bullet$ denotes the entrywise matrix product, or *Hadamard* product, i.e. $(\mathsf{A} \bullet \mathsf{B})_{ij} = A_{ij}B_{ij}$. The

[5] Note that the 2×2 submatrices on the main diagonal of R usually contain only zero entries, as they refer to binary potentials of a putative match with itself. Kumar et al. (2009) proposed a different formalization of R, which keeps the binary potentials on the main diagonal, and subtracts them explicitly from the unary potentials. This makes R positive semidefinite.

missing outer product structure of X is then compensated by introducing additional linear constraints (4.53) and (4.54), resulting in the final LP formulation

$$\begin{aligned}
\text{minimize} \quad & \boldsymbol{r}^{\mathsf{T}}\boldsymbol{x} + R \bullet X && && (4.51)\\
\text{subject to} \quad & \sum_{i\in\{0,1\}} x_{2n+i} = 1, && \forall n \in \mathcal{N} && (4.52)\\
& \sum_{j\in\{0,1\}} X_{ni;mj} - x_{ni} = 0, && \forall (n,m) \in \mathcal{U}_2; \forall i \in \{0,1\} && (4.53)\\
& X_{q;r} - X_{r;q} = 0, && \forall q,r \in \mathcal{Q} && (4.54)\\
& \boldsymbol{x} \in [0,1]^{Q} && && (4.55)\\
& X \in [0,1]^{Q\times Q} && && (4.56)
\end{aligned}$$

Here, the constraints (4.54) enforce symmetry of X, therefore often denoted as *symmetry constraints.*

The role of the constraints (4.53) is less obvious, but of great importance for obtaining a tight relaxation, so we will describe it in more detail. First, one has to restrict the weights distributed within each particular row of X in order to prevent a possible overemphasis of single variables. In fact, the LP relaxation described in Li (2009) restricts the sum of weights over complete rows of X (Li, 2009, Eq. 9.54). This allows very sparse solutions for X, with high weights for a small number of row coefficients and zero weights for others, which often leads to poor solutions. The constraints (4.53) are much stronger: Here the row weights are restricted separately for each pair (n, m) of variables in the original problem. More precisely, given a particular label i for the variable n, which corresponds to one row in X, we require the sum of each two row coefficients $X_{ni;mj}$, $j \in \{0, 1\}$, to be equal to the weight x_{ni}. This constraint is related to marginalization of the conditional probabilities within pairwise cliques of the original problem, i.e. the probability of variable v_n having label i must respect

$$P(v_n = i) = \prod_{j\in\{0,1\}} P(v_n = i \mid v_m = j) \qquad (4.57)$$

for each pairwise clique it is involved in, given that the maximum effective clique order of the original MRF is two. Consequently, one usually refers to (4.53) as the *marginalization constraints.*

The problem in Eqs. (4.51)-(4.56) is the so-called LP-S relaxation described in Chekuri et al. (2001), which goes back to the work of Schlesinger (1976).

4.2.3 Complexity Considerations

We use the LP formulation (4.51)-(4.56) followed by the rounding scheme described in Ravikumar and Lafferty (2006) to obtain a good approximation of the MAP estimate in (4.32). The minimization can be done using standard linear programming solvers, which are available in many variants due to the broad range of applications where linear programs occur. For example, the original use of LP's was designed for optimization of flow in transportation networks.

Most freely available LP solvers provide an efficient implementation of George Dantzig's *simplex algorithm.* In principle, this algorithm traverses the edges of the high-dimensional polytope representing the feasible region of the LP, until it arrives at the corner point with minimal energy. In contrast to classical least squares estimation techniques, the exact complexity of solving a particular linear program with the simplex algorithm cannot be given (Boyd and Vandenberghe, 2004, 1.2.2). It can be shown that the worst-case complexity of

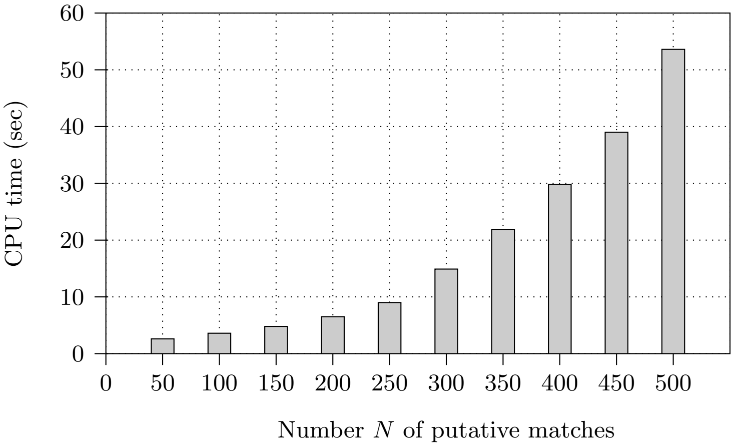


Table 4.1: CPU times for matching LOWE features on images 1 and 2 of the Graffiti sequence using the proposed method on an Intel Core 2 Duo CPU with 2.4 GHz speed. The times are given for different sizes N of the initial set of putative matches. Note that the values also include the time for extracting features and descriptors, and for evaluating the dissimilarity measures and spatial relationships. For minimizing the LP problem we used Mosek (`http://www.mosek.com`), which implements an interior point method.

the simplex algorithm is exponential in the number of extremal points, but the practical complexity for most problems is polynomial with very good convergence properties.

The more recent *interior point* or *barrier methods* instead have both polynomial average complexity and polynomial worst-case complexity with small exponents.[6] In contrast to the simplex methods, they iteratively construct strictly feasible points in the interior of the polytope which converge towards the optimal value. We use the commercial Mosek package[7] for solving our problem, which provides an efficient implementation of the interior point algorithm for solving linear programs with up to thousands of variables. For sets $\mathcal{V}$ of putative correspondences with $N = |\mathcal{V}| < 500$, we usually obtain the solution in less than a few seconds on a standard 2.4 GHz CPU. For sparsely textured scenes, N is typically smaller than 200, leading to negligible computation times for obtaining the optimal solution.

To give a feeling on the performance of the complete matching algorithm, including feature detection and description as well as evaluation of spatial relationships, CPU times for a real matching problem with increasing problem sizes are shown in Table 4.1.

4.3 Data-Driven Modeling of Energy Potentials

The proposed minimization function (4.32) consists of three basic elements:

1. The likelihood $p(s_n \mid l_n)$ describes how likely it is to observe a particular descriptor dissimilarity, given the label of the match.
2. The likelihood $p(t^g_{nm} \mid l_n, l_m)$ describes how likely it is to observe a geometric inconsistency measure of type g, given one of four possible labelings of the corresponding pair of matches.
3. The probability $P(l_n, l_m)$ indicates how likely it is at all to observe a particular labeling of a pair of matches.

In this section, we will derive particular models for these likelihoods and priors that are suited well for robust wide baseline stereo matching of images with sparse texture. *Note that the use of these particular models is not prescribed by our framework. They should be considered as one possible implementation of the proposed method.*

[6] For example, the algorithm of Karmarkar (1984) has worst-case complexity which is polynomial in the number of variables with an exponent of 3.5.

[7] `http://www.mosek.com`

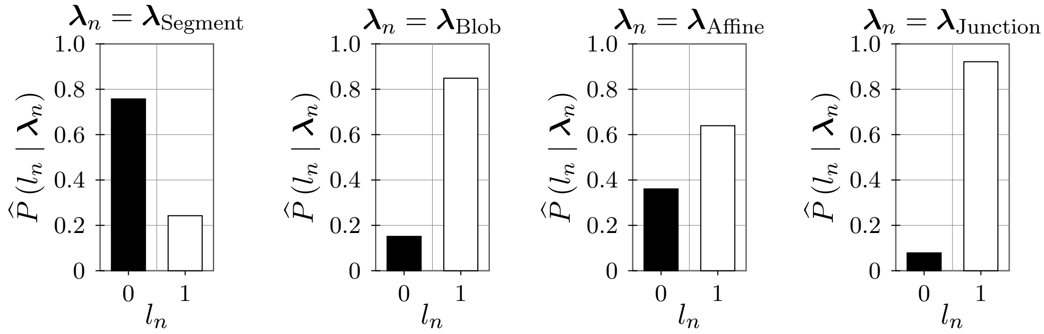


FIGURE 4.13: Relative frequency of good ($l_n = 1$) and bad ($l_n = 0$) putative matches observed on the training data, which can be seen as an estimate for the prior $\widehat{P}(l_n \mid \boldsymbol{\lambda}_n)$ for the different feature types.

We will derive parametric functions for approximating the likelihoods, using as training data the observations measured from 24 pairs of images from different datasets, shown on page 88. Ground truth labellings for the data are obtained using the automatic annotation setup described in Chapter 5. The setup of detectors, descriptors and dissimilarities is identical to that used for the final experiments. It is described in Section 6.1.1.

Note that we generally apply the preselection of putative matches as described in Section 2.4 before inferring empirical distributions from the data.

4.3.1 Dependence of Energy Potentials on the Feature Type

We will derive the likelihoods and priors separately for each type of feature, dissimilarity measure, and spatial relationship occurring in our setup. Therefore the energy potentials do not necessarily have all the same characteristics. For example, the dissimilarity measure for straight line segments that we use in our experiments is significantly weaker than that for blobs and junctions. Consequently the likelihood $p(s_n \mid l_n)$ has a significantly different shape depending on whether v_n is a line segment or a junction feature, for example.

As a consequence, the likelihood distributions depend formally on the type of $\underline{\text{f}}$eature, type of $\underline{\text{d}}$escriptor, and type of dissimilarity $\underline{\text{m}}$easure associated with a putative match, which we may denote by λ^F, λ^D and λ^M, respectively. If we use a tuple $\boldsymbol{\lambda_n} = (\lambda_n^F, \lambda_n^D, \lambda_n^M)$ for identifying the exact type of a putative match v_n, the probabilities must therefore actually read $p_n(s_n \mid l_n, \boldsymbol{\lambda}_n)$ and $p_{nm}(t_{nm} \mid l_n, l_m, \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_m)$.

In our particular setup, a feature type λ^F is always combined with the same descriptor type λ^D and dissimilarity measure λ^M, so that each $\boldsymbol{\lambda}_n = (\lambda_n^F, \lambda_n^D, \lambda_n^M)$ is uniquely determined by λ_n^F. Therefore we define the four symbols $\boldsymbol{\lambda}_{\text{Segment}}$, $\boldsymbol{\lambda}_{\text{Blob}}$, $\boldsymbol{\lambda}_{\text{Affine}}$ and $\boldsymbol{\lambda}_{\text{Junction}}$ as a shorthand notation for the particular settings related to line segments, blob features, affine region features and junction features, as described in Section 6.1.1 on page 69.

4.3.2 Prior Probabilities

As described in Section 2.4, we use different maximum ranks of descriptor dissimilarity per feature type for selecting an initial set of putative matches. This should naturally lead to different prior probabilities for different feature types.

Observe the relative frequency of good and bad putative matches for each feature type in Figure 4.13. The relative amount of inliers for the line segments, where the $k = 3$ best correspondences per feature were chosen, is less than 30%. This is very reasonable, as we have to expect 66% outliers caused by the preselection, and some more caused by the typical shortcomings of descriptor-based assignment. The same holds for the other feature types as well. For the junction and blob features, only the best match has been selected, leading to amounts of outliers significantly below 20%, even below 10% for the junctions. The preselection for affine blob features selected the two best matches, so that the amount of outliers here is between 30 and 40%.

The prior $\widehat{P}\left(l_n \mid \boldsymbol{\lambda}_n\right)$ is not used explicitly in the minimization function (4.32). Instead, an estimate $\widehat{P}\left(l_n, l_m \mid \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_m\right)$ for the prior probability of a labelling of pairs of putative matches v_n, v_m with feature types $\boldsymbol{\lambda}_n, \boldsymbol{\lambda}_m$ is required. As we have to distinguish all possible combinations of feature types, we obtain 16 different plots.

The estimated prior probabilities for pairs of matches, where the first match is a line segment, are shown in Figure 4.14. Again, we see the strong influence of the different preselection criteria per feature type on the prior: For pairs containing one line segment match and one match of another type, it is most likely that the line segment match is an outlier.

If we consider the situation for blob feature matches combined with other feature types (Figure 4.15), the effect is very different. In particular, except for a combination with line segments, the most frequently observed event is that of having two inliers. The effect for junction matches (Figure 4.17) and affine blobs (Figure 4.16) is very similar.

4.3.3 Dissimilarity of Feature Descriptors

For determining the dissimilarity of line segment descriptors, we use the color profile distance measure proposed by Bay et al. (2005), as described in Section 2.3.

We normalize the distance measure by its theoretical maximum value, which is determined as follows. The theoretical maximum length of the difference vectors $(\boldsymbol{h}_1 - \boldsymbol{h}_2)$ (cf. Section 2.3) is $\sqrt{2}$, because the histograms are unit vectors. For the value of the distance $d_{1,2}$ in Eq. (2.1) we therefore have

$$d_{1,2} = (\boldsymbol{h}_1 - \boldsymbol{h}_2)^{\mathsf{T}} \mathsf{A} (\boldsymbol{h}_1 - \boldsymbol{h}_2) < (\boldsymbol{h}_1 - \boldsymbol{h}_2)^{\mathsf{T}} (\boldsymbol{h}_1 - \boldsymbol{h}_2) \leq 2 \ , \tag{4.58}$$

using the fact that $0 \leq A_{ij} \leq 1$ for all elements A_{ij} of A. As the dissimilarity of two descriptors is the square root of the mean of the two distances corresponding to the left and right sides of the segment, the final dissimilarity measure has an upper bound of $\sqrt{(2+2)/2} = \sqrt{2}$.

The dissimilarity of SIFT descriptors is simply defined as the Euclidean distance of two descriptor vectors. Again, we normalize by the maximum theoretical value. As the descriptors are vectors of length 128, with coefficients in the range $(0, 255)$, the largest possible Euclidean distance is the length of the diagonal axis in the corresponding hypercube, which is $\sqrt{128 \cdot 255^2} = 2885$. The values observed in practice will be significantly smaller, so we only expect to see normalized dissimilarities significantly below 0.5.

Now take a look at the normalized histogram on top of Figure 4.18. It shows the dissimilarities of good $(l_n = 1)$ and bad $(l_n = 0)$ blob feature correspondences, referring to Euclidean distances of SIFT descriptors. Due to the normalization, the histogram shapes can be reasonably approximated by a Beta distribution

$$\mathrm{Beta}(s_n|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} s_n^{a-1} (1 - s_n)^{b-1} \ , \tag{4.59}$$

which is defined by two parameters (a,b), and based on the gamma function

$$\Gamma(a) = \int_0^\infty u^{a-1} e^{-u} du \ . \tag{4.60}$$

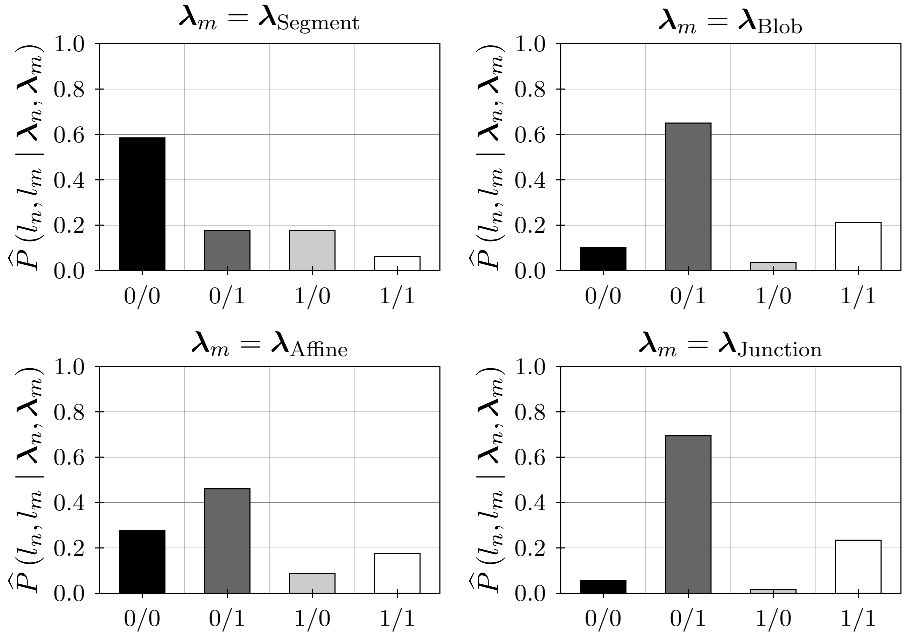


FIGURE 4.14: Empirical fraction of pairs of putative matches, where the first match refers to straight line segments ($\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_{\text{Segment}}$) for different labelings l_n, l_m as observed on the training data. We use these as priors $\widehat{P}\,(l_n, l_m \mid \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_m)$ for the different feature types.

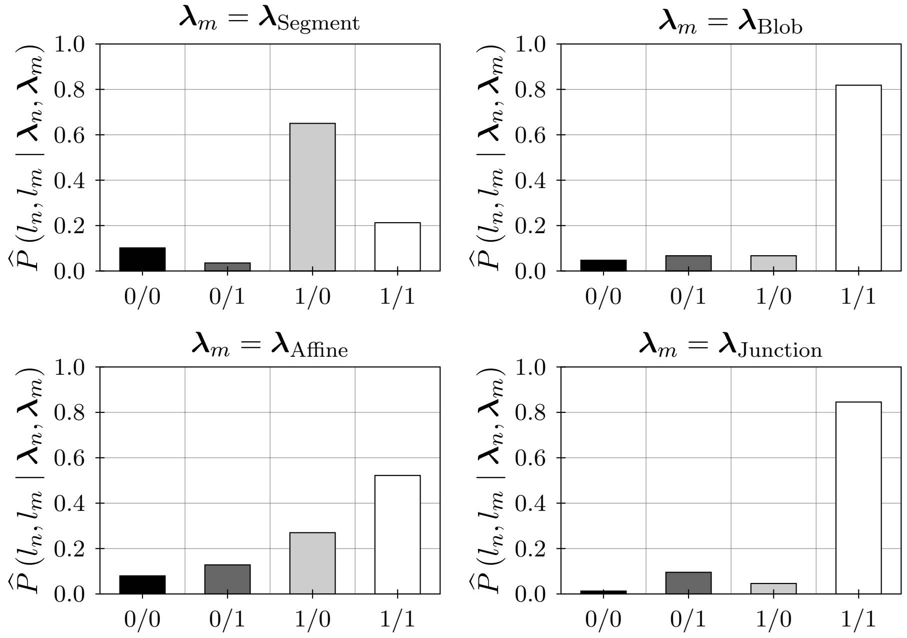


FIGURE 4.15: Empirical fraction of pairs of putative matches, where the first match refers to blob features ($\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_{\text{Blob}}$) for different labelings l_n, l_m, as observed on the training data.

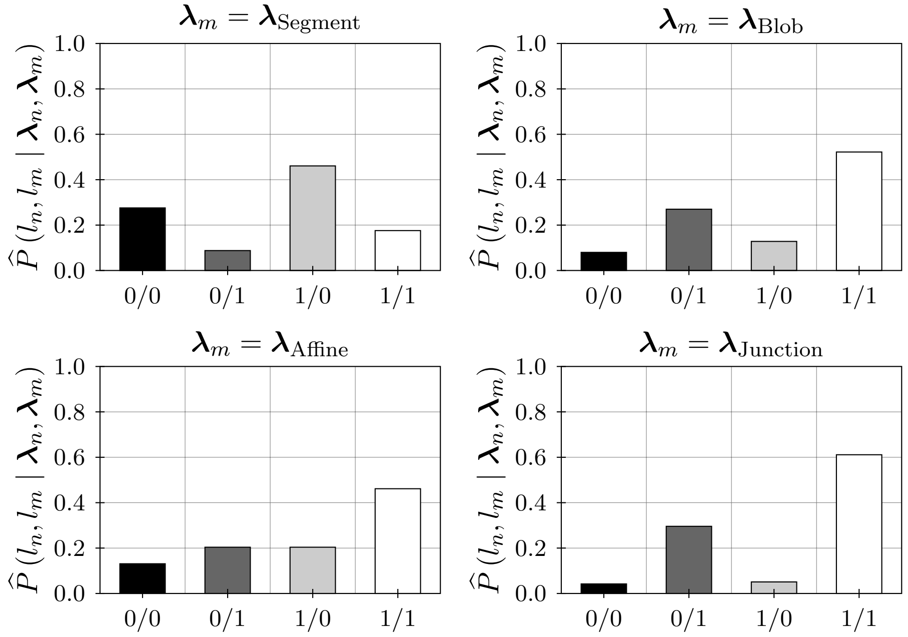


FIGURE 4.16: Empirical fraction of pairs of putative matches, where the first match refers to affine regions ($\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_{\text{Affine}}$) for different labelings l_n, l_m, as observed on the training data.

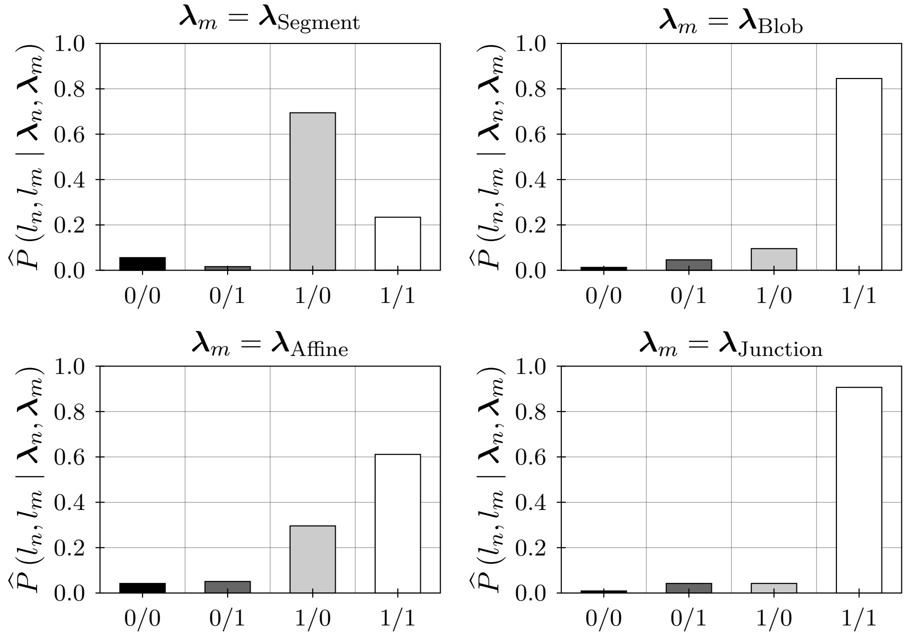


FIGURE 4.17: Empirical fraction of pairs of putative matches, where the first match refers to junction features ($\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_{\text{Junction}}$) for different labelings l_n, l_m, as observed on the training data.

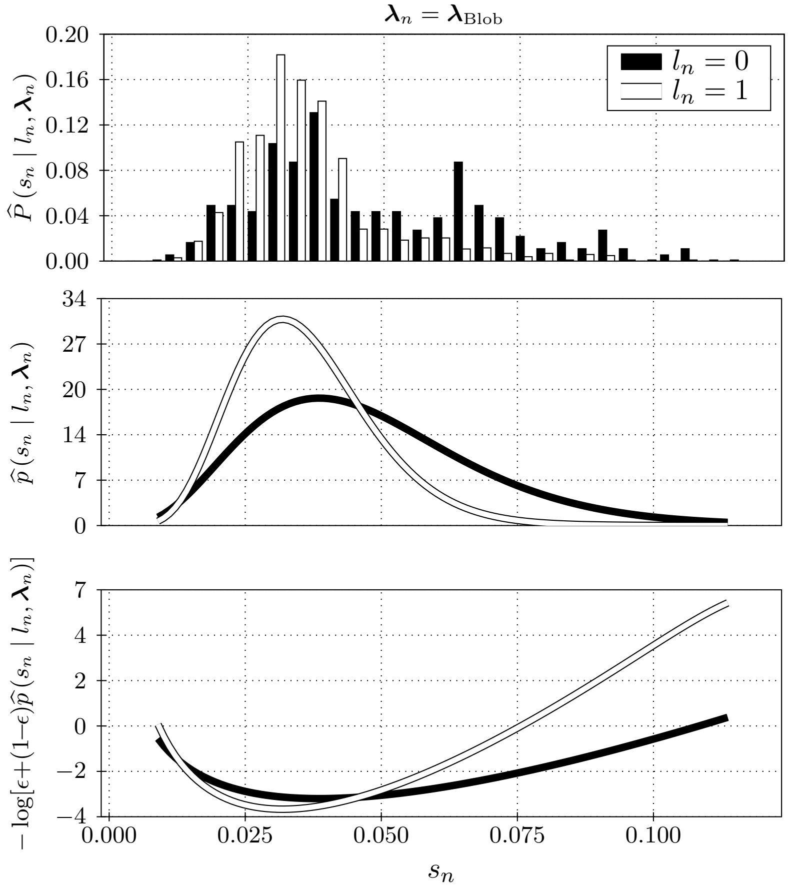


FIGURE 4.18: *Top:* Normalized histograms of dissimilarities s_n for good ($l_n = 1$) and bad ($l_n = 0$) blob feature correspondences. *Middle:* Beta distributions estimated from the histogram, used as a parametric approximation $\widehat{p}(s_n \mid l_n, \boldsymbol{\lambda}_n)$ of the likelihood function. *Bottom:* Bounded negative log likelihood derived from $\widehat{p}(s_n \mid l_n, \boldsymbol{\lambda}_n)$, which we use for the energy potentials. The observations refer to the training dataset shown on page 88. Note that the theoretical range of the observations is $(0, 1)$, and that the Beta distribution is defined over the range $[0, 1]$. Here we only plot the relevant range; the densities are practically zero above $s_n \simeq 0.125$.

We estimate the parameters (a, b) from the training data separately for the inlier and outlier distributions to obtain estimates for the class conditional likelihood functions $\widehat{p}(s_n \mid f_n = 0, \boldsymbol{\lambda}_n)$ and $\widehat{p}(s_n \mid f_n = 1, \boldsymbol{\lambda}_n)$, as shown in the middle of Figure 4.18 for the blob features. We will refer to the Beta distribution parameters corresponding to the event $(v_n = 1)$, as (a_1, b_1), and use (a_0, b_0) for the event $(v_n = 0)$. Note that such two distributions often have two intersection points in the interval $[0, 1]$.[8]

The negative log likelihood $-\log \widehat{p}(s_n \mid l_n, \boldsymbol{\lambda}_n)$ that we actually use in the energy function (4.32) is shown in the bottom plot of Figure 4.18. Note that we introduce a bound on the log likelihood by using

$$-\log[\epsilon + (1 - \epsilon)\widehat{p}(s_n \mid l_n, \boldsymbol{\lambda}_n)] \tag{4.61}$$

with a small threshold $\epsilon = 0.001$. In practice, the bound only affects values s_n very close to the limits of the domain $[0, 1]$. Imposing this bound has a similar effect as adding or subtracting a value in the order of machine accuracy to observations near 0 and 1, respectively. Such observations occur very rarely in practice.

The estimates for normalized dissimilarities of line segment matches, as described in Sections 2.3 and 4.3.3, are shown in Figure 4.19. We use a beta distribution as well here, but the approximation is less accurate as for other feature types: For the good correspondences $(l_n = 1)$, the shape of the Beta distribution differs from the histogram. This is the only severe deviation of a parametric approximation from the histograms in our setup. To keep the framework simple, we refrain from choosing a more complicated model, but emphasize that it would be interesting to investigate the effect of closer approximations onto the matching results.

4.3.4 Construction of uncertain points and lines from image features

For deriving spatial relationships between features, we assume that we can always construct the normalized 2D homogeneous point

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{4.62}$$

with covariance matrix Σ_{xx} representing the position of an image feature $\boldsymbol{p}_i$. For point-like features, we can fall back to

$$\Sigma_{xx} = \sigma_x^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{4.63}$$

where σ_x corresponds to the expected localization accuracy in pixel, e.g. $\sigma_x \simeq 0.3$. Some detectors however, especially those based on the structure tensor, often provide a direct estimate for Σ_{xx} with full correlation information. For line segments, we will use the midpoint for constructing $\mathbf{x}_i$, which usually has a strong localization error along the line, and a small error perpendicular to it. Here the covariance matrix of the line and its midpoint can be computed using classical error propagation, for example starting from start-/endpoints with known localization accuracy, as described in Meidow et al. (2009).

In a similar manner, we assume that an uncertain 2D homogeneous line

$$\mathbf{l}_i = \pm \begin{bmatrix} \cos \alpha_i \\ \sin \alpha_i \\ -d \end{bmatrix} \tag{4.64}$$

[8] These Beta distributions express the conditional probability densities $p(s_n \mid l_n, \boldsymbol{\lambda}_n)$ for the observations given labels $l_n \in 0, 1$. Such unary compatibility functions are often referred to as the *evidence* for l_n.

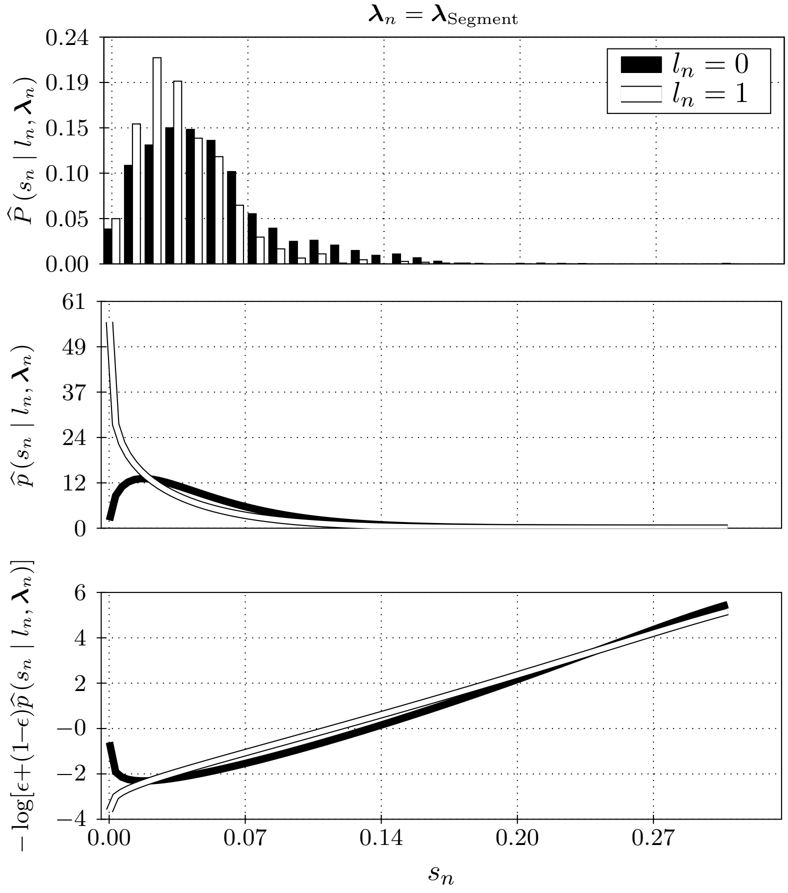


FIGURE 4.19: *Top:* Normalized histograms of dissimilarities s_n for good ($l_n = 1$) and bad ($l_n = 0$) line segment correspondences. *Middle:* Beta distributions estimated from the histogram, used as a parametric approximation $\widehat{p}(s_n \mid l_n, \boldsymbol{\lambda}_n)$ of the likelihood function. *Bottom:* Bounded negative log likelihood derived from $\widehat{p}(s_n \mid l_n, \boldsymbol{\lambda}_n)$, which we use for the energy potentials. The observations refer to the training dataset shown on page 88. Note that the theoretical range of the observations is $(0, 1)$, and that the Beta distribution is defined over the range $[0, 1]$. Here we only plot the relevant range; the densities are practically zero above $s_n \simeq 0.27$.

with covariance matrix Σ_{ll} can be constructed from each feature $\boldsymbol{p}_i$. For point-like features, we use the centroid representation of straight line segments (Meidow et al., 2009, 3.1.2), where the centroid is the image location of the original point feature, and the direction is identified with the dominant gradient orientation within the local patch, as stored in the SIFT descriptor. According to Lowe (2004, Sec. 5), we must hence expect the direction of lines constructed from point-like features to have a standard deviation of about three degrees. For straight line segments, the uncertain homogeneous line is converted from other representations as described in Meidow et al. (2009). If not otherwise available, a reasonable estimate for the covariance matrix can be obtained by assuming the covariance matrices of the start-/endpoints to have the structure (4.63) and performing error propagation accordingly.

Operator notation. Using the conversions described above, we define explicit operators

$$\mathbf{x}(\boldsymbol{p}_i) \quad \text{and} \quad \mathbf{l}(\boldsymbol{p}_i)\ , \tag{4.65}$$

which return the uncertain homogeneous 2D point or line representation for a feature $\boldsymbol{p}_i$. In the same manner we use

$$\Sigma_{xx}(\boldsymbol{p}_i) \quad \text{and} \quad \Sigma_{ll}(\boldsymbol{p}_i) \tag{4.66}$$

for constructing the corresponding covariance matrices.

4.3.5 Consistency of Pairwise Sidedness

The information whether a feature is located left or right of another feature referring to its orientation is known to be a stable cue of information for architectural scenes with mostly planar substructure and low amount of occlusions (Bay et al., 2005). Given two putative matches $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$ and $v_m = (\boldsymbol{p}'_k, \boldsymbol{p}''_l)$, the idea is to check in image $\mathcal{I}'$ whether the position $(x_k, y_k)'$ of $\boldsymbol{p}'_k$ is left or right of $\boldsymbol{p}'_i$ according to its orientation α'_i. The sidedness should be consistent with that in image $\mathcal{I}''$, using $\boldsymbol{p}''_j$ and $\boldsymbol{p}''_l$ accordingly. The sidedness relation is quite stable if no 3D occlusions are present in the scene, if the projective mapping is straight line preserving, and if the surfaces are rather flat. Nevertheless we emphasize again that all spatial relationships discussed here have limited validity (cf. Figure 3.1). This is why we will treat them in a Bayesian manner instead of using them as hard constraints.

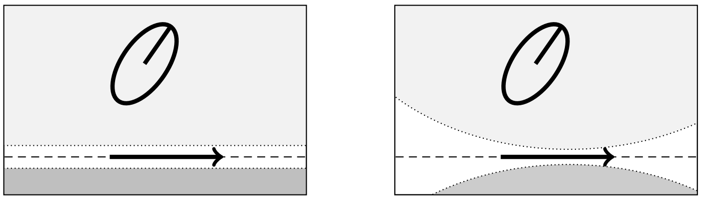


FIGURE 4.20: Illustration of possible representations for the sidedness relation between two oriented features. The brightly shaded area denotes the region where a location in the image is considered "left of" the line segment, the darker shaded area the region "right of". *Left:* Simple relation as used in Bay et al. (2005). This test applies a fixed threshold of a few pixels for taking collinear features into account (white region). It hereby neglects the uncertainty of the feature's orientation, which is usually significant for point features and "short" line segments. *Right:* Relation based on a statistical test for the incidence of an uncertain 2D point with an uncertain 2D line. The collinearity region where the test is skipped (white) is bounded by a hyperbolic shape, taking the uncertainty of the feature's orientation into account. Note that both illustrations are slightly exaggerated w.r.t. the confidence regions to illustrate the effect.

To evaluate the sidedness of two features in one image, we apply the conversion operators (4.65) and (4.66) to measure the signed distance

$$d' = d(\boldsymbol{p}'_i, \boldsymbol{p}'_k) = \mathbf{x}(\boldsymbol{p}'_i)^\mathsf{T}\mathbf{l}(\boldsymbol{p}'_k) \tag{4.67}$$

in image $\mathcal{I}'$ via the scalar product, which is normally distributed with variance

$$\sigma'_{dd} = \mathbf{x}^\mathsf{T}(\boldsymbol{p}'_i)\Sigma_{ll}(\boldsymbol{p}'_k)\mathbf{x}(\boldsymbol{p}'_i) + \mathbf{l}^\mathsf{T}(\boldsymbol{p}'_k)\Sigma_{xx}(\boldsymbol{p}'_i)\mathbf{l}(\boldsymbol{p}'_k)\ . \tag{4.68}$$

In a similar manner, we obtain (d'', σ''_{dd}) for corresponding features $(\boldsymbol{p}''_j, \boldsymbol{p}''_l)$ in $\mathcal{I}''$.

For avoiding unstable tests, one has to distinguish the case when two features are collinear in at least one of the views. A common way to achieve this is to introduce a minimum distance T_d of a few pixels. If $\min(d', d'') < T_d$, the test is not evaluated for this pair of correspondences. This realization of the sidedness test is illustrated in the top row of Figure 4.20, but it is not sufficient for our problem for two reasons:

1. It assumes negligible error σ_α referring to the feature's orientation. For point features however, where the orientation is computed from histograms of gradient orientations, it is known that on average $\sigma_\alpha \sim 4°$ (Lowe, 2004). For short line segments, significant errors may also occur.

2. By using a fixed threshold $\sigma_{(x_i, y_i)}$ of a few pixels, the significantly varying localization accuracies of different feature types, detection scales, and texture properties are ignored. This is not acceptable for our purpose, as we want to combine feature types with very different properties.

Instead we want to derive a test which properly takes the accuracy of a feature's orientation and image position localization into account. We therefore define collinearity of $\boldsymbol{p}'_i$ and $\boldsymbol{p}'_k$ by the positive outcome of the statistical test that $\mathbf{x}(\boldsymbol{p}'_i)$ is incident with $\mathbf{l}(\boldsymbol{p}'_k)$. For the features in $\mathcal{I}'$, this is the case if

$$\frac{|d'|}{\sqrt{\sigma'_d}} < \Phi^{-1}(S)\ , \tag{4.69}$$

where Φ is the normal cdf and the parameter S defines the acceptance region and is usually set to a probability near 1, e.g. $S = 0.99$.

We end up with the following scheme for determining whether the sidedness between two putative feature matches $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$ and $v_m = (\boldsymbol{p}'_k, \boldsymbol{p}''_l)$ is inconsistent across the two images:

1. We compute the signed distances (d', σ'_d) and (d'', σ''_d), where $d' = d(\boldsymbol{p}'_i, \boldsymbol{p}'_k)$ and $d'' = d(\boldsymbol{p}''_j, \boldsymbol{p}''_l)$.

2. If the incidence relation (4.69) is fulfilled in one of the views, the test is skipped for this pair of correspondences.[9]

3. Otherwise, we return the test result

$$t^s_{nm} = t^s(v_n, v_m) = t^s(\boldsymbol{p}'_i, \boldsymbol{p}''_j, \boldsymbol{p}'_k, \boldsymbol{p}''_l) = \begin{cases} 1, & \text{if } \operatorname{sign}(d') \neq \operatorname{sign}(d'') \\ 0, & \text{otherwise} \end{cases}\ . \tag{4.70}$$

The sidedness measurements that we hereby obtain are binary observations $t^s_{nm} \in \{0, 1\}$.

[9] It is also possible to include incidence as another form of sidedness, and let the test pass in case that the features in both views are collinear. We have not implemented this variant.

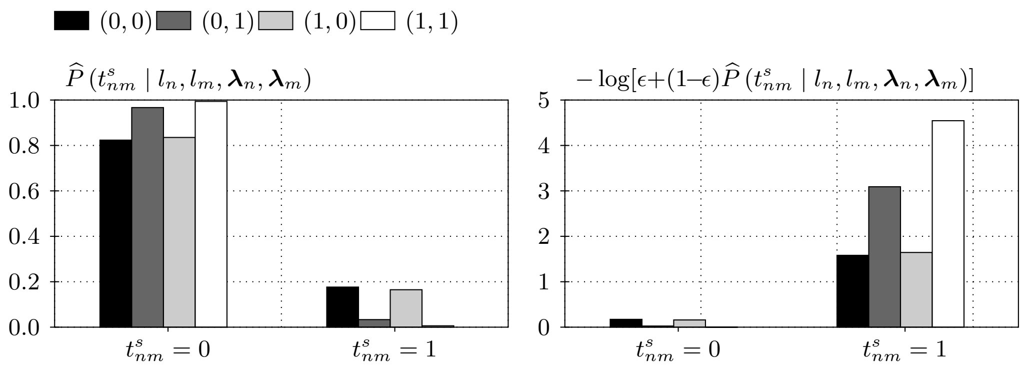


FIGURE 4.21: *Left:* Normalized histograms of discrete observations t^s_{nm} between blob and affine blob feature correspondences ($\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_{\text{Blob}}, \boldsymbol{\lambda}_m = \boldsymbol{\lambda}_{\text{Affine}}$), referring to inconsistency of pairwise sidedness. We obtain estimates for each of the events ($l_n = 0, l_m = 0$), ($l_n = 1, l_m = 0$), ($l_n = 0, l_m = 1$) and ($l_n = 1, l_m = 1$). *Right:* Negative log likelihood derived from the histograms, which we use for the energy potentials. The observations refer to the training dataset shown on page 88.

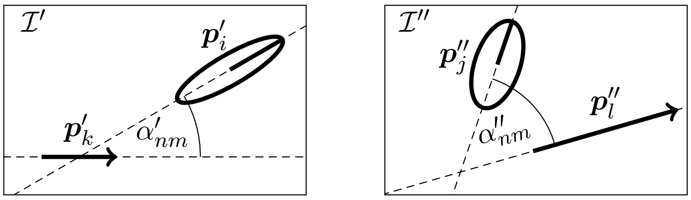


FIGURE 4.22: Illustration of the setup for computing the pairwise orientation difference $t^\alpha_{nm} = \min(|\alpha'_{nm} - \alpha''_{nm}|, 2\pi - |\alpha'_{nm} - \alpha''_{nm}|)$ for two matches $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$ and $v_m = (\boldsymbol{p}'_k, \boldsymbol{p}''_l)$.

The histogram for the observations on the training dataset, referring to pairs containing a blob and affine region feature correspondence, is shown on top of Figure 4.21. We see that observing consistent sidedness motivates to select both matches as an inlier, as $\widehat{P}\,(t^s_{nm} = 0 \mid l_n, l_m, \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_m) \simeq 1$ for $l_n = 1$ and $l_m = 1$. This is remarkable, as it shows that the statistical model in fact generates effects similar to the explicit boosting stage proposed by Bay et al. (2005): Strong feature types (here: blobs) will "boost" correspondences of weak feature types (here: affine regions). In our setup, this effect is strongest for pairs of a straight line and a blob feature correspondence.

4.3.6 Consistency of Angles between Oriented Features

Besides sidedness, we also evaluate the angle between two oriented features in one view, and compare it to the angle between their corresponding features in another view. We assume that the difference between these two angles is rather small for valid pairs of correspondences, so that large differences indicate outliers.

The principle is illustrated in Figure 4.22: For two putative matches $v_n = (\boldsymbol{p}'_i, \boldsymbol{p}''_j)$ and $v_m = (\boldsymbol{p}'_k, \boldsymbol{p}''_l)$, we compute the enclosing angle $\alpha'_{nm} \in (0, 2\pi)$ between the features $\boldsymbol{p}'_i, \boldsymbol{p}'_k$ involved in image $\mathcal{I}'$ based on their arbitrarily scaled direction vectors

$$\boldsymbol{l}(\boldsymbol{p}'_i) = \boldsymbol{l}'_i = |\boldsymbol{l}'_i| \begin{bmatrix} \cos(\alpha'_i) \\ \sin(\alpha'_i) \end{bmatrix} \quad \text{and} \quad \boldsymbol{l}(\boldsymbol{p}'_k) = \boldsymbol{l}'_k = |\boldsymbol{l}'_k| \begin{bmatrix} \cos(\alpha'_k) \\ \sin(\alpha'_k) \end{bmatrix} , \tag{4.71}$$

using the numerically robust two-parameter form of the Arcustangens

$$\alpha'_{nm} = \alpha'(v_n, v_m) = \operatorname{atan2}\left(\boldsymbol{l}'_{i2}, \boldsymbol{l}'_{i1}\right) - \operatorname{atan2}\left(\boldsymbol{l}'_{k2}, \boldsymbol{l}'_{k1}\right) \quad \bmod 2\pi\ . \tag{4.72}$$

The angle α''_{nm} spanned by the two features in image $\mathcal{I}''$ is computed accordingly. The directions $\boldsymbol{l}(\boldsymbol{p}_i)$ are constructed in a similar way as the homogeneous representations $\mathbf{l}(\boldsymbol{p}_i)$ described in Section 4.3.4.

The difference of the angles spanned in the two images is then given by

$$\begin{aligned} t^{\alpha}_{nm} &= t^{\alpha}(v_n, v_m) \\ &= \min(|\alpha'_{nm} - \alpha''_{nm}|, 2\pi - |\alpha'_{nm} - \alpha''_{nm}|), \quad t^{\alpha}_{nm} \in (0, \pi) \ . \end{aligned} \tag{4.73}$$

It is obvious that we can neither expect angles between pairs of correct matches to be always equal, nor angles between outliers to be always largely different. When investigating the empirical distribution of the consistency measures t^{α} on our training dataset, we see that they carry valuable information for our problem though. The distribution for pairs of blob and junction feature matches is shown in the top row of Figure 4.23.

The distribution indicates that for small inconsistencies t^{α} between feature correspondences of this type, it is most likely that both matches are inliers, referring to this observation only. With increasing inconsistency, it becomes more probable that the blob correspondence is an outlier, until for very high inconsistencies the labeling $(0, 0)$ is motivated, which means that both correspondences are likely to be outliers. This corresponds strongly to our initial assumptions. Similar observations can be made for other combinations of feature types, and again we see stronger feature types motivating the selection of weaker ones when the angular consistency is high.

In order to take the uncertainty of feature orientations into account, (4.73) should actually use a proper test statistic, so that the consistency becomes

$$t^{\alpha_0}_{nm} = t^{\alpha_0}(v_n, v_m) = \min(|\alpha_0(v_n, v_m)|, 2\pi - |\alpha_0(v_n, v_m)|) \ ,$$

using the normalized test statistic

$$\alpha_0(v_n, v_m) = \frac{\alpha'_{nm} - \alpha''_{nm}}{\sqrt{\sigma^2_{\alpha_{nm}}{}' + \sigma^2_{\alpha_{nm}}{}''}} \ . \tag{4.74}$$

The variances of the angles would then be computed from the covariance matrices $\Sigma'_{l_i}, \Sigma'_{l_j}, \ldots$ of the uncertain direction vectors by error propagation. Given that the vectors are already spherically normalized, we obtain

$$\sigma^2_{\alpha_{nm}}{}' = \mathsf{J}_i^{\mathsf{T}} \Sigma'_{l_j} \mathsf{J}_i + \mathsf{J}_j^{\mathsf{T}} \Sigma'_{l_i} \mathsf{J}_j \tag{4.75}$$

for the angle α'_{nm}, using the Jacobians

$$\mathsf{J}_{i/j} = \left[-\sin\phi_{i/j}, \cos\phi_{i/j}, 0\right]^{\mathsf{T}} \ . \tag{4.76}$$

Note that although we use the simple version (4.73) in our implementation, more accurate results can be expected when applying (4.74) instead.

4.3.7 Consistency of Pairwise Spatial Distance

If two features are located close to each other in one view, we also expect their correspondences in another view to be close. This simple reasoning based on proximity was already suggested by Ullman (1979). We choose to measure the distance between two feature locations, and compare it to the distance of the two corresponding features in the second image. For line segments, we measure the distance based on its midpoint. Note that it is not reasonable to

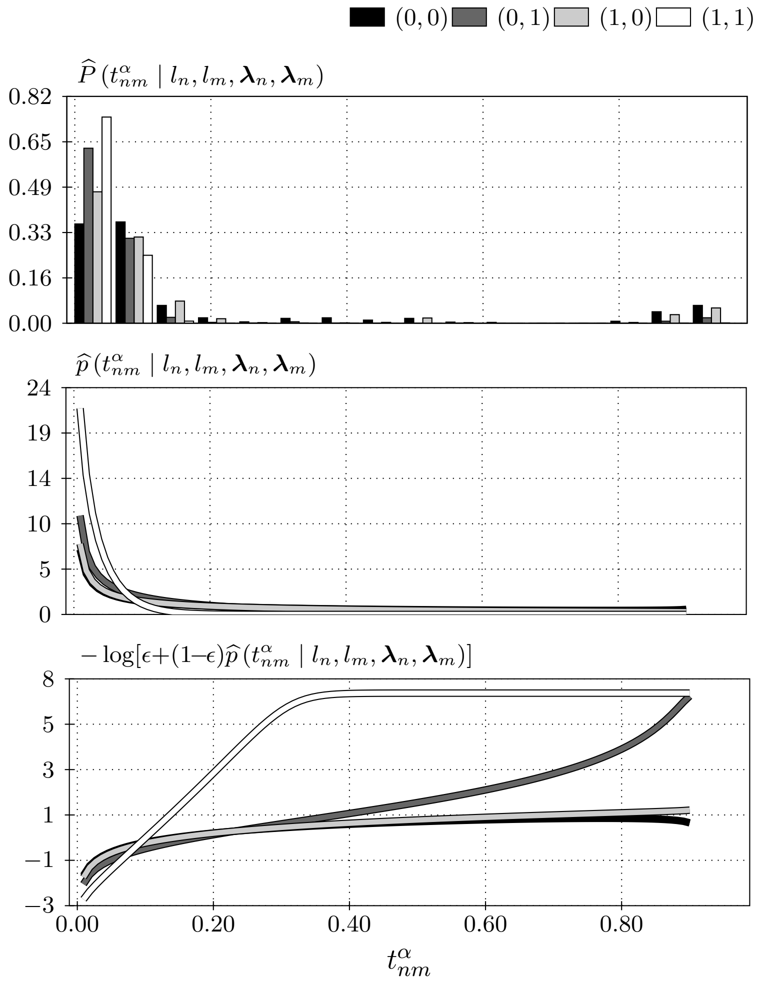


FIGURE 4.23: *Top:* Normalized histograms of observations t^{α}_{nm} between blob and junction feature correspondences ($\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_{\text{Blob}}, \boldsymbol{\lambda}_m = \boldsymbol{\lambda}_{\text{Junction}}$), denoting inconsistency of angles between pairs of oriented features. We obtain four distributions, referring to the events ($l_n = 0, l_m = 0$), ($l_n = 1, l_m = 0$), ($l_n = 0, l_m = 1$) and ($l_n = 1, l_m = 1$). *Middle:* Beta distributions estimated from the histogram, used as an estimate for the likelihood $p(t^{\alpha}_{nm} \mid l_n, l_m, \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_m)$. *Bottom:* Bounded negative log likelihood derived from $\widehat{p}\,(t^{\alpha}_{nm} \mid l_n, l_m, \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_m)$, which we use for the energy potentials. The observations refer to the training dataset shown on page 88. Note that the theoretical range of the observations is $(0, 1)$, and that the Beta distribution is defined over the range $[0, 1]$. Here we only plot the range of values that we observed on the training dataset.

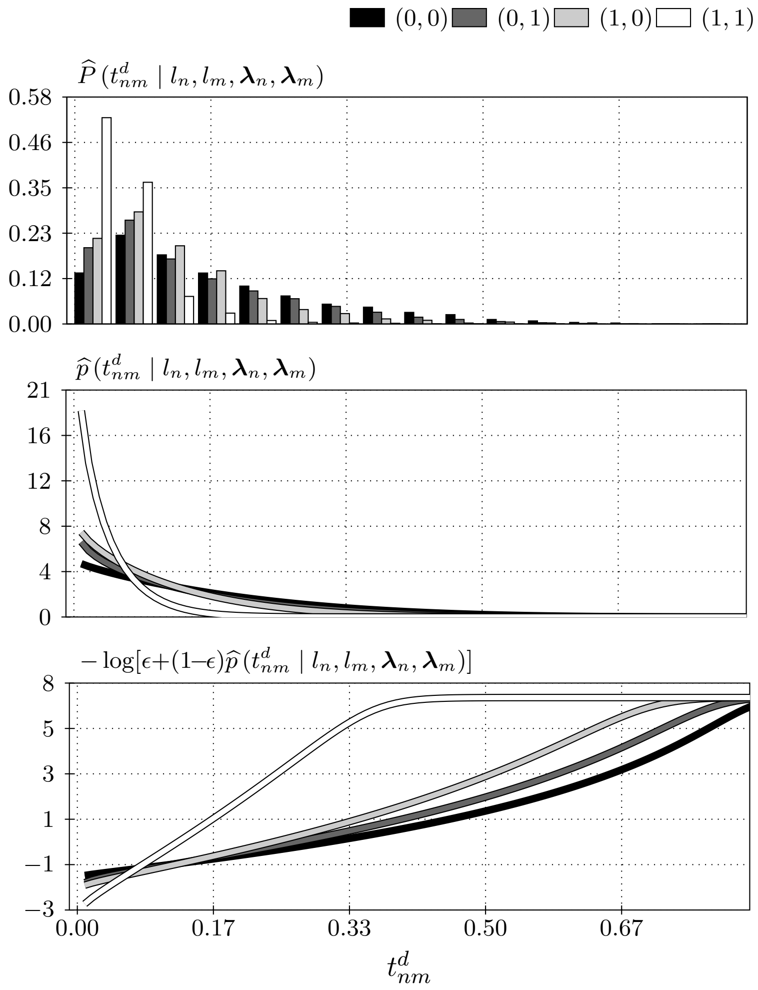


FIGURE 4.24: *Top:* Normalized histograms of observations t^d_{nm} between blob and junction feature correspondences ($\boldsymbol{\lambda}_n = \boldsymbol{\lambda}_{\text{Blob}}, \boldsymbol{\lambda}_m = \boldsymbol{\lambda}_{\text{Junction}}$), denoting inconsistency of spatial distance between pairs of oriented features. We obtain four distributions, referring to the events ($l_n = 0, l_m = 0$), ($l_n = 1, l_m = 0$), ($l_n = 0, l_m = 1$) and ($l_n = 1, l_m = 1$). *Middle:* Beta distributions estimated from the histogram, used as an estimate for the likelihood $p(t^d_{nm} \mid l_n, l_m, \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_m)$. *Bottom:* Bounded negative log likelihood derived from $\widehat{p}\,(t^d_{nm} \mid l_n, l_m, \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_m)$, which we use for the energy potentials. The observations refer to the training dataset shown on page 88. Note that the theoretical range of the observations is $(0, 1)$, and that the Beta distribution is defined over the range $[0, 1]$. Here we only plot the range of values that we observed on the training dataset.

use the distance in pixels for this purpose, as we have to take images with different resolutions into account. Therefore we normalize the distances by the length of the image diagonal.

The inconsistency of pairwise spatial distance for two correspondences v_n, v_m is defined as

$$\begin{aligned} t^d_{nm} &= t^d(v_n, v_m) = t^d(\boldsymbol{p}'_i, \boldsymbol{p}''_j, \boldsymbol{p}'_k, \boldsymbol{p}''_l) \\ &= \frac{|\mathbf{x}(\boldsymbol{p}'_i) - \mathbf{x}(\boldsymbol{p}'_k)|}{\sqrt{(N'_x)^2 + (N'_y)^2}} - \frac{|\mathbf{x}(\boldsymbol{p}''_j) - \mathbf{x}(\boldsymbol{p}''_l)|}{\sqrt{(N''_x)^2 + (N''_y)^2}} \;, \end{aligned} \tag{4.77}$$

using again the operator in Eq. (4.65), and the vertical and horizontal dimensions N'_x, N'_y of an image $\mathcal{I}'$ in pixels. The empirical distribution and estimated likelihood functions, again based on a Beta distribution, are shown in Figure 4.24 for pairs of line segment and junction feature correspondences. The effects are very similar to those described for the t^α observations.

Just as for the angular consistencies t^α, one may gain an additional benefit when replacing the Euclidean distances with the proper test statistic, i.e. by normalizing the distances with their standard deviations, which we did not realize for our experiments.

4.3.8 Dealing with Redundant Correspondences

With *redundant correspondences* we denote feature matches that refer to the same feature in one view. For example, the correspondences v_1 and v_5 shown in Figure 4.2 (page 31) both refer to feature $\boldsymbol{p}'_1$ in $\mathcal{P}'$. Due to the preselection scheme described in Section 2.4, such situations occur frequently within the set $\mathcal{V}$ of putative matches. Redundant correspondences are explicitly suppressed in most algorithms for wide baseline stereo matching, in the spirit of Ullman's *exclusion* criterion (Ullman, 1979). The suppression is usually referred to as "enforcing the uniqueness constraint". For example, Torresani et al. (2008) exclude redundant correspondences explicitly from the feasible set of the optimization problem.

By contrast, we accept redundant correspondences even in the final result, for reasons that we explain in the following. First of all, even the correct solution may contain redundant correspondences. A line segment detector, for example, usually has an internal threshold for merging neighboring pixels with a similar edge response into segments. Depending on the image noise, the merge process can easily lead to different results in two overlapping images, as shown in Figure 4.25. If the same line segment has been merged completely in one view, but only partially in the other, it will therefore be involved in two or more correct matches.

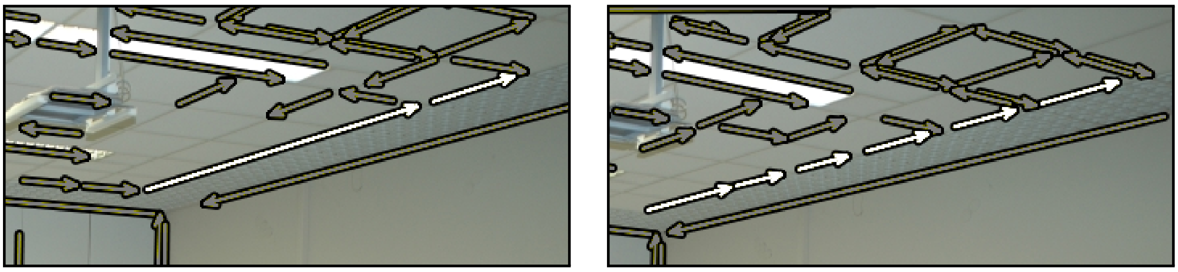


FIGURE 4.25: Line segments detected in two views of an indoor scene. Observe how the same line in 3D (white arrows) is represented by two line segments in the left, and six segments in the right view.

For point-like features, correct redundant correspondences can be caused by multiple characteristic orientations or scales, leading to multiple features with identical position in the image (cf. Sec. 2.3).

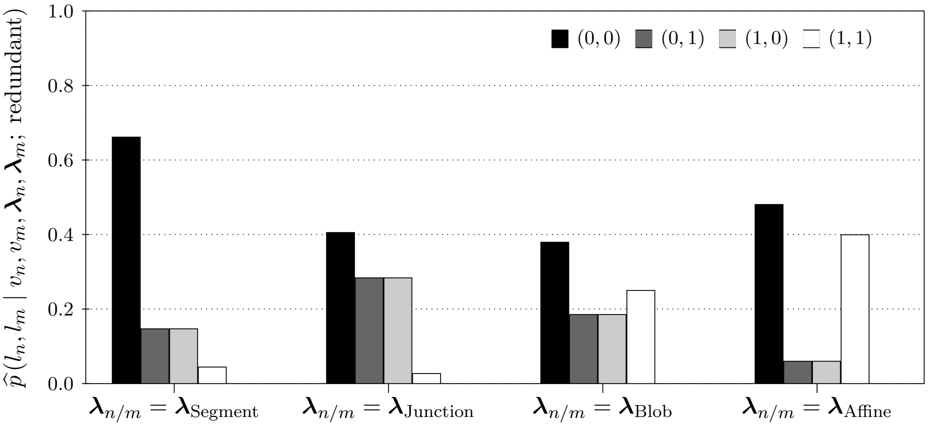


FIGURE 4.26: Relative frequency of labels (l_n, l_m) for redundant pairs of matches (Section 4.3.8). Only pairs of matches having the same feature type can be redundant. We see that for junction features, a redundant pair with both matches being inliers is very rare, other than for blob features. A possible explanation for this observation is that duplicate features with multiple dominant orientations (Lowe, 2004) arise more often for blobs. It is important to note that none of the geometric compatibility measures can be computed for a redundant pair. Therefore we use these priors as a replacement for the binary potentials when encountering a redundant pair.

The second reason for not explicitly suppressing redundant matches is the natural limitation of the matching process. We cannot make an ultimately correct decision about the matching problem on the basis of 2D information. We only want to generate a set of correspondences that constitutes a good input for the subsequent application, naturally including a certain amount of outliers. In our case, the correspondences are used for estimating camera geometries. There, a robust RANSAC scheme for estimating the pairwise epipolar geometries is often applied, which can deal well with an outlier amount of 40%. Instead of suppressing many otherwise promising correspondences in order to fulfill the uniqueness constraint, we intentionally accept a certain amount of redundant correspondences.

Redundant feature matches cannot be processed in the same way as other correspondences, because the spatial relationships in one view refer to one and the same feature and are therefore not meaningful.

To compensate for the missing observations referring to geometric consistency for such pairs, we estimate the priors $\widehat{P}(l_n, l_m \mid \boldsymbol{\lambda}_n, \boldsymbol{\lambda}_m)$ separately for redundant pairs of matches, as we expect them to have a different distribution than non-redundant pairs. The relative frequencies of labelings for pairs of redundant correspondences in our training dataset are shown in Figure 4.26. Note that redundant pairs appear only for groups of matches referring to the same feature type, in contrast to the general prior probabilities described in Section 4.3.2. The plot shows that the relative frequency of redundant groups with labeling $(l_n = 0, l_m = 0)$ – denoting that both involved candidates are outliers – is significantly higher for blobs and affine blobs than for junction features. This is an interesting observation that might need further investigation. A possible explanation would be that the dominant orientation, which is taken from the SIFT descriptors, tends to be more stable in case of junction features compared to blobs, as junction patches typically contain more edge-like structures.

4.4 Summary

In this chapter we developed a generic approach to wide baseline stereo matching. We put a special focus on its statistical interpretation as the MAP estimate of a binary classification problem. The statistical model leads to an energy function that can be approximated very well by a linear program using the LP-S relaxation of Schlesinger (1976).

We also derived a number of reasonable energy potentials from training data, which we will use for our particular implementation of the framework. They exploit three types of pairwise spatial relationships: The sidedness of one feature w.r.t. another feature, the angle spanned by two features, and the spatial distance between them. These potentials are directly derived from the likelihood functions of the observed entities, given the labels of the involved matches.

Chapter 5

Automatic Annotation of Feature Correspondences

The proposed framework for wide baseline stereo matching requires estimates of the likelihood distributions for all kinds of observations, given the labeling of involved candidate correspondences. Estimating these distributions requires labelled training data over a larger number of image datasets, where each image pair produces hundreds of putative feature matches. In this chapter, we will therefore present a novel setup that can do the annotation automatically, instead of labeling all matches manually. We will also use this annotation setup to support our final experimental evaluations.

Our evaluation scheme is based on the following simple idea: If the projection matrices of the cameras are known, we can compute a forward intersection for each putative feature match, yielding a point or line in the 3D space of the scene. Assuming that the 3D structure of the scene is known, we measure whether this 3D point or line sits on the surface of the scene in order to decide wether the match is correct or not. This requires us to gather a reference surface model of the scene, which we will obtain by laser scanner measurements for most datasets. The approach will apply uncertain projective geometry wherever possible to obtain statistically justified annotations.

Before we describe the approach in detail, we summarize some important evaluation schemes proposed by other authors, and refine the notion of "inliers" and "outliers" in our setup.

5.1 Related Work

The most frequently applied evaluation scheme for feature detection and matching is that of Mikolajczyk et al. (2005). It is designed for the class of affine invariant features and assumes that the local patches can be represented by an ellipse, which holds for the point features that we use in this work. The basic idea is to use image datasets where the point transfer between two images can be represented by a 2D homography (Hartley and Zisserman, 2004, Ch. 13). This is mainly the case if the scene consists of one single planar surface, if the baseline of the two views is zero, or if the scene objects are infinitely far away. The authors provide a number of datasets together with carefully estimated homographies, which exhibit affine distortions, rotations, scale differences, and blur, amongst others. However, despite the fact that no datasets with sparse texture are provided, this method does not cover scenes with multi-planar or complex 3D structure. Furthermore, the homographies do not provide a transfer for the line segments, which makes the approach insufficient for our purpose.[1] Nevertheless we will present experimental results for some of these datasets (GRAFFITI and BOAT, cf. Section 6.2), using point features only.

A more advanced evaluation method has been proposed by Moreels and Perona (2006), who rely on the geometry of three views, leading to *trifocal tensors* for the feature transfer (Hartley and Zisserman, 2004, Ch. 15). This establishes a transfer for 2D lines as well: While the 3D planes through corresponding lines in two views do *always* intersect in a 3D line, the planes from three corresponding views do only intersect in a 3D line if the geometry is consistent. Furthermore, the approach of Moreels and Perona (2006) does not restrict the structure of the scene, so it would in principle be suitable for our approach. However, a match can only be evaluated if the correct observation of the corresponding feature in a third view exists and is known. As we are interested in difficult datasets with very low texture and possibly low overlap, this would impose a strong restriction on our setup.

More recently, Strecha et al. (2008) have proposed a setup that uses ground truth data from LIDAR measurements for evaluating automatic image orientation and image-based surface reconstruction methods. A dense ground-truth sampling of the scene surfaces as well as ground-truth projection matrices of the cameras are obtained from the LIDAR measurements, and estimates of projection matrices and surfaces are benchmarked against this reference.

In the spirit of Strecha et al. (2008), and motivated by the shortcomings of the evaluation methods for feature correspondences discussed above, we will develop a new automatic evaluation approach for point and line feature correspondences, which uses reference measurements of the scene surfaces and is particularly well suited for our problem.

5.2 Definition of an Outlier

Several definitions of a bad correspondence, or *outlier*, are common referring to image feature correspondences. The two most typical ones are the following.

1. In the context of image orientation, an outlier usually refers to a correspondence which is not in agreement with the image geometry, up to an expected accuracy. This doesn't necessarily imply that both features in the image show the same object area: So-called "virtual correspondences" may have a valid physical geometry, but point to a non-existing or occluded object in the scene.[2]

2. In the context of object recognition, one usually requires corresponding features to represent the same visual property of an object. The accuracy or geometric consistency is not a primary concern. For example, for identifying a person's face in two different images, it is important that the correspondence reflects the same face part in both images, e.g. the left eye. It is neither required that the exact location of the feature is highly accurate, nor that the correspondence satisfies the geometry of the camera pair.

Similar to 1., we define outliers as feature correspondences that are not in agreement with the geometry of the image pair, or do not represent an element of a "real" surface in the scene. This enables us to use reference measurements of the real scene surfaces as a basis for testing. At the same time we must accept that possibly correct correspondences referring to

[1] Note that the start- and endpoints of line segments can not be used for the evaluation, as they are not stable across images.

[2] For example, the intersection point of two line segments referring to the same plane in 3D is a valid correspondence in terms of camera geometry, but may point to different scene content in the image plane. This happens especially if the intersection point sits outside of the physical 3D plane.

virtual points are classified as outliers. Such a situation may occur on specular surfaces, for example, as depicted in Figure 5.1.

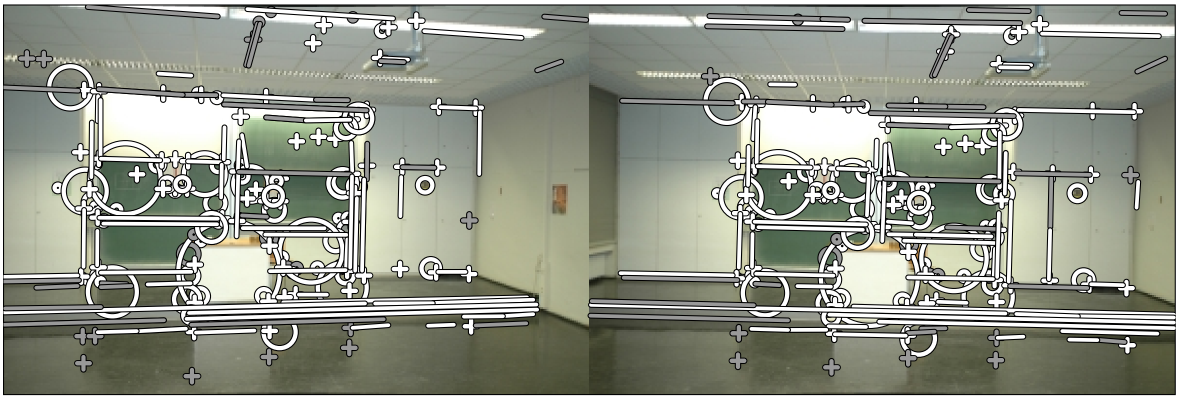


FIGURE 5.1: Image pair overlaid with matched image features, where crosses denote junction features, circles denote blobs, and lines denote straight line segments. The color of the features encodes the result of the proposed automatic annotation procedure: White features belong to correspondences classified as correct, and grey features to correspondences classified as outliers. Observe the junction feature correspondences produced by optical reflections on the floor, which are possibly consistent with the two-view geometry, but do not correspond to the physical surface. Such "virtual correspondences" are classified as outliers by our evaluation scheme.

5.3 Evaluation Scheme

The idea is to use dense 3D point measurements as a model of the true surfaces with sufficiently superior precision, similar as in Strecha et al. (2008). In particular, we will use a terrestrial laser scanner to obtain measurements for indoor scenes, and some artificial 3D models with images rendered using raytracing techniques. Furthermore, we employ some of the datasets of Strecha et al. (2008), where surface measurements taken by LIDAR devices are provided.

We make the following assumptions for a feasible dataset:

1. A dense, accurate and mostly outlier-free 3D point cloud is available, which is not necessarily textured, but represents the physical surface with a precision superior to that of a typical image-based reconstruction algorithm.

2. A set of overlapping images is available, which depicts exactly the same scene surfaces as modelled in the point cloud. This implies that image points showing an object that is not contained in the reference surface dataset will be classified as outliers.

3. A smaller number of control points is available, i.e. some of the 3D points in the surface measurements have known observations in several overlapping images. These control points are needed as a starting point for registering the different coordinate systems.

4. A good photogrammetric model is available, i.e. an estimate of the relative orientation of the images together with a larger number of several hundreds of corresponding 3D points. We will use these points to refine the registration of the coordinate systems.

For all of the elements above, we assume to have a reasonable estimate of their accuracy, provided by a covariance matrix. Example results of the annotation scheme are shown in

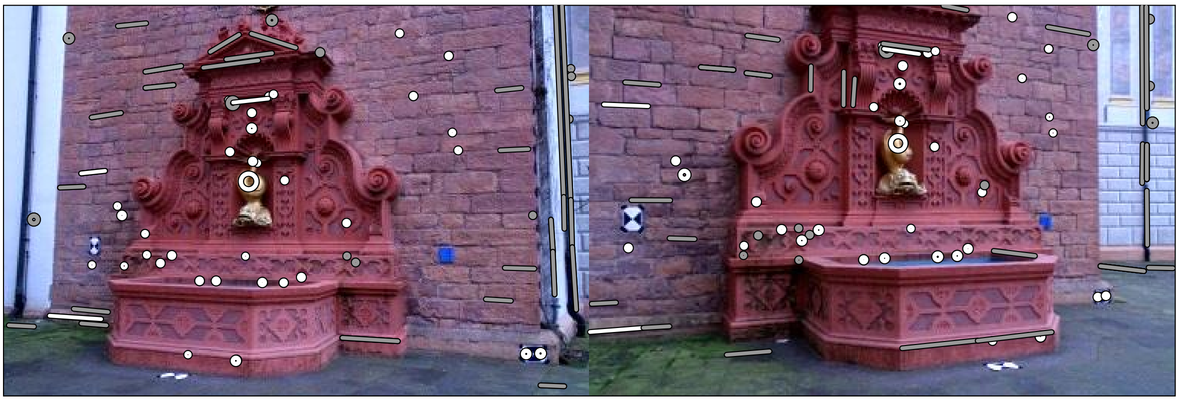


FIGURE 5.2: Example image pair from the FOUNTAIN-P11 dataset, showing a set of feature correspondences that have been automatically annotated using the procedure described in Section 5.3. As in Figure 5.1, white features are classified as correct, and grey features are classified as outliers.

Figure 5.1 using a terrestrial laser scanner in an indoor scene, and in Figure 5.2 using LIDAR-measurements in an outdoor scene.

5.3.1 Semi-Automatic Registration of Projection Matrices

We first transform the projection matrices of the cameras into the coordinate system of the surface measurements, based on the control points. The procedure works in two steps:

1. An approximate solution for the similarity transformation of the cameras into the new system is computed, using the control points, and yielding approximate values for rotation $\widehat{R}$, translation $\widehat{\boldsymbol{t}}$ and scaling $\widehat{s}$.

2. The estimated parameters are refined based on the full 3D point cloud and the forward-intersected 3D points of the photogrammetric model. We choose to implement the refinement by an Iterative Closest Point (ICP) algorithm (Zhang, 1994), which has given sufficiently accurate results in our experiments.

The final parameters are used to transform the projection matrices of the images into the coordinate system of the reference surface measurements. Additionally, the results are visually inspected for each dataset.

We would possibly obtain more accurate estimates when computing a spatial resection for each camera, based on control points and using full covariance information. However, this would require a sufficient set of visible control points to be measured for each image. Considering that a point is visible in three images on average, the number of required control points would increase strongly with the number of images of a dataset. Our procedure instead works well with about ten manually measured control points per dataset, each of which is observed in three or four images, mostly independent of the number of images in the dataset.

5.3.2 Annotation of point feature correspondences

We start by storing the reference measurements of the surface into an efficient K-D tree structure for fast nearest neighbour queries.

Given a point feature correspondence, which provides two corresponding points in an image pair, we obtain the estimated 3D point $\widehat{\boldsymbol{X}}$ by forward intersection based on the transformed projection matrices. The forward intersection is computed as a least squares estimate, using uncertain projective geometry and applying the full covariance information of the projection matrices.[3] Besides the 3D coordinates, this provides us with an estimate $\underline{\Sigma}_{XX}$ of each point's covariance matrix.

We then search for samples of the reference surface which are statistically incident with $\widehat{\boldsymbol{X}}$. To realize this, we determine the maximum eigenvalue $\lambda_{\max}$ of the covariance matrix $\underline{\Sigma}_{XX}$ as an estimate of the standard deviation in direction of the largest error in $\widehat{\boldsymbol{X}}$. We then determine the subset

$$\mathcal{X} = \{\boldsymbol{X} \mid \; |(\widehat{\boldsymbol{X}} - \boldsymbol{X})|^2 < \lambda_{\max}\}$$

of reference surface measurements, and test whether at least one element in $\mathcal{X}$ is statistically incident with $\widehat{\boldsymbol{X}}$. The incidence test is again performed using uncertain statistical reasoning, with the original covariance matrix $\underline{\Sigma}_{XX}$. The accuracy of the surface measurements, if not otherwise available, is estimated based on the average distances of measurements in a local neighborhood.

It may seem confusing that we use the set $\mathcal{X}$ instead of simply taking the nearest neighbor from the K-D tree for the incidence test. To understand this course of action, observe the situation depicted in Figure 5.3. A typical error ellipse of the forward intersected point has a lengthy shape. Just as in the illustration, the nearest point on the surface is often not statistically incident with the forward intersection, but other points on the surface are.

In the special case where the intersection angle for the forward intersection is very small (below 0.5°), the test is skipped, and the correspondence is interpreted as an outlier.

5.3.3 Annotation of line segment correspondences

For a line segment correspondence, we perform four tests. For each start- or endpoint, we forward intersect the corresponding 3D line with the 3D plane corresponding to the line segment in the other view. Again we perform the construction using uncertain projective geometry, with the full covariance matrices provided by the EDGE line segment detector. We thereby obtain an estimate of the 3D point $\widehat{\boldsymbol{X}}$ corresponding to the respective start- or endpoint. Other than for the point features however, this forward intersection is always valid.

With each of the four 3D points that we obtain from these forward intersections, we perform a statistical incidence test with the surface measurements, as described for point feature correspondences above. If any of these tests fails, the line segment match is classified as an outlier.

[3] The estimates and statistical tests are computed using the SUGR library for statistically uncertain geometric reasoning (Heuel, 2004), available at `http://www.ipb.uni-bonn.de/sugr/`.

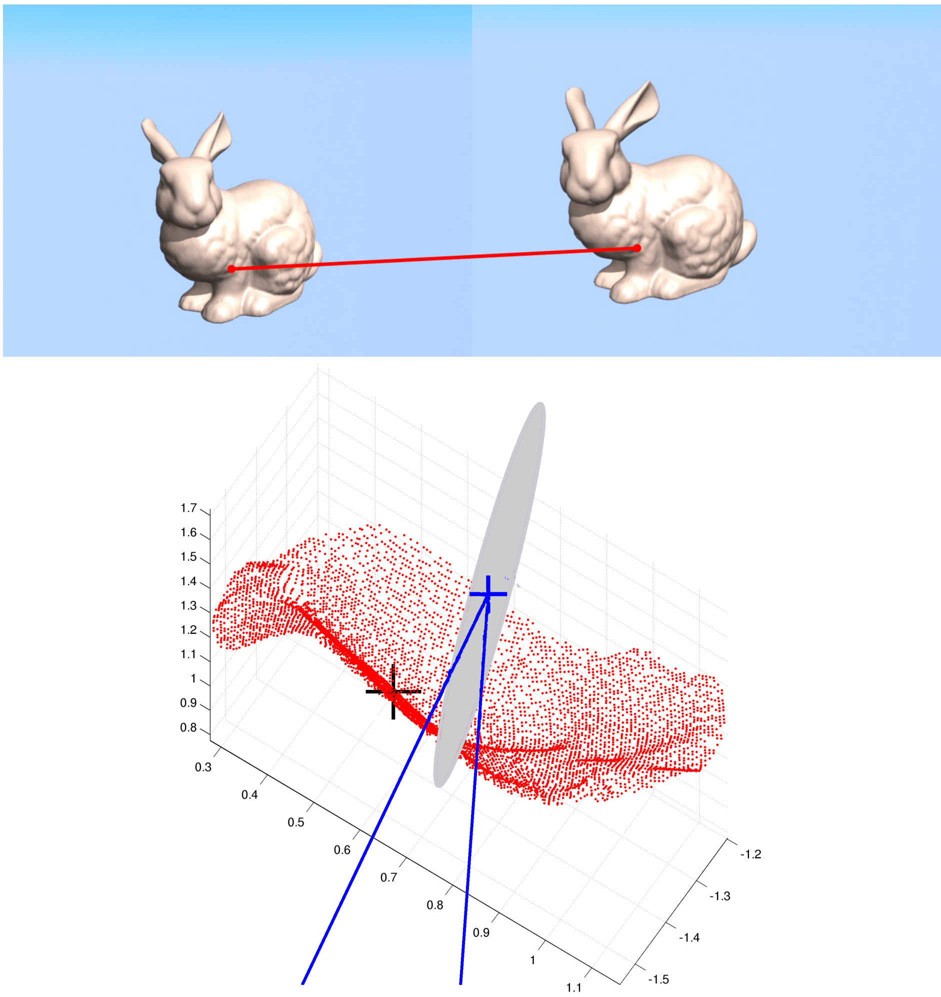


FIGURE 5.3: Top: A point feature correspondence related to an artificial image pair, showing the "Stanford Bunny". Bottom: Reference surface measurements (red dots) in the local surface area of the correspondence, estimated 3D point $\widehat{\boldsymbol{X}}$ (blue cross where the lines intersect), and the nearest point on the reference surface (black cross). The error ellipse of $\widehat{\boldsymbol{X}}$, here illustrated in grey, is typically lengthy, with a larger error in the direction of the intersecting lines. Therefore, the nearest point on the surface is often not statistically incident with $\widehat{\boldsymbol{X}}$, but another point within a radius corresponding to the maximum error according to the ellipse. The 3D model of the bunny is taken from the *Stanford 3D scanning repository* at `http://graphics.stanford.edu/data/3Dscanrep/`.

Chapter 6

Experimental Results

In this chapter, we will show that the framework for wide baseline stereo matching developed in Chapter 4 (MapMatch) allows for significantly better matching results on sparsely textured scenes than the standard best-matching approach (BestMatch-2), which only takes descriptor dissimilarities into account. We also want to make sure that our results are at least comparable to the results obtained with the method of Bay et al. (2005), which is specifically designed for sparsely textured scenes. Moreover, we investigate the performance of the algorithm on regular datasets to show that the training of energy potentials for rather specific image datasets does not lead to poor results on standard data.

We will consider a matching result better than another one if it contains more correct feature correspondences at an acceptable outlier rate. As our focus is on image orientation problems, we consider outlier rates as acceptable if they are significantly below 50%, relying on our experience that robust estimators in image orientation systems can deal well with such data. For example, if one matching algorithm returns 20 correct correspondences with no outliers, and another one returns 50 correct correspondences with 15 outliers, we consider the latter one to be better.

Besides the pure matching results, we want to investigate the effects when using different wide baseline methods as a module for a particular image orientation system. We expect our method to allow for a higher number of successfully oriented cameras than standard descriptor-based matching when working with very sparsely textured scenes. For standard scenes, we expect at least comparable results when using our method.

After describing the experimental setup, we start by showing some illustrative examples to demonstrate the behavior of the proposed wide baseline stereo matching algorithm.

6.1 Experimental Setup

6.1.1 Detectors and Descriptors

The selection of detectors and descriptors for our experiments is based on three criteria:

1. We want to use standard algorithms that have been used for wide baseline stereo matching and automatic image orientation before by other authors.

2. The detectors should have high complementarity, referring to the investigations in Dickscheid et al. (2010).

3. There should be some variability concerning the strengths of the detectors and descriptors, in the sense that the setup contains descriptors with high and low distinctiveness,

and that the robustness of the features w.r.t. variations in scale, rotation, and perspective is different.

Therefore we choose the following feature detectors:

1. The LOWE detector (Lowe, 2004) stands for the class of classical blob detectors, based on the Laplacian. It is known to have very good scale and rotation invariance. We use the original implementation kindly provided by the author, however using the original image resolution instead of the double image resolution for building the pyramid. We use SIFT descriptors for the LOWE features, also computed using the original software provided by D. Lowe. The orientation of the LOWE features is taken from the dominant gradient orientation that is assigned to the descriptor.

2. The FOP0 detector extracts interest points based on the structure tensor from the framework of Förstner (1994), and chooses the subset of junction points. These features are not scale invariant, and therefore more sensitive to affine distortions. We use the original implementation of the author, with a manually determined but fixed estimate of 0.015% for the standard deviation of the image noise. The FOP0 points are also matched using SIFT descriptors, computed on a fixed scale of $s = 4$, which corresponds to an effective window size of $3s = 12$[pel].[1] Again, the descriptor provides us an orientation for the features.

3. The MSER detector of Matas et al. (2004) stands for the class of affine invariant regions. We use the widely used implementation provided by Mikolajczyk et al. (2005). For assigning SIFT descriptors to the MSER features, we use a circular region that covers the same area as the elliptical representation of the affine invariant feature, placed at the same image location. Therefore we cannot exploit the full expressive power of the MSER features, and the results must not be understood as a representative evaluation of the MSER algorithm. The orientation for MSER features comes is also taken from the SIFT descriptors, as the ellipse orientations are only defined up to a 180 degree ambiguity.

4. The EDGE detector from the framework of Förstner (1994) provides a typical straight line segment detector. We use color-histogram based descriptors as proposed by Bay et al. (2005) for the segments (Section 2.3), which are significantly less distinctive than the SIFT descriptors for the other detectors. We use our own implementation for the descriptors, which has been carefully compared to the implementation of the authors and leads to very similar results. The EDGE features are only rotation invariant, so they will suffer from strong scale and affine distortions. The orientation of the line segments follows from the line direction. To overcome the 180 degree ambiguity inherent to the direction, we analyze the image intensities of the neighboring pixels on both sides of each segment, and define the side with the brighter pixels to be the left side w.r.t. to the segment. This method is also used by Bay et al. (2005).

We want to emphasize again that our experiments must not be understood as a comparison of detectors, but as a comparison of wide baseline matching methods. By keeping the set of detectors and descriptors together with their parameter settings fixed, all methods shown here have to cope with the same strengths and shortcomings of the features.

[1] Note that the scale parameter s refers to the value σ as used in Lowe (2004).

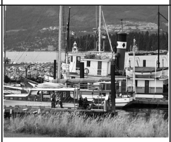
Dataset	Class	Boat	Graffiti	Blank-12	Blank-22	Dragon
Annotation	manual	homography	homography	manual	manual	surface-based
Texture	sparse	strong	strong	very sparse	very sparse	sparse
3D structure	multi-planar	quasi-planar	planar	multi-planar	multi-	complex
Distortion	affine	rotation and scale	strong affine	affine	affine	affine
Overlap	$\sim 60\%$	$\sim 100\%$	$\sim 100\%$	$\sim 90\%$	$\sim 90\%$	$\sim 100\%$
# Images	8	6	6	12	22	6
Resolution	752×500	213×170	213×170	1203×800	752×500	800×600
Example						
See page	92	91	91	89	90	93

Table 6.1: Properties of the datasets used for our experiments.

6.1.2 Matching Algorithms and Training Data

We show results for three different wide baseline stereo matching algorithms. The simplest and most common one is a classical descriptor-based best matching approach (BestMatch-2) with a 70% threshold, as described in Section 2.4.

Furthermore, we use a reimplementation of the method proposed by Bay et al. (2005), which will be denoted as TopoMatch in the following. It includes both the three-point- and the point-line topological filtering stages described in Section 3.2.2, and the boosting step. Although we reimplemented the method carefully, we cannot claim that the results apply directly to the original implementation of the authors.

Our own method developed in Chapter 4 is denoted as MapMatch in the following. *The parameters for the potential functions have been trained on the set of image pairs shown on page 88, and remain constant over all experiments.* The training images are not part of any dataset used for the experiments, except for the Class dataset.

6.1.3 Image Datasets

We show results based on five different datasets. The properties of the datasets are summarized in Table 6.1. Note that some of the images in Class are part of the training dataset (page 88), while the other datasets are not related to the training data. The Class, Blank-12 and Blank-22 datasets used a fisheye lens, and have been corrected for radial distortion.

The Boat and Graffiti datasets are taken from Mikolajczyk et al. (2005), but have been reduced to a significantly lower resolution to decrease the amount of features. This has been necessary because the complexity of the TopoMatch and MapMatch methods is too high for processing high resolution images with strong texture. As we did not want to put a restriction on the number of features into the algorithms, we decided that downsampling the images was the easiest and most natural way of reducing the amount of features.

6.2 Results for Pairwise Feature Matching

For investigating the success of a method referring directly to the extracted feature correspondences, we report the number of good correspondences (inliers) and the percentage of outliers for each matched image pair. As stated before, we consider better algorithms to have higher numbers of inliers at an outlier rate that does not exceed 40%. Although we report the statistics separately for each feature types, the matching has been performed on all feature types simultaneously.

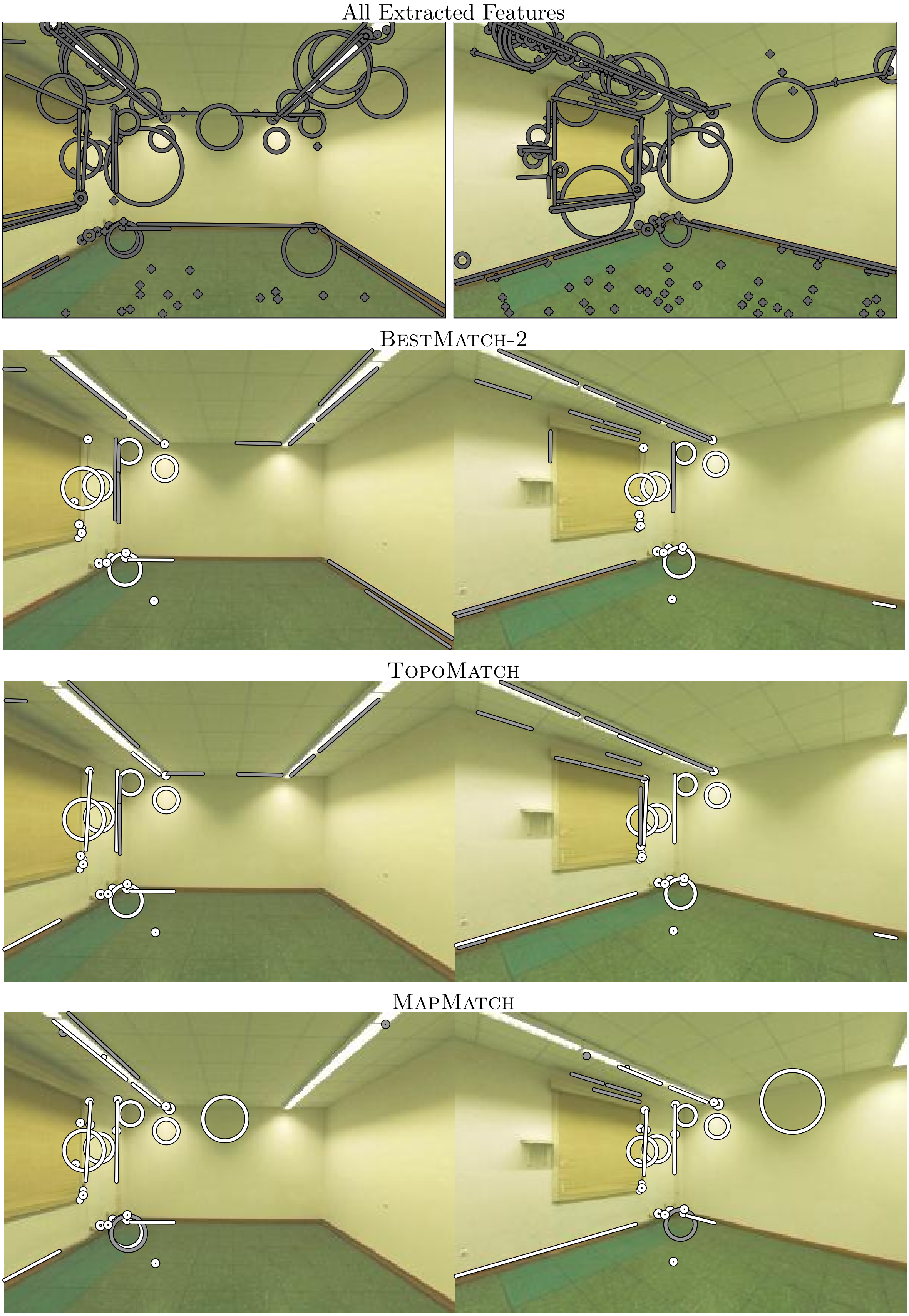
6.2.1 Sparsely textured datasets

Referring to the datasets with sparse texture, our approach MapMatch shows mostly superior matching results. First of all, consider the image pair of the Class dataset depicted in Figure 6.1. It provides a visual impression of the matching results on such scenes for the different methods. We see that the BestMatch-2 approach, relying only on descriptors, cannot compensate the weakness of the line segment descriptors, which results in many outliers among the line segment correspondences. Using the topological filter in the TopoMatch method removes many outliers, but does not lead to a higher number of point feature correspondences. The MapMatch approach (bottom) achieves both effects quite well. Figure 6.2 shows detailed results for more image pairs of the Class dataset. We see that our approach yields a constantly higher number of inliers. In case of the straight line segments, the outlier rates are also smallest for our approach. For other feature types however, it tends to have higher outlier rates than the other methods. The subset of mser feature correspondences has outlier rates exceeding the 50% border for MapMatch.

For the Blank-12 dataset (Figure 6.3), one obtains similar observations. The number of inliers is significantly higher for MapMatch over all considered image pairs and feature types, while the outlier rates are acceptable, sometimes even better than for the other two methods. In particular, MapMatch would allow to compute the epipolar geometry of the third pair 6/9 quite robustly, with a total of 36 correct point matches (ignoring the line segments), while TopoMatch with 6 point matches is clearly at the borderline, and BestMatch-2 with 21 point matches significantly weaker. The TopoMatch implementation does not yield significantly more inliers than BestMatch-2, but has lower outlier rates. This is intuitive, considering that it removes matches with inconsistent spatial relationships.

6.2.2 Strongly textured datasets

The results for the Boat dataset (Table 6.4) show that our approach yields comparable results to the classical BestMatch-2. Note that here the image pairs are sorted by increasing scale and rotation difference between the images. For strong distortions, MapMatch yields more inliers than the BestMatch-2 approach, at the cost of a slightly higher outlier rate. Nevertheless it has a tendency to extract too many outliers at times, as can be seen in case of the affine region features for image pairs 1/4 in Table 6.4, and in case of the blobs for pair 1/6. The TopoMatch approach yields very similar results to BestMatch-2, with a tendency to extract even less matches. Note that although the line segments were used for matching in all of our experiments, they are not listed for the Graffiti and Boat dataset, as the homography-based annotation cannot evaluate them automatically.



All Extracted Features

BestMatch-2

TopoMatch

MapMatch

Figure 6.1: Visual matching results for an image pair of the Class dataset for the three methods described in Section 6.1.2. Features depicted in white are correctly matched, features in grey are outliers. We see that the simple BestMatch-2 approach gives quite many inliers, especially among the line segments which have the weakest descriptors. Using a topological filter and boost stage (TopoMatch) removes a significant number of the outliers. The results for our approach (MapMatch) contain more inliers, and at the same time the lowest outlier rate. Detailed results more image pairs of the dataset are listed in Table 6.2.

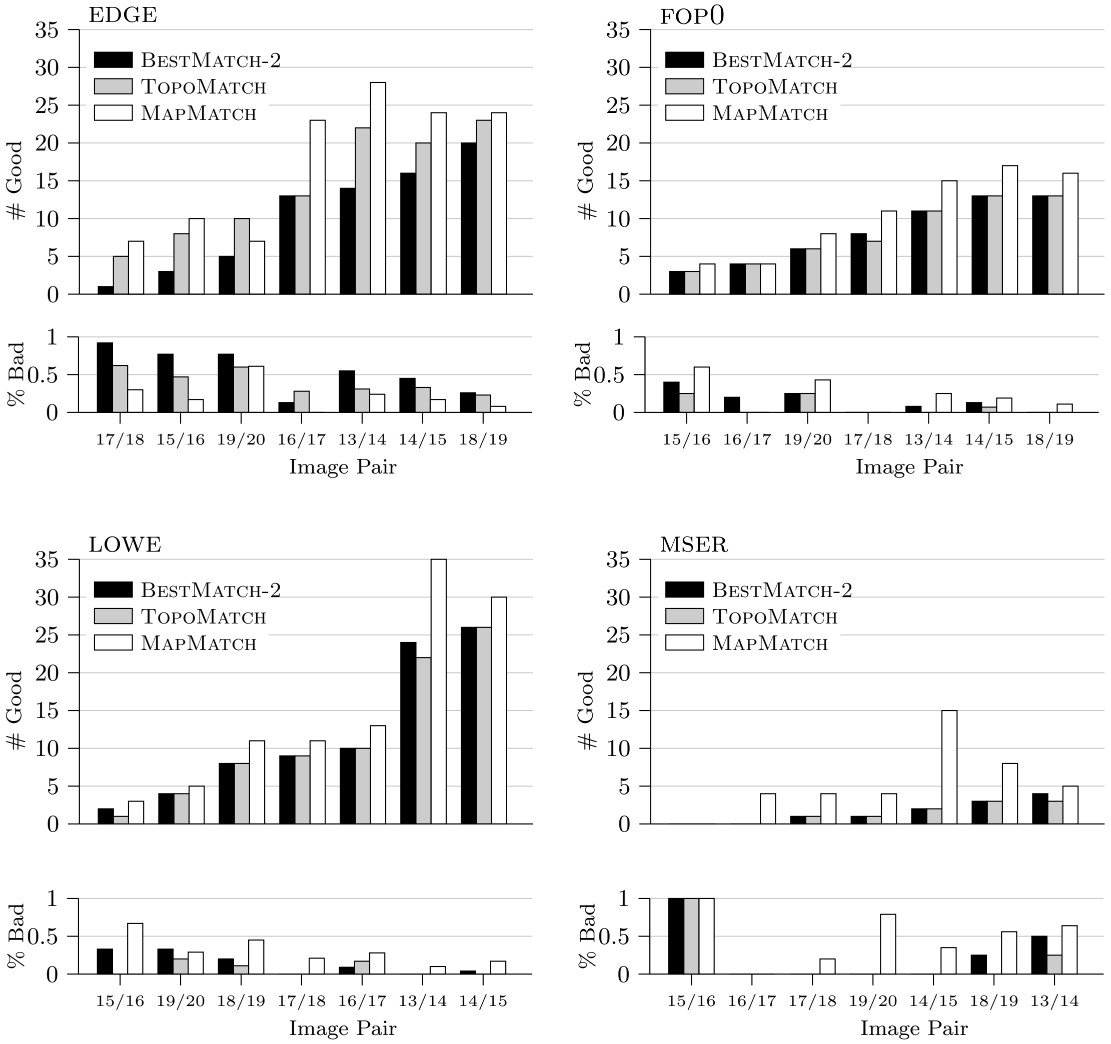


Figure 6.2: Matching results for all neighboring image pairs of the Class dataset (Section A.6 on page 92), computed with the three wide baseline stereo matching algorithms described in Section 6.1.2. Shown are the number of correct correspondences and the percentage of outliers for each feature type. The annotation has been done manually. We see that our approach (MapMatch) most often yields higher numbers of inliers then the others at slightly higher but acceptable outlier rates. For the mser features however, it tends to select too many matches here, yielding too large outlier rates.

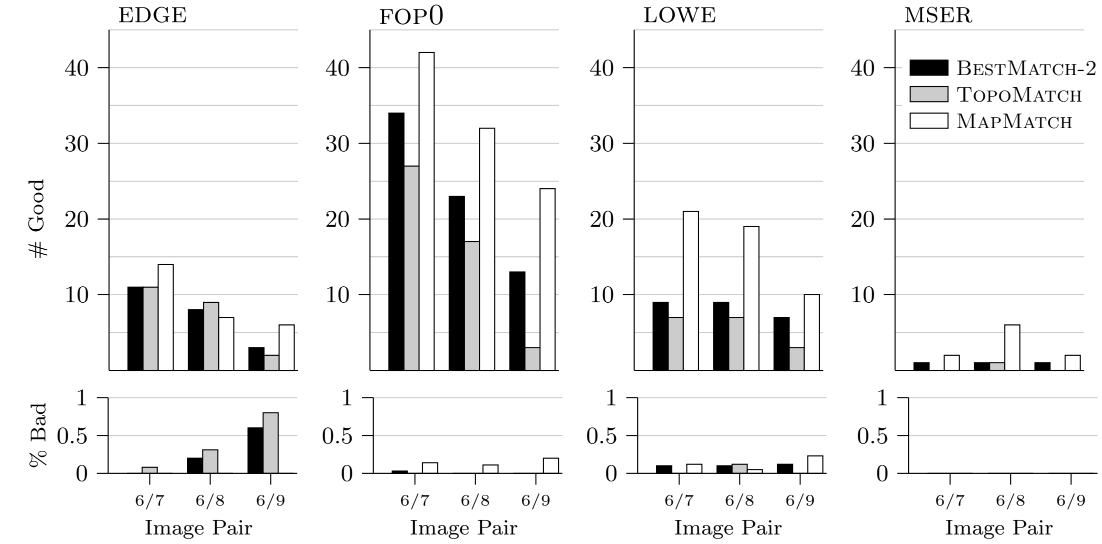


Figure 6.3: Results for three image pairs with increasing baseline taken from the Blank-12 dataset (Section A.2 on page 89). The number of inliers is significantly higher for MapMatch, while the outlier rates are still good, sometimes also better than for the other two methods. In particular, MapMatch would allow to compute the epipolar geometry of the third pair 6/9 quite robustly, with a total of 36 correct point matches (ignoring the line segments), while TopoMatch with 6 point matches is clearly at the borderline, and BestMatch-2 with 21 point matches significantly weaker.

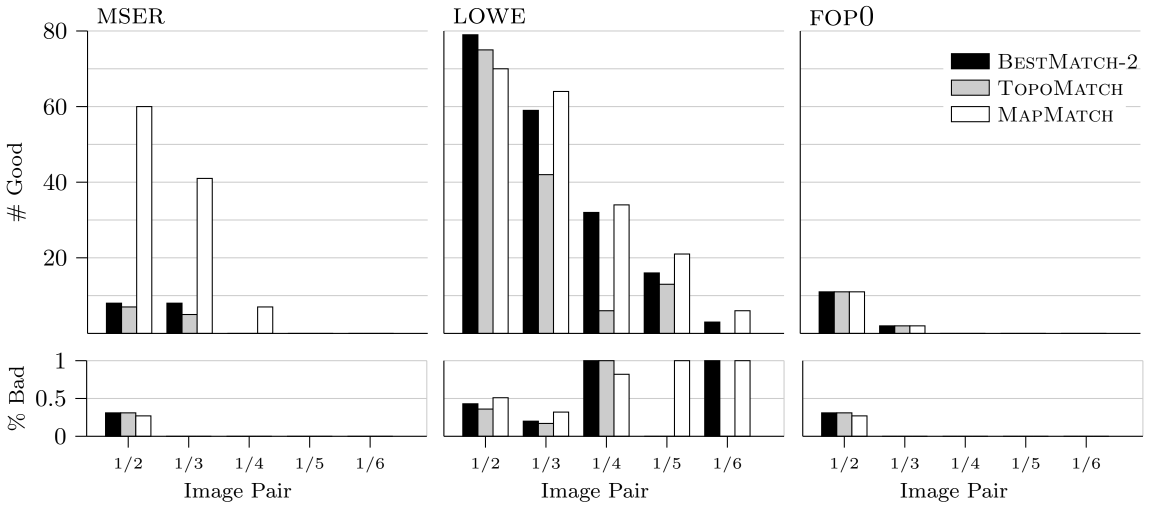


Figure 6.4: Matching results for all image pairs containing the first image of the Boat dataset (Section A.5 on page 91), computed with the three wide baseline stereo matching algorithms described in Section 6.1.2. The annotation has been done based on plane homographies, which works only for point features. The image scale and rotation difference per image pairs increases significantly from left to right.

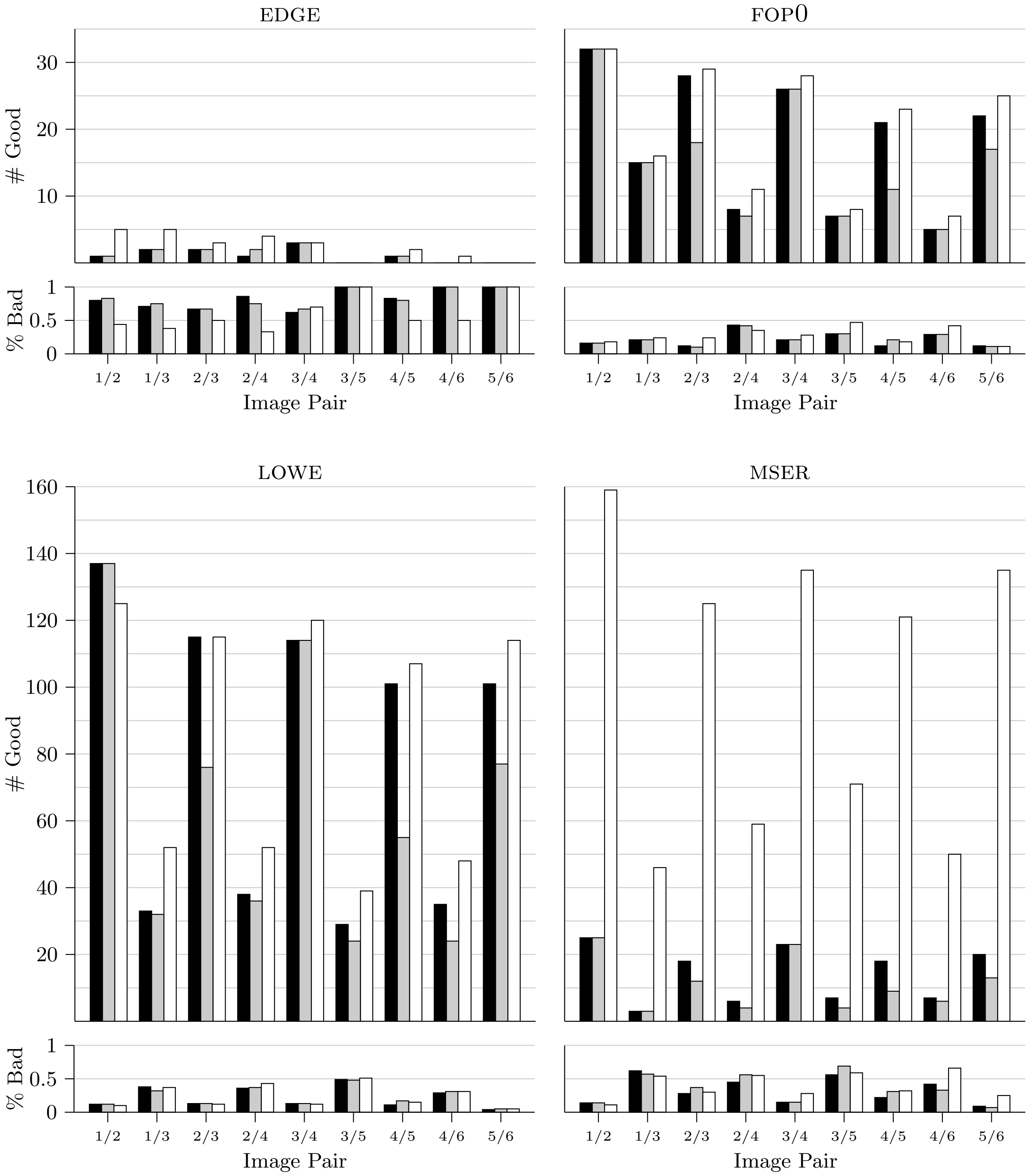


Figure 6.5: Results for overlapping image pairs for the Dragon dataset (Section A.7 on page 93). The matching of edge features seems to be particularly difficult here for all three methods. The MapMatch approach solves it significantly better, though still not satisfyingly. For the other feature types, the MapMatch approach shows consistently better results in terms of higher number of inliers at comparable and satisfying outlier rates. Observe especially the affine blobs, where MapMatch extracts between 7 and 10 times more inliers, at a only slightly higher outlier rate.

6.2.3 Results for straight line segments

The straight line features play a special role, as the matching of lines is in general more difficult due to the uncertainty of the location of the start-/endpoints, and in particular more difficult due to the weak descriptors used here. On the investigated datasets, the MapMatch approach shows better results than both other methods referring to the line segments. At the same time, the TopoMatch method shows often better results for matching lines than BestMatch-2. We can therefore conclude that the spatial relationships seem to play indeed an important role for matching features with weak descriptors.

6.3 Impact onto a System for Automatic Image Orientation

We have seen in the previous section that the proposed approach MapMatch often extracts a significantly higher number of inliers than the classical best matching approach BestMatch-2 and our implementation of the topological filter and boost approach TopoMatch. We will now investigate the effect of using different wide baseline matching methods onto a system for automatic image orientation. We base the experiment on the system aurelo for automatic relative orientation, which we will explained next.

6.3.1 The System aurelo for Automatic Image Orientation

In the following we will briefly describe the image orientation system aurelo (Läbe and Förstner, 2006) that we use for evaluating our matching algorithm. Note that besides aurelo, a variety of other automated systems for solving the relative orientation problem have been proposed (Pollefeys et al., 2000; Roth, 2004; Mayer, 2005; Vergauwen and Gool, 2006; Snavely et al., 2006; Strecha et al., 2008).

The task in automatic image orientation is to derive the relative 3D motion between cameras from a set of overlapping images (Figure 6.6).

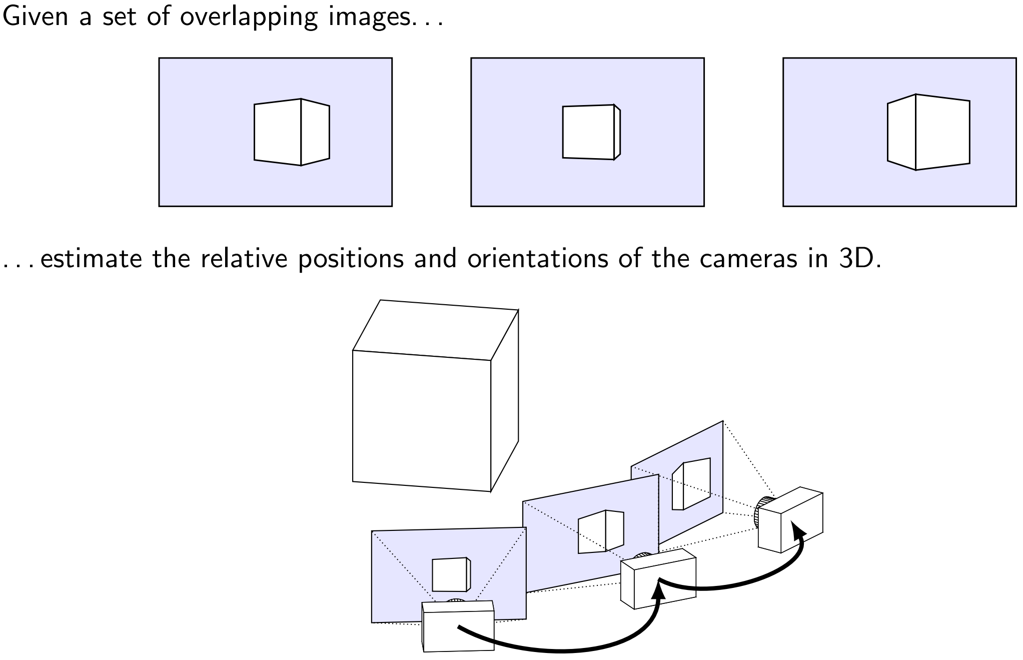

Given a set of overlapping images...
...estimate the relative positions and orientations of the cameras in 3D.

Figure 6.6: The problem of automatic image orientation, illustrated by three overlapping images of a cube.

The procedure starts by computing a set of point feature correspondences for each pair of images in the dataset. It implements the BestMatch-2 algorithm (cf. Section 2.4), applied on lowe features with SIFT descriptors (Lowe, 2004). However, like most other image orientation methods, it can use any wide baseline stereo matching algorithm that delivers point feature correspondences for pairs of images. The intrinsic camera parameters are assumed to be known aurelo. The relative orientation of each image pair is computed using the 5-point algorithm (Nister, 2004) embedded into a RANSAC scheme (Fischler and Bolles, 1981; Hartley and Zisserman, 2004). This produces robust approximate values for the pairwise epipolar geometries, and also acts as a filter on the feature correspondences, resulting in smaller sets of correspondences with usually significantly reduced outlier rates. Based on the filtered sets of pairwise feature correspondences, multiview correspondences are derived by simple propagation of the feature indices in the views.

Pairwise camera geometries are then connected in an iterative manner, prioritized by a measure of quality that is based on the number of satisfied coplanarity constraints.[2] A number of threefold correspondences, i.e. multiview feature matches spanning at least three views, are required to determine the scale between two pairwise camera geometries. aurelo will only select one set of connected camera orientations. In case that no further pairwise camera geometry estimates can be connected, the procedure stops.

Triplets of camera orientations are used for further elimination of invalid pairwise geometries. In particular, the product of the rotation matrices referring to three connected views must be approximately equal to an identity matrix, and the three involved stereo baselines have to be coplanar.

After determining 3D object points from the final multiview feature correspondences by forward intersection, the whole block undergoes a nonlinear global optimization using the sparse bundle adjustment software developed by Lourakis and Argyros (2009).

6.3.2 Evaluation Strategy using aurelo

We use the three different algorithms for wide baseline stereo matching described in Section 6.1.2 and the feature detectors listed in Section 6.1.1 for generating the input data for aurelo. Note that we use the edge detector although the actual input to aurelo consists of point feature correspondences only. This is because the line segments have an influence on the matching results when using spatial relationships, which also affects the final point feature correspondences.

For each dataset, we compute feature correspondences for pairs of images, and provide them as an input to aurelo. We reduced some of the default thresholds in aurelo to compensate for the small expected amount of correspondences due to the sparse texture. In particular, we reduced the minimum number of point feature correspondences required for estimating a pairwise camera geometry from 100 to 30, and the minimum amount of three-fold point observations for connecting two pairwise geometries to three. As aurelo contains a random component, namely a RANSAC scheme for computing robust estimates of the pairwise epipolar geometries, we repeat each experiment 20 times.

We report the following indicator values:

1. The average percentage $\overline{P}_o$ of images that have been successfully included in the final estimate.

[2] The coplanarity constraint basically states that the stereo baseline and the two rays going through corresponding image locations in the left and right view sit on the same plane in 3D space.

Method	$\overline{P}_o$	$\widehat{\sigma}_{x'}$	$\overline{N}_I$	$\overline{\sigma}_\phi$	$\overline{N}_o$
Detectors	FOP0,LOWE,MSER,EDGE				
BestMatch-2	11.0	0.72	139	0.55	3.12
TopoMatch	11.0	0.69	152	0.43	3.08
MapMatch	12.9	0.78	158	0.60	3.15
Detectors	SFOP0,LOWE,MSER,EDGE				
BestMatch-2	22.0	0.60	412	0.31	3.46
TopoMatch	22.0	0.63	443	0.29	3.47
MapMatch	22.0	0.64	421	0.31	3.56
Detectors	SFOP0,EDGE				
BestMatch-2	12.0	0.65	151	0.23	3.56
TopoMatch	12.0	0.64	152	0.22	3.53
MapMatch	18.0	0.68	221	0.25	3.85

Table 6.2: Indicator values for repeated AURELO estimates of the image orientation for the Blank-22 dataset (Page 90), using varying sets of detectors.

2. The average standard deviation of observations $\widehat{\sigma}_{x'}$ as estimated by the bundle adjustment, reflecting the accuracy of observations.

3. The average number $\overline{N}_I$ of 3D object points observed in an image, indicating the stability of the estimated orientation for each particular image.

4. The average number $\overline{N}_o$ of independent observations of the 3D object points in overlapping images, indicating stability of the estimated camera poses.

5. The average standard deviation $\overline{\sigma}_\phi$ of the camera orientation in degrees, referring to the bundle adjustment. This is an approximate value, directly computed from the variances $\sigma^2_{\mathbf{q}}$ of the rotation quaternions $\mathbf{q} = [q_0, q_1, q_2, q_3]^\mathsf{T}$ estimated by the bundle adjustment, using
$$\sigma_\phi = \frac{180}{\pi}\sqrt{\sigma^2_{q_0} + \sigma^2_{q_1} + \sigma^2_{q_2} + \sigma^2_{q_3}}\ \text{[degree]} \tag{6.1}$$
This value indicates the accuracy of the camera orientations.

6.3.3 Results

Blank-22 dataset. The Blank-22 dataset is particularly difficult due to the very sparse surface texture and strong affine distortions caused by the high viewing angle of the lens. If we run AURELO using the four feature detectors described in Section 6.1.1, none of the wide baseline methods allows a successful orientation of all cameras. The results are shown in the upper third of Table 6.2. The MapMatch approach is most successful, as it allowed for estimating 12 or 13 of the images, while the results for both BestMatch-2 and TopoMatch do not exceed 11 images.

The particularly bad results show the difficulty in processing such datasets. As mentioned in the introduction, many factors are important in such a case, above all good complementary combinations of robust detectors, and a good matching algorithm. While the applied detectors are highly complementary, the robustness is not perfect, due to FOP0 not being scale invariant, and MSER having restricted performance here (cf. Section 6.1.1). Therefore we tried these datasets by replacing the FOP0 features with scale-invariant junction features from the recently proposed SFOP detector (Förstner et al., 2009), denoted as SFOP0. The

Method	$\overline{P}_o$	$\widehat{\sigma}_{x'}$	$\overline{N}_I$	$\overline{\sigma}_\phi$	$\overline{N}_o$
Resolution	1203 × 800				
BestMatch-2	12.0	0.68	202	0.20	3.57
TopoMatch	12.0	0.63	270	0.30	3.05
MapMatch	12.0	0.66	229	0.19	3.31
Resolution	752 × 500				
BestMatch-2	9.0	0.52	109	0.29	3.41
TopoMatch	4.5	0.55	68	0.52	3.28
MapMatch	12.0	0.67	122	0.59	3.28

Table 6.3: Indicator values for repeated aurelo estimates of the image orientation for the Blank-12 dataset (Page 89), using two different image resolutions.

results are shown in the center part of Table 6.2. We see that the effect is enormous: All three methods yield a full orientation of the dataset now.

For further investigating the behavior, we computed twenty estimates with sets of sfop0 and the edge features only. The results are shown in the bottom part of 6.2. Again, the estimated camera orientations were incomplete for all methods, but MapMatch allowed the estimation of 18 cameras on average, while the two other methods yielded 12 cameras only. Here, MapMatch shows also the highest number $\overline{N}_I$ of object points and the highest average number $\overline{N}_o$. At the same time, the average standard deviation of observations $\widehat{\sigma}_{x'}$ is slightly highest for MapMatch, which might be an indicator that the additionally reconstructed cameras were supported by weaker observations.

Blank-12 dataset. Compared to the Blank-22 dataset, the Blank-12 dataset contains even less textured surfaces, as the window of the indoor scene is not shown. The only distinguished objects are a little magazine on the floor and a door. Besides this, some tiny structures, as for example a power jack, are visible, however at very fine scales due to the wide aperture of the camera. Using the default set, which includes the fop0 instead of sfop0 detector, none of the methods lead to a successful orientation. Therefore we used the sfop0, lowe, mser and edge detector for this experiment to get a more robust set of features. On an image resolution of 1203 × 800, all three methods were able to estimate a complete image orientation, as shown in the upper half of Table 6.3. The number of object points $\overline{N}_I$ is here significantly highest for the TopoMatch method, while the average estimated accuracy of camera rotation $\overline{\sigma}_\phi$ is best for MapMatch, and almost equally good for BestMatch-2. The average number of independent observations $\overline{N}_o$ is highest for BestMatch-2.

Reducing the resolution by a factor of almost two, the situation becomes more difficult, as shown in the lower half of Table 6.3. Here, the MapMatch approach seems to be most promising, as it allowed for the estimation of all twelve cameras over all repeated estimates, while both other methods yielded less complete estimates. However, the accuracy of the complete estimates achieved by MapMatch is worse than for the partial reconstructions of the other two approaches, as the average estimated standard deviation of the observations and camera rotations indicate.

Dragon dataset. The Dragon dataset (page 93) contains images rendered using raytracing software from a 3D model of a real object. The dataset has extremely sparse texture, but significant 3D structure resulting in rich object shadings. The results for repeated image orientation estimates, using different sets of detectors, are shown in Table 6.4. This dataset is obviously much easier to process than the other ones, as all three wide baseline stereo

Method	$\overline{P}_o$	$\widehat{\sigma}_{x'}$	$\overline{N}_I$	$\overline{\sigma}_\phi$	$\overline{N}_o$
Detectors	LOWE, SFOP0, MSER, EDGE				
BESTMATCH-2	7.0	0.71	430	0.14	2.96
TOPOMATCH	7.0	0.66	447	0.16	2.79
MAPMATCH	7.0	0.73	423	0.14	3.01
Detectors	LOWE, FOP0, MSER, EDGE				
BESTMATCH-2	7.0	0.74	523	0.14	2.94
TOPOMATCH	7.0	0.69	535	0.21	2.72
MAPMATCH	7.0	0.78	510	0.27	2.94
Detectors	LOWE, EDGE				
BESTMATCH-2	6.3	0.64	314	0.24	3.10
TOPOMATCH	7.0	0.69	375	0.17	2.88
MAPMATCH	7.0	0.77	378	0.14	3.16

TABLE 6.4: Indicator values for repeated AURELO estimates of the image orientation for the DRAGON dataset (Page 93), using different detector combinations.

matching algorithms yield successful estimates for all seven images in most cases. Only when reducing the set of detectors to no more than the LOWE and EDGE detectors, the descriptor-based approach BESTMATCH-2 looses stability and occasionally gives only six successfully oriented cameras. Both other methods, which use spatial relationships for the matching process, can benefit from the line segments and remain stable.

6.4 Summary

By visual inspection of matching results for a challenging image pair (Figure 6.1), we have demonstrated that our new method is able to extract more inliers at lower outlier rates compared to two other established methods. When considering a larger number of matching experiments, the method still produces more inliers on average at mostly acceptable outlier rates, which holds especially for sparsely textured scenes.

We have also investigated the impact of these matching results onto the problem of automatic image orientation. Under difficult conditions, namely very sparse texture, weak detectors, or small image overlap, our method often showed favorable results over several indicator values, especially concerning the number of successfully oriented cameras. We demonstrated again that using very strong detectors is also effective; in such cases standard matching algorithms often succeed just as well. Then, the proposed approach usually produced comparable results to established methods.

With the current parameters for the potential functions, our method tends to produce rather high outlier rates among the weaker feature types when using combinations of different detectors, as in case of the MSER features on the CLASS dataset. This behavior is probably related to the preselection of putative matches, which in our implementation shifts from the single best to the two or three best candidates in case of weak feature types (cf. Section 2.4). An investigation into more sophisticated preselection criteria might possibly overcome this problem.

Chapter 7

Conclusion and Outlook

We have proposed a statistically motivated, generic framework for wide baseline stereo matching. Given an initial set of putative feature correspondences, we perform a binary classification into good and bad correspondences. In a Bayesian treatment, the classification takes the statistics of descriptor similarities and geometric consistency of pairs of putative correspondences into account, which we infer from annotated datasets. The framework can handle different types of features, descriptors, and dissimilarity measures, and model arbitrary binary spatial relationships. The solution is obtained by solving an ordinary linear program, and represents a high-quality approximation of the global optimum of the original classification problem.

The strengths of our approach are the following:

1. It is highly generic in the sense that it can be easily extended to other detectors and descriptors with different properties concerning robustness and distinctiveness. It will exploit the strenghts of its operators by design.

2. The solution has a clear statistical interpretation as a binary MAP classification of putative matches into inliers and outliers, given the observations and independence structure described in Section 4.1.4. Therefore both the problem formulation and the result have clear semantics.

3. It requires only a minimal number of parameters to be specified, once that the likelihood distributions are determined from training data. In particular, we only define a maximum number of putative matches to restrict the complexity. This number is not critical, as today's linear programming solvers can easily handle several hundreds of putative matches, and the focus is on scenes with sparse feature sets.

Implementation of the approach is straightforward: First, the parameters of the potential functions are estimated offline from training data. As the potentials can be approximated quite well by particularly simple functions, parameter fitting can be accomplished without effort using any of numerous available standard software packages. Then, given a particular set of putative feature correspondences, the values of the potential functions are collected for each pair of correspondences, and formatted as an ordinary linear program according to Eqs. (4.51)-(4.56). The solution is obtained using standard software for convex optimization. It can be computed in polynomial time, and gives a high-quality, stable approximation of the global optimum of the MAP estimate.

For this work, we have chosen to use Euclidean distances of SIFT descriptors, weighted distances of color histograms, and pairwise consistency of orientation angles, spatial distances, and sidedness as observations. The choice is neither restricted to these properties, nor does our approach rely on any of them. Although we obtained good matching results in our experiments, we suggest to try out other relationships in order to obtain possibly better results.

We have shown in our experiments that the proposed approach MapMatch is superior to a purely descriptor-based method in terms of the number of extracted inliers at acceptable outlier rates. In particular, it is capable of producing good sets of correspondences for features with rather weak descriptors by exploiting spatial relationships. Especially for difficult datasets with very sparse texture, low image overlap, or low image resolution, we obtained a higher number of inliers at acceptable outlier rates with our method.

When combining weak and strong feature types, our framework produces effects similar to the explicit topological boosting proposed by Bay et al. (2005): Feature correspondences with rather bad descriptor similarity are explicitely motivated by consistent geometric relationships. A unique property of our approach is that this effect arises naturally from the statistics of the observed data, instead of being forced.

We do not claim that the proposed approach yields generally better results than that of Bay et al. (2005), although it outperformed our own implementation of Bay's framework on most datasets. It must be expected that the original implementation of Bay's method yields different, probably better results on some datasets. The original motivation for our approach was not to outperform the procedure of Bay et al. (2005), but to give a statistically motivated and more intuitive formulation, that is also applicable to a broader range of setups.

We want to emphasize that the choice of detectors is crucial for the final success. Even more, we think that the most important aspect for handling sparsely textured scenes is the choice of a highly complementary set of robust detectors (Dickscheid et al., 2010).

When using the image orientation system aurelo in our experiments, we have seen that the quality of a matching algorithm cannot fully compensate the choice of a weak detector. However, the proposed approach can deal significantly better with weak detectors and descriptors than other approaches that we investigated.

Preselecting putative matches based on descriptor similarity is the most heuristic part of our approach. The ROC statistic for different values of k in the BestMatch-k method (Figure 2.4) indicates that preselecting only the nearest neighbors ignores some of the true positives. Probably the use of more sophisticated criteria for making a preselection would lead to better matching results. One suggestion is to learn the best preselection from annotated data. An investigation of different preselection criteria would be interesting.

To the best of our knowledge, an empirical analysis of the BestMatch-k method over different feature detectors and values k, as presented in Section 2.4, has not been carried out elsewhere. Repeating this experiment with a more general setup of detectors and descriptors might provide valuable results for feature matching applications.

We have seen that simple descriptor-based matching is faster than our approach and still effective in case that many features are available. Therefore we propose to fall back to this standard method when the amount of detected features is high, and use our method only if the amount is low. The same certainly holds for the selection of feature detectors: In scenes with strong texture, it is usually sufficient to use a single good detector, as for example lowe, mser or sfop0. Such a decision however requires a preprocessing step which analyzes the texture properties of an image, and then selects the detectors and methods accordingly.

In case of sparsely textured scenes, the localization accuracy of features plays an important role for image orientation, because the overall number of features is very low. We therefore recommend to take localization accuracy into account when deciding on the set of detectors. Some authors have focussed on this aspect recently (Haja et al., 2008; Zeisl et al., 2009;

Remondino, 2006). In scenes with strong texture, on the other hand, bad localization accuracy is often compensated by the high redundancy of observations.

It would be interesting to investigate in how far the real-valued solution of the relaxed LP problem – before applying the rounding scheme – can be used in subsequent processing steps. It is probably a good indicator for the quality of matches, and could be used to trigger priority-driven algorithms, for example a quality-driven sample selection in a RANSAC scheme. In a similar manner, the energy of the LP solution might be of interest, as it potentially indicates the quality of the complete image-to-image matching. For example, considering the energy relative to the number of involved variables may be a reasonable indicator for image overlap.

One may obtain better matching results when choosing better approximations of the empirical likelihood distributions that lead to the energy potentials. In particular, one could replace the Beta and Binomial distributions used here by more complex distributions, or even introduce mixture models for the likelihood. This would potentially improve the results, but at the same time lead to more complex implementations. It may also be interesting to put a weighting on the different likelihood functions, reflecting the ability of each type of observation to separate good from bad matches. We performed some experiments by estimating Fisher's discriminant score for each likelihood distribution, and use it as a weighting factor. At the time of writing however, we did not obtain improved results from such weightings.

Altogether, the proposed method offers a highly generic yet intuitive framework for implementing robust feature matching algorithms. It provides a clear distinction between data-dependent elements, namely the energy potentials, and algorithmic parts, and is therefore easily applicable to different matching problems. Implementation of the algorithmic parts can be mostly covered by existing standard software packages. By choosing appropriate training datasets for the energy potentials, it is possible to implement both multi-purpose and highly specialized matching algorithms. While we have used the framework to implement a rather general algorithm, and focussed our experiments on standard datasets and man-made scenes with poor texture, it would be interesting to analyze the behavior of more specialized implementations, for example in the context of medical image registration and object recognition.

Appendix A

Image Datasets

A.1 Image Pairs Used for Annotation

Image pairs used for learning the potential functions. The images show indoor and outdoor architectural scenes with both sparse and significant texture. The outdoor images are taken from the *fountain-P11* and *Herz-Jesu-P8* datasets (Strecha et al., 2008).

A.2 Images of the Blank-12 Dataset

1

2

3

4

5

6

7

8

9

10

11

12

A.3 Images of the Blank-22 Dataset

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

A.4 Images of the Graffiti Dataset

1

2

3

4

5

6

This dataset is taken from Mikolajczyk et al. (2005).

A.5 Images of the Boat Dataset

1

2

3

4

5

6

This dataset is taken from Mikolajczyk et al. (2005).

A.6 Images of the Class Dataset

13

14

15

16

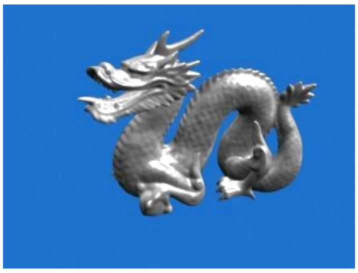
17

18

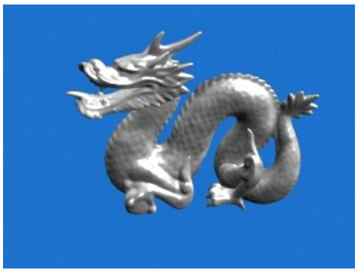
19

20

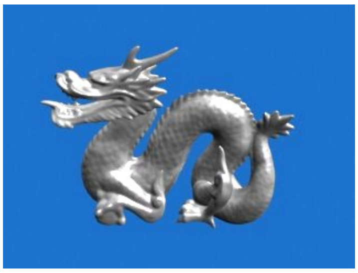
A.7 Images of the Dragon Dataset



1



2



3

4

5

6

7

The 3D model of the dragon used for rendering these images is taken from the *Stanford 3D Scanning Repository* at `http://graphics.stanford.edu/data/3Dscanrep/`. It was first presented by Curless and Levoy (1996). Note that the brightness of the images has been increased to 150% for this figure.

Bibliography

Aguilar, W., Y. Frauel, F. Escolano, M. Martinez-Perez, A. Espinosa-Romero, and M. Lozano (2009). A Robust Graph Transformation Matching for Non-Rigid Registration. *Image and Vision Computing 27*(7), 897–910.

Bay, H., V. Ferrari, and L. V. Gool (2005). Wide-Baseline Stereo Matching with Line Segments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Volume 1, Washington, DC, USA, pp. 329–336.

Bergholm, F. (1987). Edge Focusing. *IEEE Trans. Pattern Anal. Mach. Intell. 9*(6), 726–741.

Bigün, J. (1990). A Structure Feature for Some Image Processing Applications Based on Spiral Functions. *Computer Vision, Graphics and Image Processing 51*(2), 166–194.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Information Science and Statistics. Springer.

Boyd, S. and L. Vandenberghe (2004). *Convex Optimization.* New York, NY, USA: Cambridge University Press.

Canny, J. F. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*(6), 679–698.

Chekuri, C., S. Khanna, J. S. Naor, and L. Zosin (2001). Approximation Algorithms for the Metric Labeling Problem via a new Linear Programming Formulation. In *ACM-SIAM Symposium on Discrete Algorithms*, Philadelphia, PA, USA, pp. 109–118. Society for Industrial and Applied Mathematics.

Choi, O. and I. S. Kweon (2009). Robust feature point matching by preserving local geometric consistency. *Computer Vision and Image Understanding 113*(6), 726–742.

Curless, B. and M. Levoy (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, New York, NY, USA, pp. 303–312. ACM.

Delponte, E., F. Isgrò, F. Odone, and A. Verri (2006). SVD-matching using SIFT features. *Graphical models 68*(5-6), 415–431.

Dickscheid, T. and W. Förstner (2009). Evaluating the Suitability of Feature Detectors for Automatic Image Orientation Systems. In *Proceedings of the International Conference on Computer Vision Systems*, Liege, Belgium, pp. 305–314.

Dickscheid, T., F. Schindler, and W. Förstner (2010). Coding Images with Local Features. *International Journal of Computer Vision*, 1–21. 10.1007/s11263-010-0340-z.

Ferrari, V. (2004). *Affine Invariant Regions++*. Ph. D. thesis, Technische Wissenschaften, Eidgenössische Technische Hochschule ETH Zürich.

Fischler, M. A. and R. C. Bolles (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM 24*(6), 381–395.

Förstner, W. (1994). A Framework for Low Level Feature Extraction. In *Proceedings of the European Conference on Computer Vision*, Volume III, Stockholm, Sweden, pp. 383–394.

Förstner, W., T. Dickscheid, and F. Schindler (2009). Detecting Interpretable and Accurate Scale-Invariant Keypoints. In *Proceedings of the IEEE International Conference on Computer Vision*, Kyoto, Japan.

Förstner, W. and E. Gülch (1987, June). A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features. In *ISPRS Conference on Fast Processing of Photogrammetric Data*, Interlaken, pp. 281–305.

Haja, A., B. Jähne, and S. Abraham (2008, June). Localization Accuracy of Region Detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hammersley, J. M. and P. Clifford (1971). Markov Field on Finite Graphs and Lattices. `http://www.statslab.cam.ac.uk/~grg/books/hammfest/hamm-cliff.pdf`.

Harris, C. and M. J. Stephens (1988). A Combined Corner and Edge Detector. In *Proceedings of the Alvey Vision Conference*, pp. 147–152.

Hartley, R. I. and A. Zisserman (2004). *Multiple View Geometry in Computer Vision* (Second ed.). Cambridge University Press, ISBN: 0521540518.

Heuel, S. (2004). *Uncertain Projective Geometry: Statistical Reasoning For Polyhedral Object Reconstruction (Lecture Notes in Computer Science)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. In *STOC '84: Proceedings of the sixteenth annual ACM symposium on Theory of computing*, New York, NY, USA, pp. 302–311. ACM.

Kolmogorov, V. and C. Rother (2007). Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Trans. Pattern Anal. Mach. Intell. 29*(7), 1274–1279.

Kolmogorov, V. and R. Zabih (2004). What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell. 26*(2), 147–159.

Kumar, M. P., V. Kolmogorov, and P. H. S. Torr (2009). An analysis of convex relaxations for map estimation of discrete mrfs. *J. Mach. Learn. Res. 10*, 71–106.

Läbe, T. and W. Förstner (2006, March). Automatic Relative Orientation of Images. In *Proceedings of the 5th Turkish-German Joint Geodetic Days*, Berlin.

Li, S. Z. (2009). *Markov Random Field Modeling in Image Analysis* (2 ed.). Springer.

Lindeberg, T. (1998). Edge Detection and Ridge Detection with Automatic Scale Selection. *International Journal of Computer Vision 30*(2), 117–156.

Lourakis, M. A. and A. Argyros (2009). SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software 36*(1), 1–30.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision 60*(2), 91–110.

Matas, J., O. Chum, M. Urban, and T. Pajdla (2004, September). Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *Image and Vision Computing 22*, 761–767.

Mayer, H. (2005). Robust Least-Squares Adjustment Based Orientation and Auto-Calibration of Wide-Baseline Image Sequences. In *ISPRS Workshop BenCOS 2005*, Bejing, China, pp. 11–17.

Meidow, J., C. Beder, and W. Förstner (2009). Reasoning with uncertain points, straight lines, and straight line segments in 2d. *ISPRS Journal of Photogrammetry and Remote Sensing 64*(2), 125 – 139.

Meltzer, J. and S. Soatto (2008). Edge Descriptors for Robust Wide-Baseline Correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

Mikolajczyk, K. and C. Schmid (2004). Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision 60*(1), 63–86.

Mikolajczyk, K. and C. Schmid (2005). A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell. 27*(10), 1615–1630.

Mikolajczyk, K., T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool (2005). A Comparison of Affine Region Detectors. *International Jounal of Computer Vision 65*(1/2), 43–72.

Moreels, P. and P. Perona (2006). Evaluation of Features Detectors and Descriptors based on 3D Objects. In *International Journal of Computer Vision.*

Nister, D. (2004). An Efficient Solution to the Five-Point Relative Pose Problem. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 26, Washington, DC, USA, pp. 756–777. IEEE Computer Society.

Pilu, M. and A. Lorusso (1997). Uncalibrated Stereo Correspondence by Singular Value Decomposition. In *British Machine Vision Conference*, Essex.

Pollefeys, M., R. Koch, M. Vergauwen, and L. Van Gool (2000). Automated Reconstruction of 3D Scenes from Sequences of Images. In *ISPRS Journal Of Photogrammetry And Remote Sensing*, Volume 55(4), pp. 251–267.

Ravikumar, P. and J. Lafferty (2006). Quadratic Programming Relaxations for Metric Labeling and Markov Random Field MAP Estimation. In *International Conference on Machine Learning*, New York, NY, USA, pp. 737–744. ACM.

Remondino, F. (2006). Detectors and descriptors for photogrammetric applications. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XXXVI, Bonn, Germany, pp. 49–54.

Rossi, F., P. v. Beek, and T. Walsh (2006). *Handbook of Constraint Programming (Foundations of Artificial Intelligence).* New York, NY, USA: Elsevier Science Inc.

Roth, D. G. (2004, July). Automatic Correspondences for Photogrammetric Model Building. In *Proceedings of the XXth ISPRS Congress*, Istanbul, Turkey, pp. 713–718.

Schellewald, C. and C. Schnörr (2005). Probabilistic Subgraph Matching Based on Convex Relaxation. In *Proc. Int. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR'05)*, Volume 3757, pp. 171–186. Springer.

Schlesinger, M. (1976). Sintaksicheskiy analiz dvumernykh zritelnikh signalov v usloviyakh pomekh (Syntactic Analysis of Two-Dimensional Visual Signals in Noisy Conditions). *Kibernetika 4*, 113–130.

Schmid, C. and R. Mohr (1997). Local Grayvalue Invariants for Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 530–535.

Scott, G. and H. Longuet-Higgins (1991). An Algorithm for Associating the Features of Two Patterns. In *Proc. Royal Soc. London*, Volume B244.

Shapiro, L. G. and R. M. Haralick (1987). Relational matching. *Appl. Opt. 26*(10), 1845–1851.

Snavely, N., S. M. Seitz, and R. Szeliski (2006). Photo tourism: exploring photo collections in 3d. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, New York, NY, USA, pp. 835–846. ACM.

Strecha, C., W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA, pp. 1–8. IEEE Computer Society.

Strecha, C., W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen (2008). On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, Anchorage, Alaska.

Szeliski, R., R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother (2008). A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 1068–1080.

Tell, D. and S. Carlsson (2002). Combining Appearance and Topology for Wide Baseline Matching. In *European Conference on Computer Vision*, Copenhagen, pp. 68–81. Springer.

Torresani, L., V. Kolmogorov, and C. Rother (2008). Feature correspondence via graph matching: Models and global optimization. In *European Conference on Computer Vision*, pp. 596–609. Springer.

Tuytelaars, T. and K. Mikolajczyk (2008). *Local Invariant Feature Detectors: A Survey.* Hanover, MA, USA: Now Publishers Inc.

Ullman, S. (1979). *The interpretation of visual motion.* Cambridge, MA: MIT Press.

Vergauwen, M. and L. V. Gool (2006). Web-based 3d reconstruction service. *Mach. Vision Appl. 17*(6), 411–426.

Wainwright, M. J. and M. I. Jordan (2008). Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn. 1*(1-2), 1–305.

Zeisl, B., P. Georgel, F. Schweiger, E. Steinbach, and N. Navab (2009). Estimation of Location Uncertainty for Scale Invariant Feature Points. In *Proceedings of the British Machine Vision Conference*, London, UK.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision 13*(2), 119–152.