CENTER FOR
MACHINE PERCEPTION

CZECH TECHNICAL
UNIVERSITY IN PRAGUE

PhD THESIS

ISSN 1213-2365

Probabilistic Models for Symmetric Object Detection in Images

Radim Tyleček

tylecr1@cmp.felk.cvut.cz

CTU–CMP–2015–07

November 30, 2015

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/tylecek/Tylecek-PhD2015.pdf

Thesis Advisor: Radim Šára

The author was supported by the Czech Science Foundation under
Project P103/12/1578.

Research Reports of CMP, Czech Technical University in Prague, No. 7, 2015

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Cybernetics

PROBABILISTIC MODELS FOR SYMMETRIC OBJECT DETECTION IN IMAGES

Doctoral Thesis

Radim Tyleček

Prague, November 2015

Ph.D. Programme: Electrical Engineering and Information Technology

Branch of study: Artificial Intelligence and Biocybernetics

Supervisor: Radim Šára

Contents

Abstract

This thesis deals with application of symmetry principles to computer vision problems of object detection in images. The focus is put on the ways how our prior knowledge on translation, reflection and rotation symmetries can be encoded in probabilistic models. Conceptually the position of our object-centered approach lies between general symmetry detection and strongly informed procedural modeling.

In particular we present two methods for parsing of facade images, where translation symmetry manifests in the structure of architectural elements like windows, doors and cornices. In both cases the structural model is based on local interactions between objects and the symmetry is represented in the spirit of Gestaltian grouping principles of proximity, similarity and continuity.

The initial method Weak Structure Model uses efficient random sampling to infer the most probable configuration of windows. Experimental results suggest that a simple data model accompanied with appropriate symmetry prior can outperform other methods with more specific window classifiers.

The next approach called Spatial Pattern Templates aims to learn the important relations of the facade structure beforehand rather than inferring it at inference time like in the previous case. This process is facilitated by conditional random field framework, where powerful training methods are available. We have also found that the available datasets cannot provide a number of samples sufficient for such training. We have resolved this obstacle by assembly of a rich and large CMP Facade Database, which is now available to other researchers.

The last method explores the remaining reflection and rotation symmetries. At this time the Bayesian inference is used to handle a hierarchical model extending from the low-level geometry of reflection symmetry to dihedral symmetry groups. Objectness and compactness priors are included to reduce ambiguity in the detection. The increased complexity of the model is compensated by utilization of an advanced inference method, which allows to rigorously reason about number of detected components by means of model selection. In result we show this approach improves performance on standard datasets, particularly in the case when multiple objects are present.

Anotace

Tato práce se zabývá aplikací principů symetrie na problémy počítačového vidění jako je detekce objektů v obrazech. Zaměřuje se na způsoby jakými lze do pravděpodobnostních modelů zakódovat naší znalost o translační, osové a rotační symetrii. Naš přístup založený na objektech koncepčně leží mezi obecnými metodami pro detekci symetrií a silně informovaným procedurálním modelováním.

Konkrétně představujeme dvě metody pro analýzu obrazů fasád domů, kde se translační symetrie projevuje na struktuře architektonických prvků jako jsou okna, dveře a římsy. V obou případech je model struktury založen na lokální interakci mezi objekty a symetrie je reprezentované ve smyslu Gestaltovských shlukovacích pravidel pro blízkost, podobnost a návaznost.

Úvodní metoda se slabým strukturním modelem používá efektivní náhodné vzorkování pro nalezení nejpravděpodobnější konfigurace oken. Experimentální výsledky naznačují, že i jednoduchý datový model doplněný vhodným apriorním modelem může překonat jiné metody využívající specifických klasifikátorů oken.

Následující přístup založený na šablonách prostorových vzorů si klade za cíl se předem naučit významné vztahy mezi prvky fasád, narozdíl od předchozího, kde je toto součástí vzorkování. Učení je zprostředkováno použitým podmíněným náhodným polem, pro které jsou k dispozici účinné metody pro trénování. Přitom jsme zjistili, že dostupné datasety neobsahují dostatečný počet vzorků pro trénování. Tuto překážku jsme odstranili sestavením vlastní databáze fasád, která je nyní dostupná ostatním výzkumníkům.

Závěrečná metoda zkoumá osovou a rotační symetrii. V tomto případě je použita Bayesovská inference pro hierarchický model sahající od geometrie osové symetrie na nízké úrovní až po dihedrální symetrické grupy. Objektovost a kompaktnost jsou přitom použity jako apriorní vlastnosti pro snížení nejednoznačnosti při detekci. Vyšší komplexnost modelu je kompenzována využitím pokročilých inferenčních algoritmů, které umožňují rigorozně odvodit počet nalezených komponent výběrem správného modelu. Výsledky ukazují že tento přístup zvyšuje přesnost na standartních datasetech, zejména v případech kdy se v obraze nachází více objektů.

Acknowledgments

Primarily I would like to gratefully thank my supervisor prof. Radim Šára for continuous support and patience during making of this thesis. His wise guidance and critical comments have directed my research since my master thesis.

I would also like to express my thanks to prof. Vašek Hlaváč for creating and maintaining Center for Machine Perception as a friendly and inspiring environment for research. His hints towards finishing the thesis have been useful.

Many thanks go to prof. Mirko Navara, whose detailed comments significantly helped to improve the quality of the final text.

Finally this all would be hardly accomplished without the inspiration from my friends in the CMP and encouragement from my family.

Chapter 1

Introduction

"It is the harmony of the diverse parts, their symmetry, their happy balance; in a word it is all that introduces order, all that gives unity, that permits us to see clearly and to comprehend at once both the ensemble and the details."

Henri Poincare (1854-1912)

Symmetry is a natural phenomenon and our visual perception system learned to use it as a guide to explain what we see. Particularly in the cases when the observed scene is ambiguous the reasoning tends to prefer explanations which follow some innate prior principles. Psychologists in their research on human perception came up, among other concepts, with *principles of grouping* also known as *Gestalt laws* (Goldstein, 2009). The observation that humans naturally perceive objects as organized patterns and objects can be then explained with a set of principles such as proximity, similarity, continuity or symmetry. When we extend the narrow meaning of symmetry from reflection and include rotation and translation isometries as in geometry, we can cover many of these principles with a single general term – *symmetry*.

In analogy these principles are used in computer vision, where pattern recognition methods facilitate image understanding. In this context symmetry has been applied at all levels of processing, from low-level features to 3D models, and also validated as a useful *regularizer* in difficult inference tasks. In this role of a prior a range of applications opens, but at the same time the mechanism encoding the prior knowledge and its seamless integration in the model become equally important.

Simultaneous symmetry detection has become a discipline of its own, where researchers foster their methods in an effort to deliver a reliable, widely usable and effective *feature* detection technique impacting object recognition. Although considerable progress has been made over the decades, such a universal symmetry detector is still not available today.

1.1 Overview

Rather than attempting the universal symmetry detector problem, this thesis is focused more on the regularizing aspect of symmetry to computer vision problems, particularly in object detection and image parsing.

While **Bayesian framework** suits the task of prior knowledge integration naturally, it has been sparsely applied to symmetry in practice, mostly in favor of approaches defining energy functions with a form suitable to a particular optimization method. With recent advances and new methods available to implement Bayesian inference we can relax our limits on the model complexity and maintain practical tractability at the same time. We will construct **probabilistic models** to capture the essence of symmetry in the spirit of the above mentioned Gestalt principles, and make use of the added value the Bayesian framework delivers.

More specifically we will address the problem of facade image parsing with this approach, where **translation symmetry** is dominant. The world of facades is sufficiently rich in complexity of structure to be challenging while reasonably limited for analysis. In particular, we will develop methods where priors act locally and allow some degree of flexibility. The Bayesian approach will help us to resolve the underlying problem of how many objects are present. The problem of **variable number of objects** is inherent to translation symmetry, which makes it more prominent than in general object detection.

A next task in the same area is to come up with a method which is able to **learn the structure** of relations between translation symmetric objects. We will also publish a new dataset which is sufficiently large and rich for training purposes.

Based on the experience gained with simpler models we will finally construct a **hierarchical model** to deal with 'classical' reflection symmetry detection both at low-level (geometry) and high-level (component and group priors). At a high level the remaining elementary 2D symmetry, rotation, will be also used to constrain the detection with dihedral groups.

1.2 State of the Art

The goal of this section is to analyze existing methods in symmetry detection and its application to regularity modeling in computer vision. We will first give a brief introduction to symmetry, its types, groups and related concepts.

1.2.1 Symmetry Concepts

From the broad range of results accumulated in symmetry theory, we will go through the basic concepts relevant to this thesis. Let us start with the formal definition of geometric symmetry (Liu et al., 2010):

Let $S \subset \mathbb{R}^n$ be an object and g be an isometric (distance preserving) mapping g. We say S has a *symmetry* g if and only if $g(S) = S$ (automorphism). In other words the object S has invariance under the transform g. The S can be a point set, intensity or color image, surface etc., and the symmetry is its property. Note that identity is the trivial symmetry.

1.2.1.1 Primitive Symmetry Types in 2D

A *primitive symmetry* of S is atomic, i.e. it cannot be decomposed as a concatenation of two non-trivial symmetries of a different type. There is a fixed set of primitive symmetry types for a given dimension and metric. In the simplest 1D case there are *reflection* and *translation*[1] only. In Euclidean 2D space we add *rotation* and *transflection* to get four primitive symmetry types. Extending to 3D we get the helical and rotoreflection primitive symmetry types and there are more of them appearing in the higher dimensions. Hyperbolic spaces house similar primitive symmetry types analogical to Euclidean ones.

Since we are interested in image analysis, we will restrict ourselves to Euclidean 2D space and its four primitive symmetry types illustrated in Fig. 1.1, considering an image function $f(\mathbf{x})$ and a point $\mathbf{x} = (x_1, x_2)$.

Translation symmetry is defined with

$$f(\mathbf{x}) = f(\mathbf{x} + \Delta), \tag{1.1}$$

where $\Delta \in \mathbb{R}^2$ is the translation vector.

Reflection symmetry (also called bilateral or mirror) is essentially defined for the case of reflection w.r.t. axis x_2 with

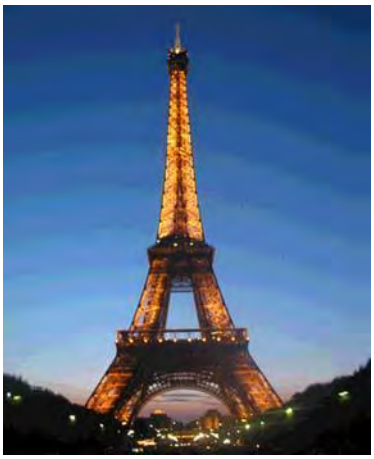
$$f(\mathbf{x}) = f((-x_1,\, x_2)). \tag{1.2}$$

In the general case we can parameterize such transformation with

$$f(\mathbf{x}) = f(\mu + \mathbf{R}(\mathbf{x} - \mu)), \tag{1.3}$$

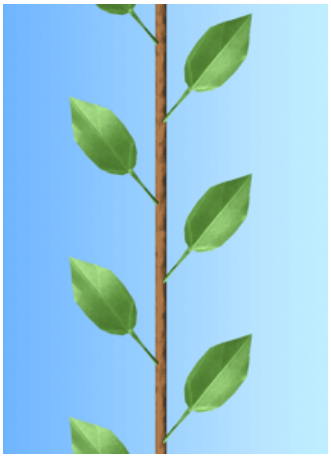
[1] Only some infinite sets are translation invariant.

a) Translation



b) Reflection

c) Rotation



d) Transflection

Figure 1.1: Objects with primitive symmetries in 2D demonstrated on real-world examples from symmetry datasets (except d). The objects shown in the images are only approximately invariant under the specified symmetry mappings.

a) Cyclic b) Dihedral c) Orthogonal

Figure 1.2: Symmetry groups in 2D with point invariance. Images from Liu et al. (2010).

where $\mu \in \mathbb{R}^2$ is the axis location and $\mathbf{R} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$ is the Householder reflection matrix for axis orientation $\mathbf{u} = (\cos\varphi, \sin\varphi)$.

Rotation symmetry is defined similarly with

$$f(\mathbf{x}) = f(\mu + \mathbf{F}(\mathbf{x} - \mu)), \tag{1.4}$$

where $\mu \in \mathbb{R}^2$ is the rotation center and

$$\mathbf{F} = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \tag{1.5}$$

is the rotation matrix for angle $\varphi = \frac{2\pi}{n}$. The integer $n \in \mathbb{N}$ is the order (fold) of rotation.

Transflection symmetry (also called glide) is a combination of partial translation and reflection. In the case of transflection w.r.t. axis x_2 it is defined as

$$f(\mathbf{x}) = f((-x_1,\, x_2 + \delta)), \tag{1.6}$$

where $\delta \in \mathbb{R}$. Contrary to the intuition this is also a primitive symmetry, because neither the given individual translation nor the reflection mapping is a symmetry of the whole object S. This type of symmetry is rare in practice and rather specific, we will not consider the transflection further.

1.2.1.2 Symmetry Groups

An interesting observation with extensive theoretical implications is that there are special symmetry sets G where symmetries $g \in G$ are compatible or complementary to each other in such a way that their compositions give the same result w.r.t. certain object $S = g(S)$.

Formally we define a *symmetry group* G of S as a mathematical group $\{G, *\}$ closed under transformation composition ($g_1 * g_2 \in G$ for all $g_1 \in G$, $g_2 \in G$ and by composition we mean chaining $g_1(g_2(S)) = (g_1 * g_2)(S) = g(S)$). The symmetry groups can be essentially characterized by discreteness, finiteness and invariance. These properties will be described on examples in Fig. 1.2.

Cyclic group C_n is formed by n rotation symmetries of order $n \in \mathbb{N}$, i.e. $\varphi_i = \frac{2\pi}{n} i$, $i = 0, \ldots, n-1$. Non-trivial C_n is a finite discrete group with a rotation center as the invariant

point[2]. The degenerate case of $n = 1$ is just identity.

Dihedral group D_n is formed by rotation of order n combined with n reflections, otherwise its characterization is the same as for the cyclic group.

Orthogonal group $O(2)$ is the limiting case of D_n when $n \to \infty$. It is an infinite continuous group of an unoriented disk with rotation center as the invariant point. It contains infinitely many rotations and reflections w.r.t. to a given invariant point.

An important class of *crystallographic groups* is found in periodic patterns repeated along some dimensions of the given space. There is a finite number of such distinct symmetry groups in any Euclidean space and there is a compositional structure (hierarchy) among them. Their invariant is a space unit delimited by the repetition period in $\mathbb{R}^n$. There are 24 crystallographic groups in 2D and even 230 in 3D. In the 2D case relevant for us there are further two following subclasses, also see Fig. 1.3. In practice we understand images capturing finite objects as 'cropped out' of an infinite pattern.

Frieze groups are *strip* patterns repeating along one dimension in 2D. There are seven discrete infinite groups formed by compositions of 1D translation with rotation (order $n = 2$), reflection (horizontal or vertical) or transflection. Fig. 1.3c shows example of group called *ml* by crystallographers, composed of translation+reflection.

Wallpaper groups are *lattice* patterns repeating in two dimensions in 2D, generated by two linearly independent vectors, which simultaneously define the lattice unit and tiling. There are 17 discrete infinite groups formed by compositions of 2D translation with rotation (orders $n = 2, 3, 4, 6$), reflection (horizontal or vertical) or transflection. Fig. 1.3d shows an example of translation+reflection group called *pmm*.

1.2.1.3 Symmetry in Images

Images of real-world objects captured by projective cameras are generally a result of perspective transformation. This causes objects with symmetry patterns to appear deformed in the images unless the camera is specifically restricted, i.e. when it is orthographic (or perspective) and fronto-parallel oriented w.r.t. planar surface of an observed object.

It is often sufficient to consider affine transformations only and to define *skewed symmetry groups* as affinely transformed Euclidean symmetry groups (Liu et al., 2010). In this affine (and also projective) space the original Euclidean symmetries are related by affine transformations and form a hierarchy.

While the symmetry patterns can be significantly deformed by perspective projections as in Fig. 1.4, there are certain characteristics called *invariant features*, which are not affected by the projection. Invariant features can be used for detection (Gool, 1998).

[2]Precisely the case of $i = 0$ is identity with plane invariance, but this includes the given rotation point invariance as well.

a) Cyclic C_4

b) Dihedral D_5

c) Frieze *ml*

d) Wallpaper *pmm*

Figure 1.3: Discrete symmetry groups in 2D demonstrated on real-world examples from various symmetry datasets. The objects shown are only approximately invariant under the specified symmetry mappings.

Figure 1.4: Affine projections of wallpaper symmetry group *pmm* in a real-world image.

1.2.2 Symmetry Detection

Symmetry detection is an important computer vision problem (Davis, 1977), which has been used to constrain other problems such as recognition (Hayfron-Acquah et al., 2003), retrieval (Lee, 2013) and reconstruction (Yang et al., 2005; Sinha et al., 2012). A survey by Liu et al. (2010) provides background to general symmetry detection from images and reviews some related methods.

A most basic method for symmetry detection, also called *direct approach*, is a straightforward implementation of symmetry definition: Apply symmetry transformation g to image S and compare the result $g(S)$ with the original S, i.e. using SSD[3] measure, and decide whether symmetry is present. When the symmetry parameters are unrestricted, the pool of tentative symmetries g has to be large, which is computationally demanding (recursive multi-resolution strategy can help). In practice this approach will work only with perfectly symmetric (artificial) images and fail in the presence of background clutter, appearance changes, partial occlusion or noise common to real-world images.

Symmetry detection methods can be characterized primarily by the scale at which they operate: *Local* symmetries are supported only by a subset of the image or shape, in contrast a *global* symmetry explains entire shape or even image. As the title of this thesis suggests we will focus on global symmetries, which can be attributed to one or more objects in the image.

1.2.3 Reflection and Rotation

The first reference to an algorithmic treatment of bilateral reflection symmetry by Birkhoff (1932) goes back even before computer vision itself was established. Over the decades a number of algorithms has been proposed for different types of symmetry, for a comprehensive overview we forward the reader to the survey by Liu et al. (2010).

Some of the more theoretical results in global symmetry detection, such as basis function (i.e. RBF) and moment-based methods (Marola, 1989) turned impractical for real-world images. A modern approach based on matching of local SIFT features proposed by Loy and Eklundh (2006) is now considered a baseline method. Recently, methods based on different features such as image edges (Wang et al., 2014) have been proposed.

The standard inference technique used for symmetry detection however remains *voting* in Hough or similar space, i.e. every two keypoints determine a reflection symmetry axis and if a symmetry test (geometry, similarity) is passed they cast a vote into a bin given by the axis parameters. Symmetry instances are retrieved from maximal peaks in the voting space accumulator, where thresholding and non-maximal suppression are used to avoid false positives and multiple detections. Corresponding parameters and voting space discretization choices are mostly empirical and their optimal tuning for images with multiple instances of symmetry is difficult. The discrete nature of the binning also does not allow for exact estimation of the symmetry parameters.

[3] Sum of Squared Differences, $\mathrm{SSD}(\mathbf{x}, \mathbf{y}) = \sum_i (x_i - y_i)^2$.

The only method detecting *dihedral* and *cyclic* symmetry groups known to us was presented by Lee et al. (2008). It applies polar transformation to the image with centers at all pixel locations and efficiently analyses the obtained 'frieze expansions' using DFT to determine rotation order and group. This exhaustive scheme resembles direct approach, also by requiring $\approx 10\times$ more processing time compared to Loy and Eklundh (2006).

The application of Gestalt theory for reflection symmetry detection has been investigated by Michaelsen et al. (2013), where local SIFT feature symmetries are grouped together following the continuation and proximity principles. This clustering approach however does not discard remaining local symmetries, which results in false positives when global symmetries are the goal. A general question arises from this behaviour: Where is the line between local and global symmetry and how can an algorithm distinguish them?

1.2.4 Translation

Translation symmetry detection, often found as a subgroup of repeating wallpaper patterns, is essentially described with a generating lattice (or grid in the orthogonal case). In real-world images it is usually characterized as *near-regular texture* (Liu et al., 2004), which allows deviations from the exact symmetry in both geometry and appearance. The lattice extraction can be formulated as higher order correspondence problem, where individual texture elements (texels) are detected using SIFT or correlation (Hays et al., 2006b), the search is however computationally intensive and sensitive to noise.

A more efficient algorithm for deformed lattice detection has been proposed by Park et al. (2009). It uses keypoints clustered by appearance to propose a pair of vectors generating the lattice, which initialize a regular MRF model for lattice element locations. The locations are estimated using mean-shift belief propagation followed by thin-plate spline warping.

Rather than seeing the image as continuously repeating texture with its element not clearly specified, we will be interested in the case when there are multiple instances of a known object distributed according to a lattice or similar regular layout, such as in the next section.

1.2.5 Facade Parsing

While the output from a general translation symmetry detector has limited direct use, we can make use of symmetry principles to constrain structured object detection by relaxing the wallpaper symmetry class constraints to reach a wider range of applications.

While facades as man-made scenes exhibit strong regularity and structure, when compared to arbitrary natural scenes, they still present a great variety of styles, configurations and appearance. The design of a general facade model that is able to cover their range is thus a challenging problem, and several methods have been proposed to deal with it.

There are two major approaches to the facade parsing problem. Top-down approach relies on the construction of a generative rule set, usually a grammar, and the result is obtained

stochastically as a word in the language best matching the input image (Simon et al., 2011). Automatic construction of a grammar has been proposed by Martinović and Van Gool (2013) but they do not generalize well outside of the style they were generated for, particularly due to recursive orthogonal splitting of the facade image. Learning is possible also for simple grammars like grid in Tyleček and Šára (2011b), but such model does not express more complex structural relations.

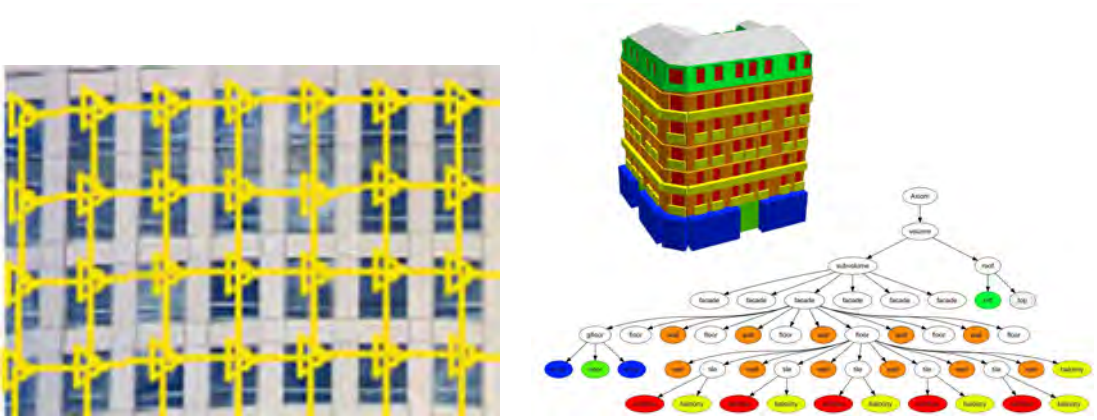
Bottom-up approaches instead combine weak general principles, which are more flexible and their parameters can be learned. The hierarchical CRF (Ladicky et al., 2009), which aggregates information from multiple segmentations at different scales, has been applied to facades in Yang and Förstner (2011), where binary potentials model consistency of adjacent labels within as well as across segmentations. Here neighboring segments with similar appearance are more likely to have the same label (contrast-sensitive Potts model). The three-staged method Martinovic et al. (2012) combines local and object detectors with a binary Potts CRF on pixels. The result is further sequentially processed to adjust the labels according to the alignment, similarity, symmetry and co-occurence principles, each of them applied with a rather heuristic procedure. Additional principles are designed for a specific dataset and in fact resemble grammatical rules.

Shape grammars, as introduced in Gips (1975) and later picked up by Zhu and Mumford (2006), are the basic essence of all recent methods based on the procedural modeling to overcome the limitations of traditional segmentation techniques. The idea of shape grammars is that an image can be explained by terminal symbols (objects) obeying a set of rules.

Some aspects of probabilistic approach were first discussed in Alegre and Dellaert (2004), including the use of RJMCMC. The proposed grammar is simple, based on splitting, and the results are demonstrated for highly regular facades only. In a similar fashion Müller et al. (2007) determines the structure by splitting the facade to a regular grid of individual tiles and subdividing them. Mayer and Reznik (2007) presented a pipeline for multi-view interpretation, where heuristics based on interest points were designed to detect positions of windows, and subsequently used MCMC to localize their borders. They also include rectification algorithm based on RANSAC to extract vanishing points from straight lines. Ripperda and Brenner (2007) has designed a comprehensive dictionary of domain-specific rules; the results presented on simple facades show this approach has difficulty to achieve good localization.

A method of Teboul et al. (2010) combines trained randomized forest classifiers with a shape grammar to segment Haussmannian[4] facades into eight classes. Their model assumes the windows form a grid while allowing different intervals. In the second step, positions of rows and columns in the grid are stochastically estimated by a specific random walk algorithm that does not propose dimension changes. Subsequently they proposed a new parser based on reinforcement learning to speed up the process in Teboul et al. (2011). In the same domain, the work of Chun and Gagalowicz (2011) demonstrates how a specific

[4] Architectural style widely used during the reconstruction of Paris in 19th century.



a) Near Regular Texture (Hays et al., 2006b)

b) Shape Grammar (Teboul et al., 2011)

Figure 1.5: General and strong approach to facade image analysis.

segmentation algorithm can be engineered for a particular regular style.

We argue that regular texture analysis (Fig. 1.5a) is too general to understand the structure in the image, because it does not uniquely specify the image element. On the other side shape grammars, particularly split-based (Fig. 1.5b), tend to be overly domain specific and restrictive or, in other words, 'strong'. Our interest lies therefore in investigating the gap between general and strong, which can be characterized with the adjective 'weak'.

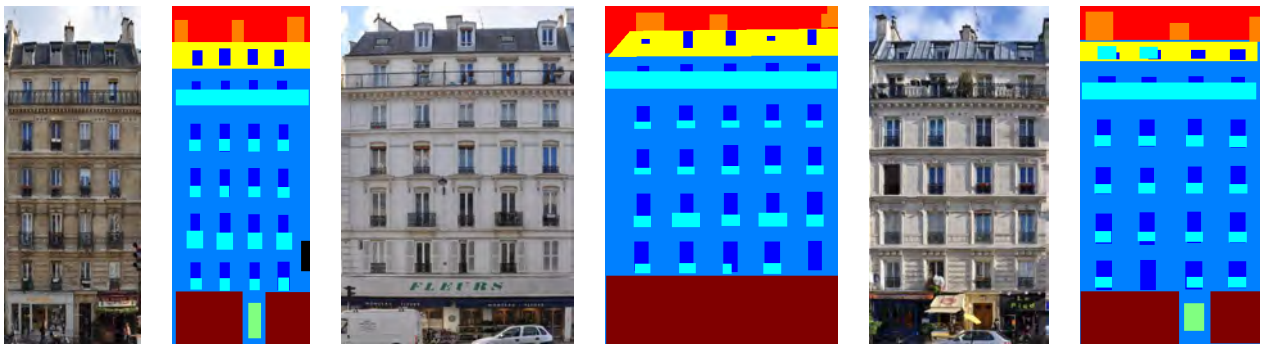
Recent development in the construction of virtual worlds like *Google Earth* or *Microsoft Bing Maps 3D* heads toward a higher level of detail and fidelity. The popularity of application such as Street View shows that reconstruction of urban environments plays an important role in this area. While acquisition of extensive data in high resolution is feasible today, their automated processing is now the limiting factor for delivering more realistic experience and it is a task for computer vision at the same time. In urban settings, typical acquired data are images of buildings' facades and their interpretation can help discover 3D structure and reduce the complexity of the resulting model; for example, it would allow going beyond planar assumptions in dense street view reconstructions presented by (Micusik and Kosecka, 2009). The work of (Pauly et al., 2008) dealing directly with structural regularity in 3D data also supports our ideas. The complexity is particularly important when the representation has to scale with the size of cities in applications such as (Hohmann et al., 2009). The fresh results of Martinovic et al. (2015a) show that depth information from 3D model helps to classify the facade surface and suggest that integration of 2D and 3D features with weak rules is promising.

1.2.6 Facade Datasets

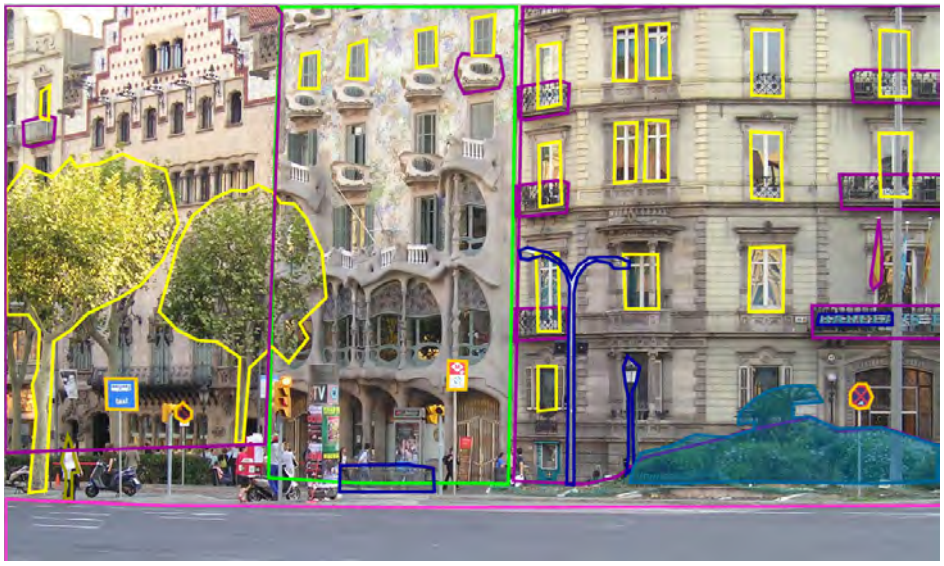
The increased interest in facade image parsing has led to introduction of several annotated datasets, which allow to quantitatively assess performance of new methods and compare their results with the previous ones. In the following we list datasets in the order they appeared and discuss their properties and relevance.

eTRIMS Dataset (Korč and Förstner, 2009) A consistent dataset of 60 non-rectified facade images was created in a dedicated project. They follow rather weak architectural principles as only sparse structure is present in the case of small houses. A small size of this dataset limits learning of structure models, which usually require more samples.

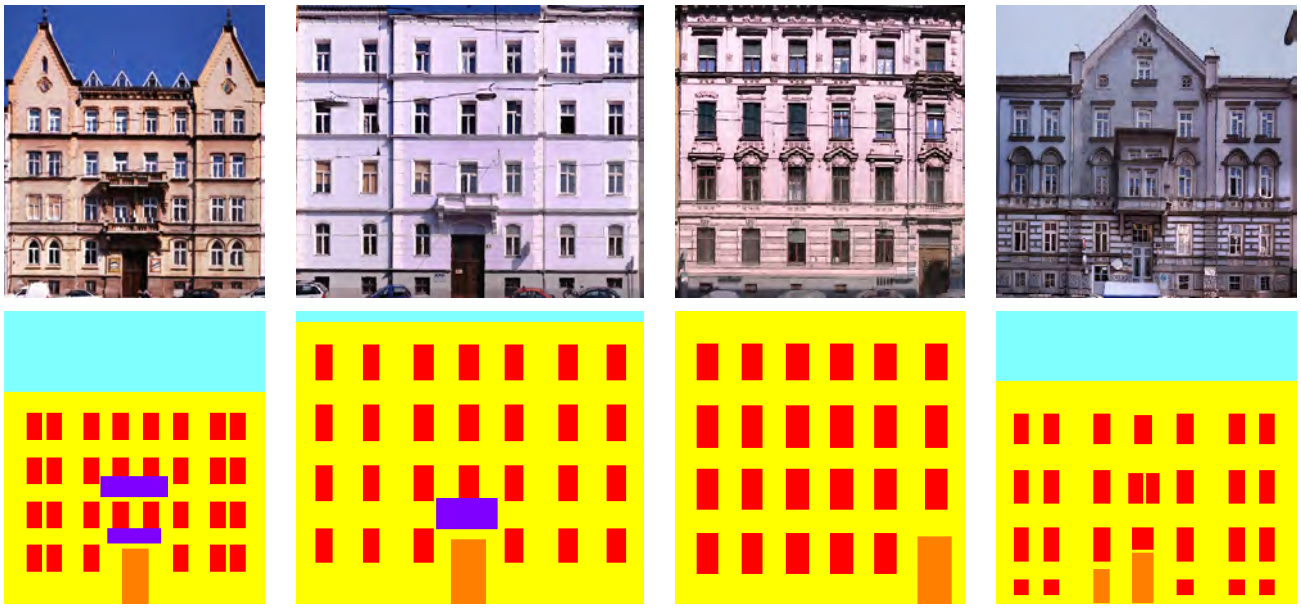
ECP Dataset (Teboul et al., 2010) A dataset of 104 Haussmannian facades from a single street in Paris (Rue Monge) is quite homogeneous. The images were rectified and background removed. This simplified setting has led to popularity among researchers and allows procedural approach to be directly applied. A revision of the initial annotation was submitted by Martinovic et al. (2012). Current methods score above 90% pixel-wise accuracy, which reaches the margin of annotation error. Recently also additional data have been captured in the same street to perform multi-view 3D reconstruction (Riemenschneider et al., 2014).



LabelMe Database (Fröhlich et al., 2010) There is a large dataset of general street images with abundance of object classes annotated (Russell et al., 2008), however only a small subset of them can be used in practice due to low consistency and completeness of facade elements annotation (randomly missing windows). A subset from this dataset was selected by Fröhlich et al. (2010) to match ECP classes in an effort to increase the number of samples for learning, but the resulting quality is not satisfactory for translation symmetry analysis.



Graz Dataset (Riemenschneider et al., 2012) This small dataset is in its form similar to ECP, but the architectural styles from Graz are varied (Classicims, Biedermeier, Historicism, Art Noveau). It is rather over-simplifying as windows out of the dominant lattice are not annotated.



Dataset	**eTRIMS**	**ECP**	**LabelMe**	**Graz**
Images	60	104	895	50
Classes	8	8	8	4
Building, Wall	●	●	●	●
Car	●			
Door	●	●	○	●
Pavement	●			
Road	●			
Sky	●	●	●	●
Vegetation	●			
Window	●	●	○	○
Balcony		●	○	
Roof		●	●	
Chimney		●	●	
Shop		●	○	

Figure 1.6: Comparison of existing facade datasets. Incomplete annotation is marked ○.

The contents of mentioned datasets is summarized in Fig. 1.6. Their analysis shows there is no dataset fulfilling the desirable properties of variability, consistency, completeness with a number of annotated images sufficient for structure learning. In reaction to this fact a new dataset fulfilling both quantitative and qualitative demands will be presented in this thesis (Chapter 3).

1.3 Preliminaries on Probabilistic Modeling and Inference

This section will present common terminology, notation and techniques related to models and methods proposed in this thesis. These are just tools with respect to this thesis and the following text discusses most popular options and does not aim to be a comprehensive enumeration of the state-of-the-art in this area.

1.3.1 Probabilistic Model

Most of the formalism presented here follows Šára (2014) and our previously published results presented in this thesis were updated to match it.

1.3.1.1 Primitive Elements

Let $X = \{x_1, \ldots, x_n\}$ represent data in a given image I. The elements of X will be called *primitive elements* or *primitives* in short. Each primitive corresponds to an independent measurable observation specific to a given problem, such as data point, correspondence or image pixel. They represent a minimal substructure (atom) participating in the inference, like a minimal sample in RANSAC. Their set and the number n are fixed.

1.3.1.2 Components

The class of problems this thesis is concerned with aims to identify unknown number k of *instances* of an object in X, such as number of clusters in clustering problems. The number k will be called *complexity* of the model. The individual object instances $j = 1, \ldots, k$ will be called model *components*.

For instance, in a standard point clustering problem the components are clusters with centers and we want to find an unknown number of clusters k. The primitives $x_i \in X$ are the individual points in $\mathbb{R}^d$, and the probabilistic model describes a point deviation from the centers.

1.3.1.3 Configurations

We assume that data X can be explained by allocating (assigning) each primitive x_i to one of the k components or to background (clutter). The primitives assigned to some component are considered *inliers*, while those assigned to background are *outliers*. We will formally consider background as the $(k+1)$-th component indexed with $j = 0$ to simplify the notation by assigning all primitives to a component. The partitioning of the set of n primitives X into $k+1$ component sets will be called a *configuration* Z with representation specific to a particular method.

1.3.1.4 Groups

A set of components can be further partitioned into subsets called *groups*, which allows to model component interaction through statistical depndencies. There are $\breve{k}$ groups in the model and allocation of components to groups is a *grouping* $\breve{Z}$.

1.3.1.5 Parameters

Let $\theta = (\dot{\theta}, \breve{\theta}, \hat{\theta}, \bar{\theta})$ be model parameters, where

$\dot{\theta}$ are *configuration parameter*s (global), e.g. component probability,

$\hat{\theta}$ are *shape parameters* common to all components, e.g. common cluster size, orientation etc.,

$\breve{\theta}$ are *group parameters*, e.g. group centers, and

$\bar{\theta} = (\bar{\theta}_1, \ldots, \bar{\theta}_k)$ are *component parameters,* e.g. cluster centers.

Parameters θ are considered random variables. Random variables associated to primitives or components will be called *attributes* where appropriate.

The following table summarizes variable notation used throughout this thesis:

Symbol	*Description*	*Domain*	*Cardinality*
θ	model parameters		
$\hat{\theta}$	common shape parameters		
$\dot{\theta}$	configuration parameters		
ξ	prior hyperparameters	continuous	
k	complexity (number of components)	discrete	
$\bar{\theta}$	component parameters	continuous	k
$\breve{\theta}$	group parameters	continuous	$\breve{k}$
$\breve{k}$	number of groups	discrete	1
I	image	continuous	
X	data	continuous	n
x_i	primitive element of data	continuous	d
Z	configuration	discrete	n
z_i	primitive allocation	discrete	
$\breve{Z}$	grouping	discrete	k
$\breve{z}_j$	group allocation	discrete	

The concrete domain will be specified in the individual models. Some symbols will be clarified in the following text.

1.3.1.6 Features

In order to simplify the model we will sometimes define *feature functions* or *features* in short, which transform attributes $\mathbf{x} = (x_1, \ldots, x_n)$ to a different (feature) space more convenient for desired modeling purposes w.r.t. the given image I. This is particularly useful in the case when it would be oversimplifying to assume independence $p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i)$. The general form for a discriminative feature $\mathbf{y} = (y_1, \ldots, y_n)$ defined as the output of a transformation function $\mathbf{f}$ is given by

$$\mathbf{y} = \mathbf{f}(\mathbf{x}; I) = \mathbf{f}(x_1, x_2, \ldots x_n), \tag{1.7}$$

$$y_i = f_i(\mathbf{x}; I). \tag{1.8}$$

We apply the chain rule for variable substitution in a pdf by inserting the determinant of the transformation Jacobian J_f to get

$$p(\mathbf{x}) = p(\mathbf{y}) \underbrace{\left| \frac{\mathrm{d}\mathbf{f}(\mathbf{x})}{\mathrm{d}\mathbf{x}} \right|}_{J_f(\mathbf{x})}, \tag{1.9}$$

assuming $\mathbf{f}$ is parametric, differentiable (smooth) and bijective. This is followed by independence assumption

$$p(\mathbf{y}) = \prod_{i=1}^{n} p(y_i) = \prod_{i=1}^{n} p(f_i(\mathbf{x})) \tag{1.10}$$

resulting in

$$p(\mathbf{x}) = J_f(\mathbf{x}) \prod_{i=1}^{n} p(y_i). \tag{1.11}$$

1.3.2 Structured Models

We will understand *structured models* as models for a set of individual objects or *components* which describe interaction between the components (their *structure* or context). By modeling their dependencies they differ from unstructured models which assume independence of the components.

Examples of some early computer vision instances of structured probabilistic models as we understand them in this thesis are the works by Moghaddam and Pentland (1995) or Schmid (1999), where spatial coherence of sets of correspondences in recognition is modeled.

1.3.2.1 Variable Number of Parts

Problems this thesis is concerned with have a common property that the number of objects present in the image or *complexity* k is not known in advance and has to be inferred from the data. This is different from a class of deformable part models with a given number of specific parts, such as in human body detection or face recognition (Felzenszwalb and Huttenlocher, 2005; Girshick et al., 2011), where parts mark eye, nose, head, torso, limbs etc.

The problem at hand is to identify k instances of an object in data X, for which we will have a parametric probabilistic model. In the trivial case there can be even no object present ($k = 0$), which case is often ignored by detection algorithms, that pick up the first best object at hand, or make the decision based on thresholding of the detection score. In general this problem is similar to estimation of a Gaussian mixture with unknown number of components (Green, 1995). In the context of computer vision the problem has been encountered i.e. in motion segmentation (Weiss and Adelson, 1996) and detecting the number of model instances is still considered one of the most difficult things in model fitting (Wang et al., 2012).

1.3.2.2 Bayesian Models and Priors

The classical approach reasons about data X by directly evaluating their statistical function $p(X \mid \theta)$, formally seen as a *likelihood* function

$$\mathcal{L}(\theta \mid X) = \log p(X \mid \theta) \tag{1.12}$$

of the unknown parameters θ with fixed X, and expressed in logarithm for convenience. However the likelihood function is not a conditional pdf w.r.t. θ and also generally it is not considered a density. Traditionally in statistics a data sample X is a set of independent (iid) sample points (measurements). If we associate X with a given intensity image the interpretation has to change slightly because each pixel measures intensity of a different part of the observed scene and the pixels are generally not iid. The segmentation of pixels into independent parts (i.e. background/foreground) is then subject to the image data model.

We can however extend the classical observation model and invert the arguments explicitly using Bayes theorem. This requires to define a *prior* distribution on parameters $p(\theta \mid \xi)$, where ξ are its own (given) parameters; these are usually called *hyperparameters* to distinguish them from the original model parameters. The actual inversion then proceeds with

$$p(\theta \mid X, \xi) = \frac{p(X \mid \theta, \xi)\, p(\theta \mid \xi)}{p(X \mid \xi)}, \tag{1.13}$$

where we can identify individual terms of a *parametric Bayesian statistical model* (Robert, 2007) as follows:

$p(\theta \mid X, \xi)$ is the *posterior* probability (density),

$p(X \mid \theta, \xi)$ is the *data* probability (density),

$p(\theta \mid \xi)$ is the *prior* probability (density),

$p(X \mid \xi)$ is the data *evidence*, in the continuous setting this term equals the marginal

$$p(X \mid \xi) = \int p(X \mid \theta, \xi)\, p(\theta \mid \xi)\, \mathrm{d}\theta \tag{1.14}$$

and can be thought as the normalizing function for the posterior.

Note that $p(X \mid \theta, \xi)$ may serve two roles – a generative model for data X given parameters θ, ξ or as likelihood of parameters θ given data X and hyperparameters ξ.

Priors generally allow to regularize the problem, which brings the final estimate of the parameters closer to the desired values, especially in the presence of noise and outliers. Priors are fundamental to structured models by encoding our knowledge on the structure.

A choice of appropriate priors is essential to Bayesian modeling and there is a significant body of literature which deals with this task. There are several general considerations.

When the posterior distribution is in the same family as the prior distribution we call it a *conjugate prior*. This convenient property is achieved when the integration (1.14) can be carried out with a closed-form result. For all exponential family distributions (Sec. 1.3.7.1) there is such conjugate prior (Gelman et al., 2003).

The hyperparameters ξ in (1.13) are considered given or fixed, but picking a single value for each of them can be suboptimal. In this case we can simply chain a new prior for these hyperparameters in our model; such prior is usually called a *hyperprior*. Let ψ be its hyperparameters, then

$$p(\theta, \xi, \psi \mid X) = \frac{p(X \mid \theta, \xi, \psi)\, p(\theta \mid \xi, \psi)\, p(\xi \mid \psi)\, p(\psi)}{p(X \mid \xi, \psi)}, \tag{1.15}$$

is a *hierarchical Bayesian model* (Robert, 2007) with hyperparameters ξ, ψ and hyperpriors $p(\xi \mid \psi)\, p(\psi)$. As suggested in (1.15) there can be naturally multiple levels in the hierarchy. Actually it is just a standard Bayesian model with a superset of parameters $\Theta = \{\theta, \xi, \psi\}$

and chain rule applied to its prior $p(\Theta)$. With respect to this observation the use of *hyper-* is just a convention and there is no strict line between 'normal' and 'hyper'. Theoretical results (Robert, 2007) suggest that a fully specified hierarchical Bayesian model (1.15) is a better estimator of the posterior distribution than one with some hyperparameters fixed (1.13).

If there is no information about a hyperparameter, *uninformative* prior may be appropriate, such as for $p(\psi)$ in (1.15). These have generally uniform distribution with no more hyperparameters, but for unbound parameters (i.e. real) a special approach proposed by statisticians is needed (Jeffrey's priors, improper priors (Gelman et al., 2003)).

1.3.2.3 Graphical Models

A *Probabilistic Graphical Model* (PGM) uses a graph to conveniently represent dependencies within a set of parameters (variables), where graph nodes correspond to the random variables (component or configuration parameters) and edges to direct probabilistic interactions between them (Koller and Friedman, 2009).

In the case of a directed graph we talk about a *Bayesian network*. An oriented edge $u \to v$ in this graph indicates conditional dependency $p(v \mid u)$. A conditional factorization requires chaining of the components, which is usually not available in two-dimensional images.

In the undirected case it is called a *Markov Random Field* (MRF). It is a generalized case of a linear Markov Chain (MC). The probability is factorized as a product of specific *potential* functions, which are usually taking the exponential form in

$$p(Z, X; \theta, \mathcal{Q}) \propto \prod_{q \in \mathcal{Q}} \exp\left(- \sum_{j \in \phi(q)} \theta_j \, \varphi_j(\mathbf{z}_q, \mathbf{x}_q) \right), \tag{1.16}$$

where $\mathcal{Q}$ is the set of cliques (complete subgraphs), φ_j are non-negative potential functions (factors) from a predefined set $\phi(q)$ defined for a clique q. The φ_j is a function of all node variables $(\mathbf{z}_q, \mathbf{x}_q)$ in a collection $\phi(q)$ and its weight is θ_j. This factorization is possible thanks to the fundamental theorem of Hammersley and Clifford (1971) which links the MRF with *Gibbs distribution* when the joint density (1.16) is strictly positive. Note that potentials φ_j are not expected to be probability distributions (i.e. marginal), but they are just terms in a joint distribution $p(Z, X)$, which need to be summed over Z, X space for normalization of (1.16). However, any exponential family pdf (Sec. 1.3.7.1) can be embedded in (1.16).

The MRF is popular in image segmentation and classification (Gould et al., 2008) thanks to its ability to handle high-dimensional spaces, efficient parameter inference and availability of hyperparameter learning methods. On the other hand it requires the graph to be fixed for a given data input, which does not leave much space to apply it to problems with variable number of parts. We can rather cast these problems as assignment of primitives to classes interpreted as semantic components.

A variant called *Conditional Random Field* (CRF) introduced by Lafferty et al. (2001)

models directly the conditional distribution $p(Z \mid X)$, where Z is a configuration (Sec. 1.3.1.3) of labels and θ_j .

1.3.3 Inference Methods

A complex probabilistic model would be useless if we had no practical method to infer (estimate) its parameters from data. In this section we will mention the standard methods which can be applied to the structured models of our interest.

The process of inference of the model parameters θ from the data X can be generally expressed with

$$\theta^* = \arg\max_{\theta} f(\theta \mid X). \tag{1.17}$$

1.3.3.1 Maximum Likelihood Estimation

The estimation in the classical case (1.12) is known as *Maximum Likelihood* (ML) estimation, which is a direct maximization of the

$$\theta^* = \arg\max_{\theta} p(X \mid \theta), \tag{1.18}$$

treated as a function of θ, which is the only function we have to specify. With the assumption of independence of observations we can write

$$p(X \mid \theta) = \prod_{x_i \in X} p(x_i \mid \theta), \tag{1.19}$$

which conveniently translates into log-likelihood as

$$\mathcal{L}(X \mid \theta) = \sum_{x_i \in X} \log p(x_i \mid \theta), \tag{1.20}$$

and the estimate θ^* maximizing (1.18).

The ML estimation is usually chosen when there is no additional information on the parameters but the data observation model; as such it simply cannot be used with structured models.

1.3.3.2 Maximum a Posteriori Estimation

When we plug in a prior in (1.13) we proceed with *Maximum A Posteriori* (MAP) estimation

$$\theta^* = \arg\max_{\theta} p(X \mid \theta)\, p(\theta), \tag{1.21}$$

where we can safely drop $p(X)$ from (1.13) because it does not depend on θ. The estimate can be found similar to ML with an extra term for the prior in (1.20). However, chances to get a closed-form estimator from MAP are generally lower and a search for appropriate prior distribution can be cumbersome, which is the major point in the criticism of Bayesian approach.

In the context of models with variable number of parts we can consider complexity k as one of the parameters, $k \in \theta$, and use (1.21) to estimate them all simultaneously with MAP.

1.3.3.3 Bayesian Estimation

Both ML and MAP return only a single estimate of values θ^* for the parameters. In contrast the *Bayesian estimation* calculates the full posterior distribution $p(\theta \mid X, \xi)$, for which the denominator $p(X \mid \xi)$ from (1.13) must be also calculated. This further restricts the prior choice such that integration (1.14) can be carried out. However $p(X \mid \xi)$ need not to be available in closed-form, we can calculate it numerically (enumerating a discrete distribution) or it can be sufficient to estimate it (using random sampling).

The benefit of obtaining the full posterior is we can further analyze the parameter space. We can calculate the variance associated with the MAP estimate or find alternative estimates when the posterior is multimodal. We can also use it for prediction.

1.3.3.4 Model Selection and Two-Level Inference

Rather than direct estimation of complexity k hinted in Sec. 1.3.3.2 we can employ Bayesian estimation and treat the problem of the unknown number of components as a *model selection* problem. Following Šára (2014) we can treat complexity k not as a parameter of a single model but instead we consider multiple models with different complexity $k = 0, 1, 2, \ldots, k_m$. This results in a hierarchical model

$$p(X, \theta, k) = p(X \mid \theta, k)\, p(\theta \mid k)\, p(k). \tag{1.22}$$

Two-level Bayesian inference (MacKay, 2003) is then used to perform model selection, the task of selecting a model from a set of candidate models, given data. The selection criterion in this case is Bayes factor (evidence ratio) generalized to multiple models. In the context of this thesis the goal of inference is two-fold:

1. Determine the most probable **complexity** k^* according to

$$k^* = \arg\max_k p(k \mid X) = \arg\max_k \int p(X, \theta, k)\, \mathrm{d}\theta, \tag{1.23}$$

   in which $p(k \mid X)$ is the posterior marginal from (1.22).

2. Given k^*, determine the most probable **parameters** θ

$$\theta^* = \arg\max_\theta p(\theta \mid X, k^*). \tag{1.24}$$

In practice the actual inference procedure usually performs both levels simultaneously. Approximate information criteria for model selection (Bayesian BIC, Akaike AIC, etc.) are overly simplifying for complex models of our interest.

1.3.4 Random Sampling Methods

One of the possible options for implementing the general approaches mentioned in the previous section is to use random sampling. Computing marginals of complex probabilistic functions typically requires a sampling method. Naive sampling methods would result in an algorithm that is too slow in practical-size problems. In this section we will review relevant sampling methods and discuss their applicability to structured models with a variable number of parts.

1.3.4.1 RANSAC

Although not probabilistic, *Random Sample Consesus* (RANSAC) by (Fischler and Bolles, 1981) is presumably the most popular algorithm for stochastic inference of parameters in computer vision. Its key idea is to use sampling of parameters θ from the empirical distribution of data X, which makes it efficient; we can make use of this mechanism also in the context of probabilistic sampling.

Even when the optimized 'energy' function can be arbitrary, we cannot directly apply RANSAC to problems with variable number of components. Greedy sequential estimation of individual components turns out to be suboptimal as discussed in (Zuliani et al., 2005), where it has been extended for joint sampling of all components but still the complexity k is considered given. This has been overcome by method of Wang et al. (2012) but the determination of k remains empirical.

1.3.4.2 MCMC

Markov Chain Monte Carlo (MCMC) is a class of advanced methods for sampling from arbitrary probability distributions, which is particularly useful for Bayesian inference (Gilks and Roberts, 1996). The major advantage over independent sampling (like in RANSAC) lies in the Markov process where a new sample is conditioned on the previous one (but not on further preceding states). The dominant sub-classes in MCMC are random walk methods, but some variants implement deterministic 'shortcuts' to improve convergence and efficiency (Duane et al., 1987; Roberts and Tweedie, 1996).

Gibbs sampling is a popular MCMC method (Geman and Geman, 1984) because it is formally simple, but requires marginal distributions for all parameters, which usually does not allow to apply it for complex structured models.

The universal method in the MCMC family is *Metropolis-Hastings* (MH) algorithm, which allows to obtain samples from an arbitrary *target distribution* $\pi(\theta)$ even when we cannot directly sample from it. The basic idea is that instead of direct sampling we take samples from auxiliary *proposal distribution* $q(\theta' \mid \theta)$ and filter them by an probabilistic acceptance algorithm. A sample is accepted randomly with *acceptance probability* $a(\theta \mid \theta')$. It is derived from the *detailed balance* condition that guarantees reversibility of MC for its stationary distribution π:

$$\pi(\theta)\, T(\theta' \mid \theta) = \pi(\theta')\, T(\theta \mid \theta'), \tag{1.25}$$

where we express the transition $T(\theta \mid \theta')$ as the proposal q and acceptance-rejection a

$$T(\theta' \mid \theta) = q(\theta' \mid \theta)\, a(\theta' \mid \theta). \tag{1.26}$$

Together we get the acceptance equation

$$\frac{a(\theta' \mid \theta)}{a(\theta \mid \theta')} = \frac{\pi(\theta')\, q(\theta \mid \theta')}{\pi(\theta)\, q(\theta' \mid \theta)}, \tag{1.27}$$

which a particular acceptance function must fulfill. A common choice (Hastings, 1970) is

$$a(\theta' \mid \theta) = 1 \wedge \frac{\pi(\theta')\, p(\theta \mid \theta')}{\pi(\theta)\, p(\theta' \mid \theta)}, \tag{1.28}$$

where $a \wedge b = \min(a, b)$.

While in theory MH can accommodate structured models with any level of complexity, there is a price associated with this generality. In practice we are limited by our ability to design proposal distributions sufficiently close to the target distribution. If unsuccessful, the majority of proposed samples would be rejected (low *acceptance rate*), rendering the sampler inefficient and slowly converging. Associated performance indicator is the *mixing rate* describing the sampling process agility and efficiency in exploring configuration and parameter spaces; it can be loosely characterized as average correlation of consecutive states in MC.

With rapidly increasing number of random variables in computer vision models this problem is aggravated by the *curse of dimensionality* (Bellman and Bellman, 1961) in optimization. Most notably high-dimensional spaces are sparse and random walk must travel further to explore them because finding a tight bounding distribution is usually difficult. Also combinatoric aspect gets in the way as the number of possible explanations of the data grows exponentially.

1.3.4.3 Reversible Jump

The standard MH algorithm needs to be extended when the dimension of the parameter space $\theta \in \Theta$ is uknown, which is the case of the models with variable number of parts k. The mechanism accounting for the dimension changes in accordance with the measure theory is known as *Reversible Jump* (Green, 1995).

In the standard implementation of RJ a proposal only changes the complexity k by a fixed step (e.g. ± 1, add/remove a component). A typical implementation involves also a pair of component split/merge proposals (Jain and Neal, 2000). Recently Pandolfi et al. (2014) has proposed a more efficient sequential multipoint proposal variant, where several sequential complexity proposals are jointly considered as candidates from which one is selected.

The simplest method to obtain $p(X \mid k)$ in Bayesian selection of complexity (Sec. 1.3.3.4) is a histogram of posterior samples obtained from a RJ-MCMC sampler. For each k, the sampler

also remembers the best configuration found for the particular complexity in terms of (1.22). With detailed balance and reversibility conditions fulfilled the resulting configuration is asymptotically globally optimal.

1.3.4.4 Adaptive Methods

As mentioned above in Sec. 1.3.4.2, the choice of proposal distributions is critical for practical efficiency of MCMC. The proposals are usually controlled by a set of inference parameters such as the variance of the proposed model parameter change (length of a step in random walk). For given input data it is possible to find an optimal value of such parameter w.r.t. convergence, however for a different input the value will be no longer optimal.

A solution to this problem is on-line adaptation of selected proposal distribution parameters (Rosenthal, 2011), where basically the step length in a random walk is adjusted to achieve target acceptance[5]. Its goal is to achieve efficient mixing (Atchadé, 2006; Shaby and Wells, 2010a). The introduction of adaptation caused a small revolution in MCMC methods and brought them closer to practical sampling and inference methods.

1.3.4.5 Population Methods

With parallel computation resources becoming available in the recent years, several approaches have been proposed to enhance MCMC methods both quantitatively (speed up by parallel sampling, (Neiswanger et al., 2014)) and more interestingly qualitatively (efficiency, convergence (Laskey and Myers, 2003)) by running a *population* of MH samplers simultaneously. The statistical information from a population of samplers is used to inform the proposal distributions for individual samplers in the population. Experimental results (Laskey and Myers, 2003) show that the population learns more efficiently than the individual samplers with no information exchange.

Our recent experience shows that population methods can solve harder inference problems where there are many maximizers of (1.24) that are distant in the configuration space, or in other words the alternative solutions θ_1^*, θ_2^*, are close w.r.t. target distribution $|\pi(\theta_1^*) - \pi(\theta_2^*)| \to 0$, but distant w.r.t. proposal distribution $q(\theta_2^* \mid \theta_1^*) \to 0$.

1.3.4.6 Hybrid Methods

Several approaches have been proposed that complement the random walk in MCMC with deterministic steps (Neal, 2011; Duane et al., 1987; Roberts and Tweedie, 1996), which has led to call them generally *hybrid methods*.

Stochastic Approximation Expectation-Maximization (SAEM) algorithm (Delyon et al., 1999) is a modern variant of Monte-Carlo EM algorithm. From the perspective of MCMC, this technique provides a way how all the produced samples from $\pi(\theta)$ can be stochastically averaged (expectation) to (re-)estimate the parameters θ (maximization).

[5] In Gaussian setting short steps (small changes) have higher acceptance than long steps.

The histogramming of posterior complexities for obtaining $p(X \mid k)$ mentioned in Sec. 1.3.4.3 can be replaced by a more efficient marginal estimation, e.g. using the thermodynamic integration (Calderhead and Girolami, 2009) and the best-sample wait can be replaced by the EM algorithm for parameter estimation.

A consistent framework for such process has been implemented in a hybrid inference method called *LiSAEM* (Šára, 2014), which blends several existing concepts together and not only efficiently estimates the number of components but also provides estimates for other parameters (component parameters, variance, outlier probability). To achieve computationally efficient algorithm, its inference model is constructed so that the amount of random sampling is kept to a minimum. This is achieved by combining hybrid sampling ideas with PEARL-like optimization based on a set of random labels (Isack and Boykov, 2012) and a Riemannian version (Bui-Thanh and Ghattas, 2012) of Metropolis-Adjusted Langevin algorithm (Roberts and Tweedie, 1996) as an efficient proposal mechanism for parameters θ. The engine uses many additional ideas from the literature some of which are mentioned above.

1.3.5 Inference and Learning for Graphical Models

From a vast number of methods providing inference and learning in graphical models (Koller and Friedman, 2009), we pick up a selection related to MRFs.

In general case exact inference in MRFs is not possible, but approximation techniques are available. The standard approximate algorithm is *Loopy Belief Propagation* (LBP), which is based on message passing over graph edges when nodes iteratively 'vote' for their neighbors values given their own value. Its convergence properties are however degrading with increasing complexity of the graph structure; this also holds for stochastic approximation methods (MCMC). Variants of LBP covering the original graph with subgraphs and combining the particular solutions on the subgraphs have shown better performance (Kolmogorov, 2006).

The specific cases when exact inference is possible either limit the graph topology or the choice of potential functions. If in the first case the graph is a chain or tree, message passing (LBP) converges to exact solution (in analogy to the forward-backward and Viterbi algorithms for linear chains). In the latter case if a *submodularity* condition on the potentials holds then max-flow algorithms (alpha-expansion, alpha-beta swapping) return exact solutions (Kohli et al., 2009).

In the case of CRF learning of weights θ (hyperparameters in this case) is possible using ML or *Maximum Pseudo Likelihood* (MPL) approximation to a joint distribution assuming conditional independence (Lafferty et al., 2001).

The limiting factor for application of MRFs to our problems of interest is that parameter set including complexity must be fixed prior to inference. We therefore do not discuss methods from this area in detail.

1.3.6 Notation Remarks

Vectors will be generally typeset in bold face, e.g. $\mathbf{x} = (x_1, x_2, \dots)$ with elements x_i, and matrices in bold capitals, e.g. $\mathbf{X}$. The bold face does not apply to Greek alphabet.

1.3.7 Probability Distributions

We list abbreviations used throughout this thesis, along with the common parametrization for reference. Detailed explanation can be found in the most of textbooks on probability and statistics.

Continuous:

pdf	probability density function
$\mathcal{N}$	Normal (Gaussian), univariate, $\mathcal{N}(x;\ \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, $\sigma > 0$,
$\mathcal{N}_c$	Circular Normal (von Mises), $\mathcal{N}_c(x;\ \mu, \kappa) = \frac{e^{\kappa\cos(x-\mu)}}{2\pi I_0(\kappa)}$, $\kappa > 0$,
Exp	Exponential, $\mathrm{Exp}(x;\ \lambda) = \lambda e^{-\lambda x}$, $\lambda > 0$,
$\mathcal{IG}$	Inverse Gamma, $\mathcal{IG}(x;\ \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left(-\frac{\beta}{x}\right)$, $\alpha, \beta > 0$,
Be	Beta, $\mathrm{Be}(x;\ \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}$, $\alpha, \beta > 0$,
Dir	Dirichlet, symmetric, $\mathrm{Dir}(x_1, \dots, x_{k-1}; \alpha) = \frac{\Gamma(\alpha k)}{\Gamma(\alpha)^k} \prod_{i=1}^k x_i^{\alpha-1}$, $\alpha > 0$,
$\mathcal{U}$	Uniform, $\mathcal{U}(\Omega) = \frac{1}{\lvert\Omega\rvert}$

Discrete:

pmf	probability mass function
$\mathcal{B}$	Binomial, $\mathcal{B}(k;\ n, p) = \binom{n}{k} p^k (1-p)^{n-k}$, $p \in [0, 1]$, $n \in \mathbb{N}$,
Ber	Bernoulli, $\mathrm{Ber}(k;\ p) = p^k(1-p)^{1-k}$ for $k \in \{0, 1\}$.
Pois	Poisson, $\mathrm{Pois}(k;\ \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$, $\lambda > 0$,
$\mathcal{U}$	Uniform, $\mathcal{U}(n) = \frac{1}{n}$, $n \in \mathbb{N}$.

These abbreviations will be used to refer to the *probability density (mass) function* of a given type, i.e. the fact that variable x has normal probability density function with a given mean μ and variance σ^2 will be for convenience expressed as

$$p(x) = \mathcal{N}(x;\ \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

where symbols after ';' are fixed distribution hyperparameters. This is slightly different to the common notation

$$x \sim \mathcal{N}(\mu, \sigma^2),$$

where the symbol $\mathcal{N}$ represents the distribution itself, we will use the both expressions. The symbol $\sim$ is used to emphasize that x is sampled (generated) from the given distribution.

The symbol p will be used more loosely for both probability density and mass functions and likelihood, the concrete meaning is determined by the context and by the discrete or continuous domain of its arguments. We will use the simplified notation $p(x)$ instead of $p_x(x)$, i.e. the identity of the function p is determined by its arguments. For example $p(x)$ and $p(y)$ are two different functions, precisely $p_x(x)$ and $p_y(y)$. In the case we need to specify two different functions of the same arguments this will be denoted explicitly, i.e. $p_1(x)$ and $p_2(x)$. In the case of arguments with variable indices such as $p(x_i)$, $i = 1, 2, 3, \ldots$, we specify the same function for all x_i, unless explicitly specified otherwise.

1.3.7.1 Exponential Family Distributions

An *exponential family distribution* can be written as

$$p(x \mid \theta) = h(x)g(\theta) \exp\left[\sum_{w=1}^{W} \eta_w(\theta)\, T_w(x)\right], \tag{1.29}$$

where η_i are *natural parameters* and T_i are *sufficient statistics* of the exponential-class model. All distributions given above belong to the exponential family, in some cases certain parameters must be fixed (those that change the support of the distribution).

1.4 Thesis Goals

Based on the analysis of state of the art we set the following goals:

- Try to use principles of weak grouping for symmetric object detection.
- Design a probabilistic model for image symmetries involving multiple elements.
- Propose an inference mechanism for detecting such symmetries that does not oversegment.
- Provide a good estimator of complexity for symmetries of unknown order.
- Attempt to learn important structural relations.

Each of the following chapters addresses some of these goals. The most complex model is presented in Chapter 4. The main results are summarized in Chapter 5.

Chapter 2

Weak Structure Model

"From now on we can compare our data with the model we actually want to use rather than with a model which has some mathematical convenient form. This is surely a revolution."

Peter Clifford (1993)[1]

2.1 Introduction

For our initial approach to symmetry detection we have chosen a level of structure complexity that allows us to solve some real-world problem while we can oversee its individual parts and analyze their impact on the overall performance. This has brought us to translation symmetry detection in the world of facades with a large pool of facade elements in different architectural styles. It has become our playground for recognition of structured images.

2.2 Overview

We will present a method for detection of windows in facade images. Given an ability to obtain local low-level data evidence on individual components (windows), we determine their most probable number, locations, size and neighborhood relation. The embedded structure is weakly modeled by pair-wise attribute constraints, which allow structure and attributes to mutually support each other. We will use a general framework of Reversible Jump MCMC (Sec. 1.3.4.3) to perform MAP estimation of component count and parameters (Sec. 1.3.3.2), which is the simplest probabilistic approach applicable to structured models with variable number of components.

We initially designed a probabilistic model based on a grid with rows and columns (Tyleček and Šára, 2011b) which also allows exceptions both in locations and structure, see Fig. 2.1.

[1] The Royal Statistical Society meeting on the Gibbs sampler and other statistical Markov Chain Monte Carlo methods, Journal of the Royal Statistical Society, Series B, 55(1), p. 53.

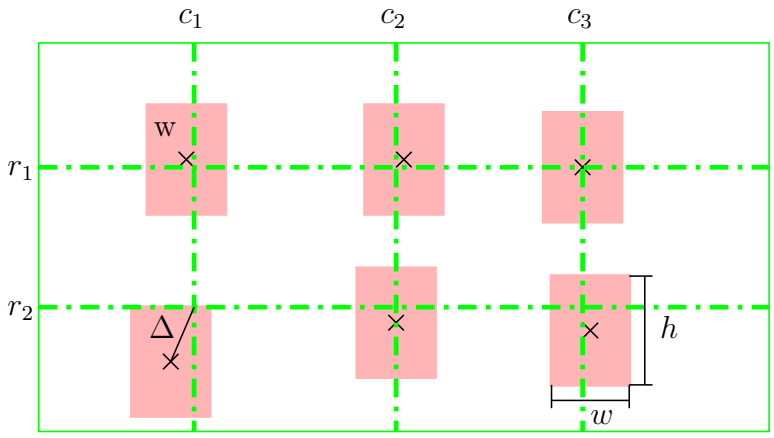


Figure 2.1: Grid model with rows r_i and columns c_i allows local deviations Δ from the grid for limited flexibility.

With a classifier trained specifically for windows it performed well, but its flexibility was limited.

To allow also loosely regular configurations like those in Fig. 2.14, we have proposed another method, where the structure is modeled softly by local pair-wise constraints (Tyleček and Šára, 2011a, 2012). The two variants of this model with different structural prior will be presented in this chapter. In the domain of window recognition in facade images we will demonstrate that the result is an efficient algorithm achieving performance of other strongly informed methods for regular structures. The majority of the algorithms for single-view facade interpretation mentioned in Sec. 1.2.5 work with hard constraints on grid configurations of windows and employ strong domain-specific heuristics, which may result in overfitting.

Our work can be also seen as an object-specific extension of a general lattice discovery method by Park et al. (2009), but in our case the layout is not constrained to a lattice, which results in a more complex model.

2.3 Problem Description

We consider the problem of recognizing specific objects (facade windows), which correspond to components. In this case the primitive elements are image pixels (Sec. 1.3.1). We assume the input image is orthographically rectified, as in Fig. 2.8. This was achieved by an automatic rectification method using vanishing point detection similar to (Förstner, 2010).

Our model parameters θ consist of complexity k (the number of components), component parameters $\bar{\theta}$ (window locations and size) and configuration parameters $\dot{\theta}$ (neighborhood of components). The recognition task can then be formulated as follows: Given image data I, we search for model parameters $\theta = (k, \bar{\theta}, \dot{\theta})$ by finding the mode of the joint distribution

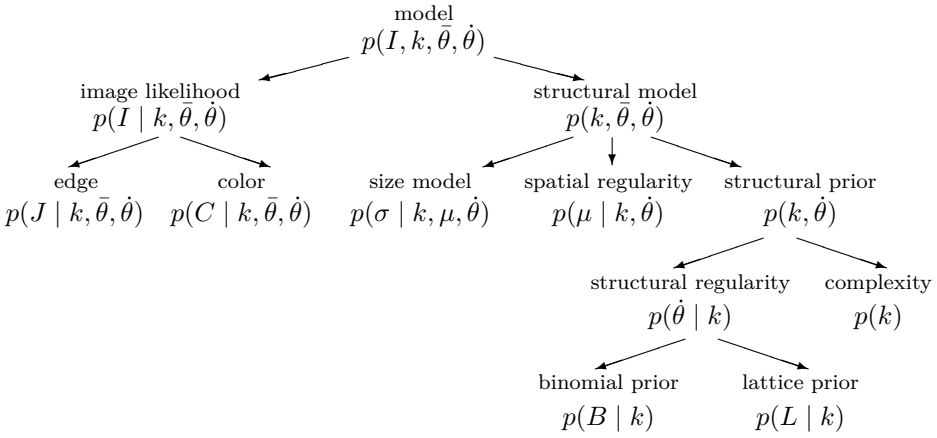


Figure 2.2: Hierarchy in the probability model. Term in this diagram is a product of its two child terms. The structural prior is the discrete part of the model which conditions the remaining continuous part.

$p(I, \theta)$ with

$$\theta^* = \arg\max_{\theta} p(I \mid \theta)\, p(\theta), \tag{2.1}$$

which is computed with Bayes theorem from data model $p(I \mid \theta)$ and structural model prior $p(\theta)$. We will decompose our probability model hierarchically as shown in Fig. 2.2 and propose pdfs specific for the task of window detection in facade images. Then we will apply stochastic RJMCMC framework (Sec. 1.3.4.3) based on random walk to find the optimal value θ^* by effectively sampling from the space of possible combinations of parameters θ. More details on its implementation will be given in the following sections.

We will now describe individual terms in our model basically from right to left as they appear in Fig. 2.2, starting with the independent variables. The terms will be summarized at the end of this section in Tab. 2.1.

2.4 Probability Model

We design a probabilistic structural model $p(k, \bar{\theta}, \dot{\theta})$ in which $(k, \bar{\theta}, \dot{\theta})$ is a configuration. The model captures rules for appearance of a set of similar components in an image with a semi-regular spatial distribution. Rather than explicitly imposing a lattice or a similar global layout, the model is based on local pair-wise component neighborhood and parameter constraints. We are given a set of k components with parameters $\bar{\theta} = (\mu, \sigma)$. The location parameters in vector μ are defined in the unit image plane with

$$\mu = (\mu_1, \ldots, \mu_k),\ \mu_i \in (0, 1)^2 \tag{2.2}$$

and similarly the size and shape is described with vector

$$\sigma = (\sigma_1, \ldots, \sigma_k),\ \sigma_i \in (0, 1)^2. \tag{2.3}$$

Our neighborhood representation $\dot{\theta}$ is independent on the locations μ and it is based on a complete graph N, where nodes correspond to components and edges to neighborhood relationship between them. Our goal is to define neighbors as components that are in proximity of each other and similar to each other in size and shape, i.e. they share some parameter values.

The neighborhood is encoded with pairwise labels L on edges in N as

$$L = (l_{uv} \in \{0, 1\}\,;\, (u, v) \in \{1, ..., k_m\}^2,\, u < v) \tag{2.4}$$

that are recovered as a part of the solution of (2.1). The mutual neighborhood of two components is indicated by $l_{uv} = 1$ (active edge), otherwise $l_{uv} = 0$ when the neighborhood is suppressed (inactive edge). In other words $l_{uv} = 1$ means the components u and v are *neighbors*. In the following text the (u, v) will denote edges in N, i.e. component pairs from (2.4).

The prior $p(\theta)$ in our model will be specified up to a normalization term, which is difficult to obtain in closed form as a function of k due to the complex dependencies between component parameters. Instead we will fix the number of variables in the model by including terms for all possible k_m components. The set of k_m components is split in k active components and $\bar{k} = k_m - k$ inactive components. All edges from an inactive component are also inactive ($l_{uv} = 0$). The terms in $p(\theta)$ for inactive components are uniform and there are no component parameters $\bar{\theta}$ specified for them. The normalization term is then a function of fixed k_m and a constant w.r.t. the maximization in (2.1) which allows to carry out MAP inference (Sec. 2.6).

2.4.1 Structural Prior

This prior describes a class of 2D graphs that are similar to a lattice (grid) graph[2], but its drawing need not be a regular tiling. The goal is to allow higher level of flexibility, which is required in practice for structures like in Fig. 2.14. The model consists of a structural regularity term $p(\dot{\theta} \mid k)$ and complexity term $p(k)$. The configuration parameters $\dot{\theta}$ represent global model structure, concretely the component neighborhood L.

2.4.1.1 Structural Complexity

The prior on the number of components is modeled by a binomial distribution

$$p(k;\, p_c, k_m) = \mathcal{B}(k;\, p_c, k_m) = \binom{k_m}{k} p_c^k (1 - p_c)^{k_m - k}, \tag{2.5}$$

where $k_m \in \mathbb{N}$ is the maximum number of components in the model and $p_c \in (0, 1)$ models their expected count relative to k_m.

[2]We will understand 'grid' as a regular square plane graph, which is a special case of a more general 'lattice'.

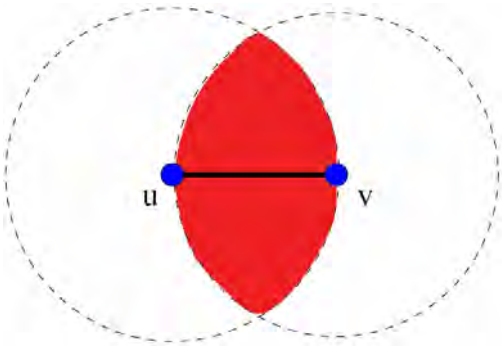


Figure 2.3: Relative Neighborhood Graph condition. Two planar points u and v are connected by an edge whenever there does not exist a third point r that is closer to both u and v than they are to each other (in Euclidean metric). The condition defines a 'forbidden zone' (red).

2.4.1.2 Structural Regularity

We want to regularize the neighborhood L by introducing grid-like constraints. We have evaluated two options.

Since we are dealing with image components parameterized by their locations μ in the image plane, we can limit the edge labels $l_{uv} = 1$ in L to induce only planar subgraphs of N. The *Relative Neighborhood Graph* (RNG[3]) is a natural choice (Tyleček and Šára, 2011a) and it is defined by the condition demonstrated in Fig. 2.3. This choice defines a function $\bar{\theta} \mapsto L$, which forces $l_{uv} = 0$ where the actual component locations μ violate the RNG constraint.

However in certain situations the RNG is too restrictive, preventing the neighborhood where it would be desirable. In Tyleček and Šára (2012) we have presented the second option, which is a more general *Softly Bipartite Graph* (SBG) (see Fig. 2.6). A bipartite graph is two-colorable, meaning that we can assign a binary label $b_i \in \{0, 1\}$ to every node such that every edge connects nodes with different labels. This property imposes strong constraints on the structure of graph cycles. However, in our case we relax this condition by allowing edges connecting equally colored nodes but assigning them a low probability p_b (softness). For the SBG we extend $\dot{\theta}$ with a set of hidden variables $B = (b_i;\, i = 1, \ldots, k_m)$ and model them with a Gibbs distribution w.r.t. N in

$$p(B \mid L, k) \propto \prod_{u,v} p(b_u, b_v \mid l_{uv}), \tag{2.6}$$

where the joint distribution for a pair of binary variables b_u, b_v is given by

$$p(b_u, b_v \mid l_{uv};\, p_b) = \begin{cases} \frac{1}{2} p_b, & l_{uv} = 1,\ b_u = b_v, \\ \frac{1}{2}(1 - p_b), & l_{uv} = 1,\ b_u \neq b_v, \\ \frac{1}{4}, & l_{uv} = 0 \,(\text{inactive edge}). \end{cases} \tag{2.7}$$

[3]RNG can be computed from *Delaunay Triangulation* efficiently in $\mathcal{O}(n)$ time.

Note that the regular grid graph and its node-induced subgraphs are bipartite, but this no longer holds when some nodes are removed from the grid and the associated edges are joined in both directions. The softness of the SBG however allows for irregular lattices where odd-length cycles are present, such as in Fig. 2.14b.

The structural regularity term takes in the SBG case the form of

$$p(\dot{\theta} \mid k) = p(B \mid L, k)\, p(L \mid k). \tag{2.8}$$

The $p(B \mid L, k)$ term was omitted for the RNG case.

The term $p(L \mid k)$ common to both the structure priors is described next. Let $d_c(k)$ be the number of edges in the complete graph N (the number of variables l_{uv})

$$d_c(k) = \binom{k}{2}. \tag{2.9}$$

The preferred number of neighbors (active edges in the graph N) is the number of edges in a regular square grid with k nodes

$$d_g(k) = 2(k - \left\lfloor \sqrt{k} \right\rfloor), \tag{2.10}$$

where $\lfloor x \rfloor$ denotes the greatest integer number lower or equal to x. Let

$$q(k) = \frac{d_g(k)}{d_c(k)} = \frac{4}{\sqrt{k}(\sqrt{k}+1)}. \tag{2.11}$$

Then the actual number of neighbors

$$\Sigma(L) = \sum_{uv} l_{uv} \tag{2.12}$$

is modeled with

$$p(L \mid k) = q(k)^{\Sigma(L)} \left(1 - q(k)\right)^{d_c(k) - \Sigma(L)}, \tag{2.13}$$

which corresponds to a binomial distribution compensated for the number of graphs with the same observed number of active edges.

2.4.2 Spatial Regularity

This part of the model describes rules for the relative location of neighboring components similar to translation symmetry in a lattice or continuity and proximity principles in Gestalt theory.

We parameterize the spatial relation of components in relative polar coordinates (see Fig. 2.4) by

$$p(\mu \mid k, \dot{\theta}) \propto p(\rho, \phi \mid k, \dot{\theta})\, p(\mu_1), \tag{2.14}$$

where $\rho = (\varrho_{uv};\ (u,v) \in N)$ and $\phi = (\varphi_{uv} \in [0, 2\pi));\ (u,v) \in N)$ such that for a given pair of components (u,v) the distance is calculated with

$$\varrho_{uv} = \varrho(\mu_u - \mu_v) = ||\mu_u - \mu_v|| = \varrho_{vu} \tag{2.15}$$

and the orientation (angle) of the location difference vector $\mu_u - \mu_v$ with

$$\varphi_{uv} = \varphi(\mu_u - \mu_v) = \varphi_{vu} + \pi, \tag{2.16}$$

i.e. the opposite direction ($u \to v$ or $v \to u$) is associated with the opposite angle. Note the original Cartesian coordinates μ can be recovered from relative polar coordinates ρ, ϕ up to a global offset, which can be specified e.g. by the absolute location of the first component μ_1. Its pdf $p(\mu_1) = 1$ is uniform.

In order to establish a distribution on ρ, ϕ let us introduce a line graph D dual to N, where nodes in D correspond to neighbors (active edges) in D and there is an edge between two nodes in D iff the two edges in N share a common node. There is a maximal clique (complete subgraph) in D associated[4] with each node u from N. The corresponding local neighborhood of u will be denoted with $N(u)$ and further specified as a sorted circular list of its n_u neighbors v_i ordered by angles relative to u:

$$N(u) = \left\{ v_i;\ i = 1, \ldots, n_u,\ l_{uv_i} = 1,\ \varphi_{uv_i} \le \varphi_{uv_{i+1}} \right\}, \tag{2.17}$$

where no component is preferred as starting. Then we can implement the symmetry principles in a Gibbs distribution w.r.t. $D(N)$ in

$$p(\rho, \phi \mid k, \dot{\theta}) \propto \underbrace{\prod_{u,v} p(\varrho_{uv})\ p(\varphi_{uv})}_{\text{priors}}\ \underbrace{\prod_{u,v,w} p_s(\varrho_{uv}, \varrho_{uw})}_{\text{spacing}}\ \underbrace{\prod_{u} p_a\left(\varphi_{uv_i};\ v_i \in N(u)\right)}_{\text{alignment}}, \tag{2.18}$$

where $p(\varrho_{uv})$, $p(\varphi_{uv})$ are prior terms unary w.r.t, We assume local Markov property, i.e. component parameters $\bar{\theta}_j$ are conditionally independent of all other, given its neighbors. The exponential form of the distribution following (1.16) is straightforward when the factors in (2.18) described below have the form of an exponential family distribution, which is our case. The continuous unit domain of μ guarantees the normalization of (2.18) can be carried out and the resulting partition function is fixed by k_m. As such the normalization of (2.18) is not required to perform the maximization in (2.1).

[4]The inverse however does not hold because there can be cliques of order three (triangles) in D, which do not correspond to to any node in N, i.e. in a triangular lattice there are also such cliques corresponding to faces (triangles) in N. We do need to identify such cases because there is a uniform model for all order 3 cliques (Sec. 2.4.2.3).

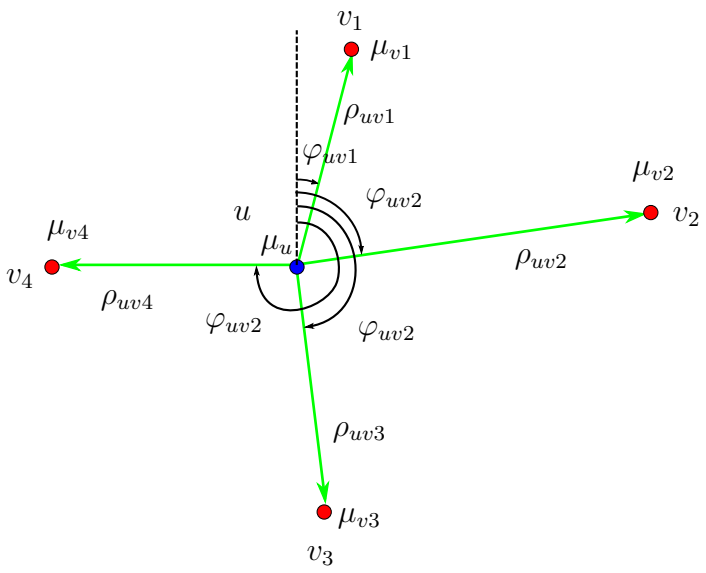


Figure 2.4: A component u with four neighbors $v_i \in N(u)$ and their relative polar coordinates (angle φ and distance ρ) parametrize spatial regularity for locations μ.

2.4.2.1 Spatial Priors

The prior assumption on the position of components is that neighboring components should be horizontally or vertically aligned parallel to axes of the rectified input image. This translates in a prior for orientation φ_{uv} preferring certain absolute angles: $-\frac{\pi}{2}, 0, \frac{\pi}{2}, \pi$. The prior orientation is modeled with circular normal (von Mises) distribution in

$$p(\varphi_{uv};\, \kappa_c) = \frac{1}{4}\mathcal{N}_c(4\varphi_{uv};\, 0, \kappa_c) = \frac{1}{4}\frac{e^{\kappa_c \cos 4\varphi_{uv}}}{2\pi I_0(\kappa_c)}, \tag{2.19}$$

where κ_c is the concentration parameter and I_0 is the modified Bessel function of order 0. Note that the prior is symmetric, $p(\varphi_{uv};\, \kappa_c) = p(\varphi_{vu};\, \kappa_c)$.

The prior on relative distances ϱ_{uv} is a beta distribution with pdf

$$p(\varrho_{uv};\, \alpha_d, \beta_d) = \mathrm{Be}(\varrho_{uv};\, \alpha_d, \beta_d) = \frac{\varrho_{uv}^{\alpha_d - 1}(1 - \varrho_{uv})^{\beta_d - 1}}{\mathrm{B}(\alpha_d, \beta_d)}. \tag{2.20}$$

2.4.2.2 Spacing

The second assumption is that the distance ρ_{uv} between components in a neighborhood should most probably be equal. We model the assumption pairwise in

$$p_s(\varrho_{uv}, \varrho_{uw};\, \beta_r) = \frac{1}{\varrho_{uv} + \varrho_{uw}}\, \mathrm{Be}(\frac{\varrho_{uv}}{\varrho_{uv} + \varrho_{uw}};\, \beta_r, \beta_r), \tag{2.21}$$

where $u \neq v \neq w$ are indices of two edges in N sharing a node u and β_r is the concentration parameter. We choose beta distribution in this case because it describes our knowledge on the observed data sufficiently well and matches with the rest of the model. Similar practical

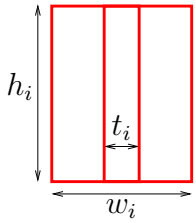


Figure 2.5: Window shape template. It is parametrized by its width $w_i \in (0,1)$, height $h_i \in (0,1)$, both relative to image height I_h, and the width of the central column $t_i \in (0,1)$ relative to the window width.

reasons motivate our choices of terminal (beta) distributions also elsewhere in this model.

2.4.2.3 Alignment

The assumption for orientations φ_{uv} is that the neighbors of a given component u should be evenly distributed around it in terms of their relative angles as in Fig. 2.4. We model the alignment in a joint function of these orientations

$$p_a\left(\varphi_{uv_i};\, v_i \in N(u);\, \alpha_\varphi\right) = \frac{1}{n_u} \times \begin{cases} \mathrm{Dir}\left(\left(\frac{\varphi_{uv_2}-\varphi_{uv_1}}{2\pi}, \ldots, \frac{\varphi_{uv_n}-\varphi_{uv_{n-1}}}{2\pi}\right);\, \alpha_\varphi\right), & n_u \geq 4, \\ \left(\frac{1}{\pi}\right)^{n_u}, & n_u \in \{2,3\}, \end{cases} \tag{2.22}$$

and the factor $\frac{1}{n_u}$ is due to the free starting component. The Dirichlet pdf assigns the highest probability to configurations in which the differences between the neighbors' angles are equal to each other, i.e. π for two neighbors, $\frac{2\pi}{3}$ for three, $\frac{\pi}{2}$ for four, etc. The cases $n_u = 3$ corresponds to the corners and sides in the case of grid structure and we fall back to the uniform distribution $\mathcal{U}(\pi) = \frac{1}{\pi}$ for every φ_{uv_i} in the clique. The case $n_u = 1$ is covered by the prior (2.19).

2.4.3 Size Parameters

Aside from the locations μ, the appearance of components is described with size and shape parameters σ. Our components are represented by a rectangular shape template with its borders parallel to image borders. The size and shape parameters of a component is a vector

$$\sigma_i = (w_i, h_i, t_i) \in (0,1)^3, \tag{2.23}$$

its parts are described in Fig. 2.5, the central column position parameter t_i is relative to the width h_i and specific to 'window' components.

The shape model is a Gibbs distribution w.r.t. N with unary and binary factors

$$p(\sigma \mid k, N, \mu) \propto p_0(\sigma \mid \mu) \underbrace{\prod_{i=1}^{k_m} p_1(\sigma_i \mid k)}_{\text{prior}} \underbrace{\prod_{u,v} p_2(\sigma_u, \sigma_v \mid l_{uv})}_{\text{similarity}}, \tag{2.24}$$

where we additionally specify a global non-overlapping prior by setting $p_0(\sigma \mid \mu) = 0$ when any two shape rectangles overlap each other (with the simplifying assumption of independence on p_1 and p_2). The exponential form of the distribution following (1.16) is straightforward when the factors p_1 and p_2 are based on exponential family distributions, just as in the following text.

2.4.3.1 Size Prior

The unary factors are size and shape parameter priors

$$p_1(\sigma_i \mid k) = \begin{cases} p(t_i \mid h_i)\, p(w_i \mid h_i)\, p(h_i), & i \leq k, \\ 1, & k < i \leq k_m, \end{cases} \tag{2.25}$$

where the term for inactive components is uniform $\mathcal{U}(1)$ for all size parameters. We choose to model the central column width with beta distribution

$$p(t_i \mid h_i;\, \alpha_t, \beta_t) = \mathrm{Be}(t_i;\, \alpha_t, \beta_t). \tag{2.26}$$

The typical aspect ratio (rectangular window) is modeled with Beta distribution

$$p(w_i \mid h_i;\, \alpha_a, \beta_a) = \frac{1}{w_i + h_i} \mathrm{Be}\left(\frac{w_i}{w_i + h_i};\, \alpha_a, \beta_a\right), \tag{2.27}$$

where the factor $\frac{1}{w_i+h_i}$ is due to transformation $w_i \mapsto \frac{w_i}{w_i+h_i}$ for the beta pdf. The height prior is chosen as

$$p(h_i) = \mathrm{Be}(h_i;\, \alpha_h, \beta_h). \tag{2.28}$$

2.4.3.2 Size Similarity

Our size constraints reflect the similarity principle, i.e. neighboring components should most probably have the same size and shape. This can be described with binary factors in

$$p_2(\sigma_u, \sigma_v \mid l_{uv}) = \begin{cases} p(w_u, w_v)\, p(h_u, h_v)\, p(t_u, t_v), & \text{if } l_{uv} = 1, \\ 1, & \text{if } l_{uv} = 0, \end{cases} \tag{2.29}$$

where

$$p(w_u, w_v) = \frac{1}{w_u + w_v} \mathrm{Be}\left(\frac{w_u}{w_u + w_v};\, \alpha_s\right) \tag{2.30}$$

is a Beta distribution with its mode at $w_u = w_v$, in the case of $l_{uv} = 0$ the distribution is uniform. Analogically we define the pdfs for h and t similarity.

Term	*Eq.*	*Param.*	*Description*	*pdf*	*Hyperparameters*
$p(k)$	(2.5)	k	complexity prior	$\mathcal{B}$	$k_m = 100,\ p_c = 0.5$
$p(b_u, b_v \mid l_{uv})$	(2.6)	$\dot{\theta}$	bipartite coloring	Ber	$p_b = 0.01$
$p(L \mid k)$	(2.13)	$\dot{\theta}$	edge count	$\mathcal{B}$	-
$p(\varphi_{uv_i};\ v_i \in N(u))$	2.22	$\bar{\theta}$	location alignment	Dir	$\kappa_a = 10$
$p(\varphi_{uv} \mid l_{uv})$	(2.19)	$\bar{\theta}$	alignment prior	$\mathcal{N}_c$	$\kappa_o = 10$
$p(\varrho_{uv}, \varrho_{uw} \mid l_{uv})$	(2.21)	$\bar{\theta}$	location spacing	Be	$\alpha_d = 5, \beta_d = 20$
$p(\varrho_{uv} \mid l_{uv})$	(2.20)	$\bar{\theta}$	spacing prior	Be	$\beta_r = 20$
$p(w_i \mid h_i)$	(2.27)	$\bar{\theta}$	aspect prior	Be	$\alpha_a = 20, \beta_a = 10$
$p(h_i)$	(2.28)	$\bar{\theta}$	height prior	Be	$\alpha_h = 2, \beta_h = 40$
$p(t_i \mid w_i)$	(2.26)	$\bar{\theta}$	column prior	Be	$\alpha_t = 2, \beta_t = 40$
$p(w_u, w_v)$	(2.30)	$\bar{\theta}$	size similarity	Be	$\alpha_s = 3$

Table 2.1: Structural model parameters and their distributions.

2.4.4 Hyperparameters

In this chapter we avoid the complete specification of hyperpriors and associated hierarchical Bayesian inference. For simplicity we restrict ourselves to empirical estimation of hyperparameters. The initial values of parameters of the structural model were obtained by Maximum Likelihood fitting of the respective distributions to values computed on the annotated training image set described in Sec. 2.7. In our case this however resulted in too concentrated pdfs (low variance), which did not perform well during inference (low mixing rate).

We therefore performed grid search with several higher variance parameter values and picked up those which performed best on the training set (highest accuracy) shown in Tab. 2.1. This also helped to establish balance between individual parts of the model.

For this setting we have verified our model $p(k, \bar{\theta}, \dot{\theta})$ by constructing a random sample generator from the distribution, generating a sequence of 10^6 samples and selecting the most probable sample in the sequence. As expected, we got a regular configuration shown in Fig. 2.6.

2.5 Data Model

The input image $I = (i;\ i = 1, \ldots, I_w \cdot I_h)$ is defined as a set of pixels and we assume it is rectified, i.e. the window borders are parallel to the image borders, and I_w, I_h are image width and height. Although our model parameters are continuous relative to the image frame, we will discretize them to evaluate image data pixel-wise. This can be seen as allocation of pixels to components.

In the data likelihood model $p(I \mid k, \bar{\theta}, \dot{\theta})$ we express the probability of observing an image

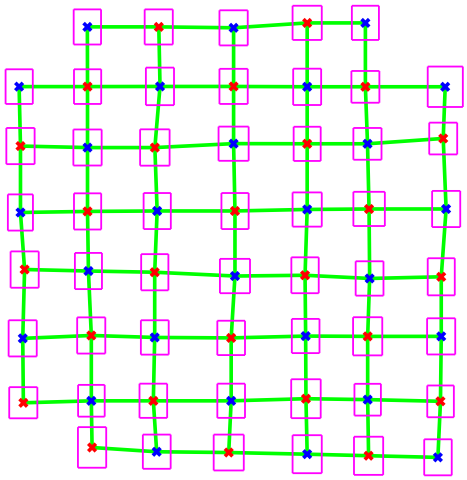


Figure 2.6: A random sample close to the mode of a Softly Bipartite Graph. Nodes on positions μ are marked with crosses colored red or blue according to labels B. Edges with $l_{uv} = 1$ are in green, component shapes are in magenta.

I given a configuration $(k, \bar{\theta}, \dot{\theta})$. We combine two independent features: image edges J and color C in

$$p(I \mid k, \bar{\theta}, \dot{\theta}) = p(J \mid k, \bar{\theta}, \dot{\theta})\, p(C \mid k, \bar{\theta}, \dot{\theta}). \tag{2.31}$$

We use weak color information to detect regions of interest and image edge features for precise localization of the window borders.

2.5.1 Image Edge Model

We assume that window borders correspond to edges, and use Canny detector to find them. However, this model will not fully hold in real world situations, when we obtain the input by detecting edges in a picture—there can be windows which do not have all pixels with underlying edges and vice versa, some edges do not belong to any windows at all. The latter case will typically prevail.

We use binary imaging model for window edges represented by oriented edge image $J = \{J_i \in \{0, h, v\}\,;\, i \in I\}$, where $J_i = h$ if pixel i belongs to a horizontal edge detected in I (foreground), resp. $J_i = v$ for vertical edge; otherwise $J_i = 0$ (background). We define $d(J) \in [0, 1]$ as a distance transform of the edge image J normalized by $\max(I_h, I_w)$, see Fig. 2.8. We use the gradient of $d(J)$ to distinguish between horizontal and vertical edges.

Similarly, we introduce edge image $R(\bar{\theta})$ rendered from the current configuration specified by $\bar{\theta}$ and the shape template in Fig. 2.5 with nearest neighbor discretization of relative

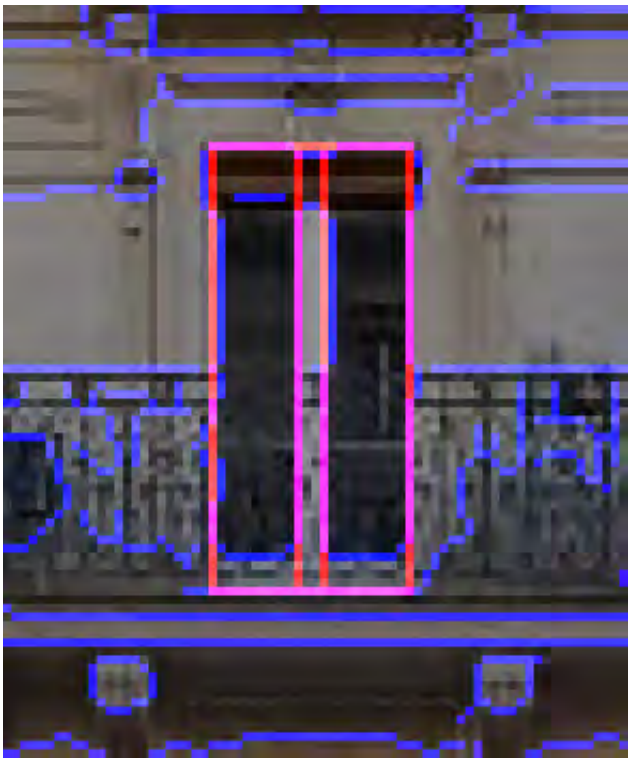


Figure 2.7: The shape template (red) is matched with image edges (blue)

parameters $\bar{\theta}$ into pixel domain I. Assuming pixel independence, we can write

$$p(J \mid \bar{\theta}) = \prod_{i \in I} p(J_i \mid R_i(\bar{\theta})) \tag{2.32}$$

where the probability of observing a pixel i in the edge image J given the rendered configuration R is given by

$p(J_i \mid R_i)$	$J_i = 0$	$J_i = h$	$J_i = v$
$R_i = 0$	$p_0 = 0.8$	$p_n = 0.1$	$p_n = 0.1$
$R_i = h$	$p_d(d(i))\,(1 - p_x)$	$p_d(0)\,(1 - p_x)$	p_x
$R_i = v$	$p_d(d(i))\,(1 - p_x)$	p_x	$p_d(0)\,(1 - p_x)$

Each row in this table is a conditional probability summing to one. In the case of $R_i \in \{h, v\}$ it is a mixed distribution of explicit penalty p_x for edge orientation mismatch and a continuous Beta pdf based on edge distance $d(i) \geq 0$

$$p_d(d(i)) = \mathrm{Be}(d(i);\, \beta_d = 500, 1)$$

which makes rectangles close to edges more probable and acts as a guide for directing the random walk in the inference (Sec. 2.6). The $p_x = 10^{-9}$ is the probability assigned when the edge specified by the configuration crosses an image edge in the opposite direction (horizontal $\times$vertical).

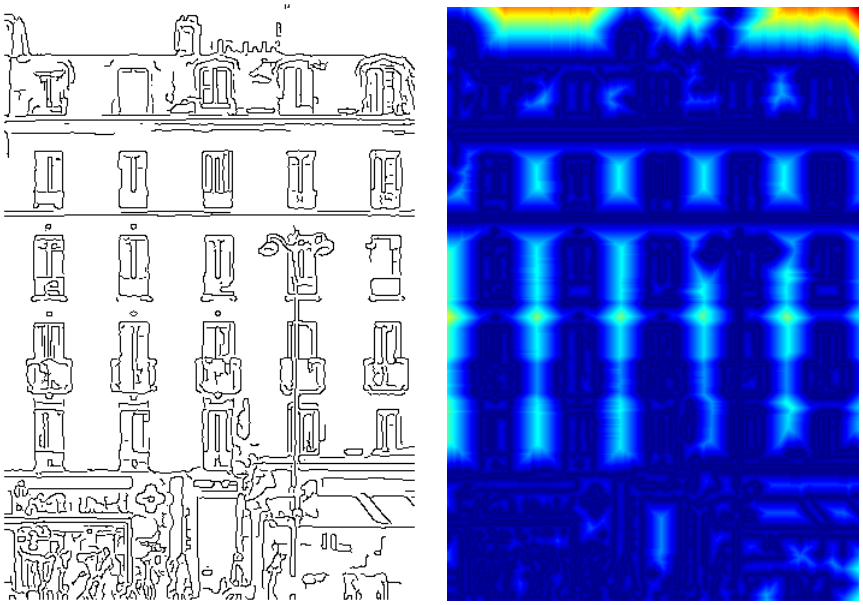


Figure 2.8: Sample edge image J (left) and its distance transform $d(J)$ (right), where bright colors indicate pixels far from edges corresponding to high penalty for component edges passing through them.

The edge terms can be efficiently evaluated from pre-computed integral edge images, one for each orientation h, v, yielding constant computational complexity $\mathcal{O}(1)$ per edge; this speed-up is possible thanks to rectified images and helps make random sampling (described in Sect. 2.6) very efficient.

2.5.2 Image Color Model

We extend the simple color model from Tyleček and Šára (2011a) and model the input color image $C = (c_i \in [0,1]^3;\ i = 1, \dots, I_w \cdot I_h)$ with a multivariate Gaussian mixture distribution with $m = 3$ components that targets the 'window' class. We use the configuration θ to partition pixels either to foreground (window) set C_f or background (non-window) set C_b such that $C_f \cap C_b = 0$. Assuming pixel independence, the probability of observing a segmented image is

$$p(C \mid \bar{\theta}) = \prod_{i \in C_b} p_b(c_i) \prod_{j \in C_f} p_f(c_j), \tag{2.33}$$

where the background probability $p_b(c_i) = p_b = 1$ is uniformly constant on the unit domain and the foreground color model is expressed by

$$p_f(c_j) = \sum_{i=1}^{m} \omega_j \, \mathcal{N}(c_j \mid \mu_i^c, \Sigma_i^c). \tag{2.34}$$

The mixture parameters $\omega_j, \mu_i, \Sigma_i$ are learned as ML estimates obtained with the EM algorithm (Dempster, A.P. et al., 1977) by fitting color of 'window' class pixels sampled from the annotated training image set.

Like in edge model, color is evaluated using pre-computed integral images in linear time, we query four values per component. As (2.33) suggests, we evaluate foreground pixels only.

2.6 Inference

We have chosen Reversible Jump MCMC framework (Green, 1995) that fits our task of finding the most probable interpretation of the input image in the terms of target probability $p(\theta, I)$ in (2.1), which has a very complex pdf as it is a joint probability of both shape, locations and structure. This approach has been used by independent researchers in similar inference tasks with variable dimensions (Ripperda and Brenner, 2007, 2009). Our solution θ^* is found as the most probable parameter value $\theta = (k, \bar{\theta}, \dot{\theta})$ the chain visits in a given number of samples. The result is a naive MAP estimation of the number of components by direct maximization of a posterior of variable dimension (Sec. 1.3.3.2) and the procedure requires the probabilities over terms with variable dimensions to be properly normalized in order to compare configurations with different complexity. The MAP choice has alternative in the two-level inference (Sec. 1.3.3.4) that will be applied in Chapter 4.

As suggested in Sec. (2.4) our probability model is however not fully normalized due to the Gibbs distributions in the spatial regularity part (2.18) and (2.24), which are defined up to a constant depending on N, which is generally hard to estimate.

While the MH algorithm itself is simple, we need to carefully design proposal distribution q that should approximate target distribution $p(\theta, I)$ well for the efficient sampling. We should point out that the quality of the resulting interpretation is determined by the probability model, on the other hand the time necessary to reach the solution is influenced by the proposal distributions. It turns out that by exploiting the estimated structure we can efficiently guide the random walk of our chain by repeatedly sampling the new state θ' from the vicinity of the current state using conditional probability $q(\theta' \mid \theta)$.

The conditional sampler $q(\theta' \mid \theta, I) \rightarrow \theta'$ is a mixture of individual samplers such that each modifies a subset of parameters θ based on a specific proposal distribution $q_m(\theta' \mid \theta, I)$. The top-level sampler only chooses from $q(m \mid \theta)$ which of the individual samplers m will be used to propose the next move. Their design must fulfill Markov Chain properties of detailed balance and reversibility of all moves (Winkler, 2003), i.e. given a move there must always exist a reverse move m', and their probability ratio must be reflected in the acceptance ratio of Metropolis-Hastings (MH) algorithm (Sec. 1.3.4.2). The chain is initialized with $k = 0$, then the only allowed proposal is to add a new component (Sec. 2.6.3).

2.6.1 Proposal Selection

The sampler mixture distribution $q(m)$ is constructed hierarchically, we first choose a probability $q_{RJ} = 0.1$ of reversible jump proposals, from which it follows that the ordinary MH jumps have $q_{MH} = 1 - q_{RJ} = 0.9$. In the second step, we choose uniformly one of the jumps

from the appropriate set of proposals (either $q(m \mid MH)$ or $q(m \mid RJ)$) presented in the next Sections 2.6.2 and 2.6.3.

Proposing dimension changes is expensive, therefore we adapt the proposal distribution according to the current state to achieve a speed up by reducing reversible jumps. This is done by constructing a conditional distribution

$$q_t(RJ \mid \theta_t) = q_{RJ} + Te^{-\frac{t}{\tau}}, \tag{2.35}$$

we choose in practice $T = \frac{1}{4}$, $\tau = 10^4$. The vanishing adaptation (i.e. $q_t(RJ \mid \theta_t) \to q_{RJ}$) guarantees convergence of the chain even if it is no longer ergodic due to its adaptation (Andrieu and Thoms, 2008).

2.6.2 Metropolis-Hastings Moves

The moves introduced in this section perform size, shape or location modifications, thus do not modify the model complexity k and can be evaluated by a classical MH algorithm (Sec. 1.3.4.2).

2.6.2.1 Size and Location Modification

This move picks up a component $i \sim \mathcal{U}(k)$ from a discrete uniform distribution and perturbs some of its parameter values randomly. Additionally, these samplers can be designed to exploit image data to increase the acceptance rate. In the window detection scenario, we have implemented three variants for this type of proposals (also see Fig. 2.9):

- *Drift* - random variation of position by $\Delta \sim \mathcal{N}(0, \sigma_\Delta)$ without changing the size,

$$\mu_i' = \mu_i + \Delta. \tag{2.36}$$

- *Resize* - change size by randomly picking up one of four rectangle sides (left/right/top/bottom) or corners and moving it by $\Delta \sim \mathcal{N}(0, \sigma_\Delta)$

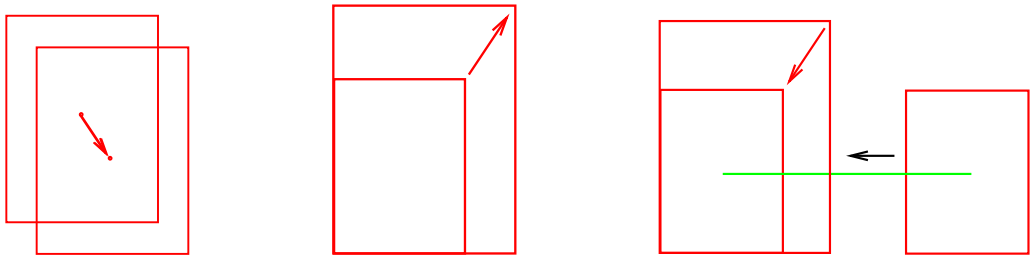
$$\bar{\theta}_i' = \bar{\theta}_i + \Delta. \tag{2.37}$$

  The *drift* and *resize* both propose similar local changes and share the same σ_Δ in order to reduce the number of free parameters in the method.

- *Flip* - fix one of the rectangle sides and flip the window around it, size is not changing,

$$\mu_i' = \mu_i \pm w_i \quad \text{or} \quad \mu_i' = \mu_i \pm h_i. \tag{2.38}$$

  This allows for faster exploration of the configuration space when the fixed side is matching a salient image edge (*drift* has small acceptance in this case).



a) *Drift* b) *Resize* c) *Inherit Size*

Figure 2.9: Illustration of component parameter proposals. Components are red rectangles, red arrows mark the change. Green edge connects two components and the black arrow indicates inheritance of size.

Instead of these purely random proposals, it would be possible to implement a more advanced Langevin diffusion (Marshall and Roberts, 2012), which can be adapted for optimal performance but involves computation of the gradient of log probability (ev. also the Hessian). Particularly in the case of oriented edge model (2.32) there are some difficulties, which prevented us from using it in experiments for this problem, but it will be applied in the following chapter.

2.6.2.2 Component Resampling

This move is a more radical variant of the previous one, we pick up a component i and change of all its parameters by sampling from the prior distribution $\sigma_i' \sim p_1(\sigma_i)$ given in (2.24).

2.6.2.3 Inherit Size

This move is in spirit similar to the exchange of genes in genetic algorithms (crossover). It proposes changes to the parameter of a component according to a chose neighbor,

$$\bar{\theta}_i' \sim q(\bar{\theta}_i \mid \bar{\theta}, N). \tag{2.39}$$

We uniformly choose a random edge (u, v) and transfer a randomly selected component parameter (μ,h,w or t) value over the edge from one component to another according to the specific constraints, resulting in proposal such as $h_u' = h_v$ or $\mu_u' = \mu_v$ (see Fig. 2.9c).

2.6.2.4 Switch Edge

A move to allow changes to the neighborhood structure picks up a random edge $(u, v) \sim q(u, v \mid \bar{\theta})$ and changes its label $l_{uv}' = 1 - l_{uv}$, effectively suppressing or recovering the given edge.

The edge proposal $q(u, v \mid \bar{\theta})$ is an empirical distribution on $\{\frac{1}{\rho_{uv_i}};\, v_i \in N(u)\}$ to prefer nodes closer to each other, reflecting the idea of proximity of neighbors.

2.6.2.5 Switch Node Color

When SBG is used this move picks up a random node $i \sim q_b(i \mid N)$ and changes its node color to $b'_i = 1 - b_i$. The distribution $q_b(i \mid N)$ is constructed to prefer nodes i from a set where the two-coloring property of softly bipartite graph is violated, i.e. some of its neighbors u have the same color $b_u = b_i$. We choose from this set with $q_b = 0.9$.

2.6.3 Reversible Jump Moves

An inseparable part of our task is to find the number of components k that controls the dimension of active component parameters $\bar{\theta}$. While the number of variables is fixed with k_m (Sec. 2.4) in practice the change of k means activation or inactivation of components. Activation process is however equivalent to sampling of the component parameters for a new components and we will use the standard RJ terminology including 'dimension matching' in the following text even when it is not precise.

The standard MH acceptance (Sec. 1.3.4.2) has to be extended for RJ with

$$A = 1 \wedge \frac{p(I \mid \theta')\, p(\theta')}{p(I \mid \theta)\, p(\theta)} \cdot \underbrace{\frac{q(m \mid \theta')}{q(m' \mid \theta)}}_{a_m} \cdot \underbrace{\frac{q_m(\theta \mid \theta')}{q_m(\theta' \mid \theta)}}_{a_q} \cdot \underbrace{\frac{q_{\leftarrow}(u_{\leftarrow} \mid \theta')}{q_{\rightarrow}(u_{\rightarrow} \mid \theta)}}_{a_u} \cdot J_{\rightarrow}, \tag{2.40}$$

where a_m reflects the choice of individual samplers, a_q is the proposal density ratio ($a_q = 1$ when the proposals are symmetric), a_u and $J_{\rightarrow}$ are related to complexity changes in reversible jumps (described below). The proposed move is accepted with probability $A \in (0, 1]$ (given by truncated probability ratio). In order to compare the models in (2.40) we need to define dimension matching functions $q_{\rightarrow}, q_{\leftarrow}$ for both direct and reverse moves, where $\rightarrow$ refers to direct move, $\leftarrow$ to reverse move, u are dimension matching (communication) variables and

$$J_{\rightarrow} = \left| \frac{\partial f_{\rightarrow}(\theta, u_{\rightarrow})}{\partial(\theta, u_{\rightarrow})} \right| \tag{2.41}$$

is the Jacobian of the transformation, following the notation given in Green (1995).

There is a set of edges and neighborhood variables l_{uv} associated with each (in)activated component, concretely all edges linking a removed component are suppressed ($l_{uv} = 0$) and corresponding proposal pdf $q(u, v)$ from Sec. 2.6.2.4 must be included in the acceptance ratio. When activating a component its associated edges stay suppressed, unless otherwise specified below. If some edge is enabled ($l_{uv} = 1$), its proposal pdf is also included in (2.40).

2.6.3.1 Birth and Death

By inserting a new component into our model we propose an increase of dimension $k \mapsto k' = k + 1$. We choose the communication variables to be $u_{\rightarrow} = [\sigma_*, \mu_*]$, where we sample the parameters of the new component $\bar{\theta}_* = (\sigma_*, \mu_*) \sim q(\sigma, \mu)$ and obtain a new state where

we append them[5] in $\sigma' = (\sigma, \sigma_*)$ and $\mu' = (\mu, \mu_*)$. The corresponding dimension matching function is

$$f_{\rightarrow}(\bar{\theta}, u_{\rightarrow}) = f_{\rightarrow}(\bar{\theta}, \bar{\theta}_*), \tag{2.42}$$

which inserts $\bar{\theta}_*$ into the set, and its Jacobian $J_{\rightarrow} = 1$. We will use the following notation within this section: terms in $[\dots]$ refer to communication variables and terms in $\{\dots\}$ to parameters.

The reverse move is *death*, for which we have no communication variable $u_{\leftarrow} = [\,]$ (empty), only choose a component i to be removed from the set. To establish reversibility, we define inverse matching function as

$$f_{\leftarrow}\left(\bar{\theta}', u_{\leftarrow}\right) = f_{\leftarrow}\left(\bar{\theta}', [\,]\right), \tag{2.43}$$

where σ_i, μ_i are the removed[6] variables and $\sigma = \sigma' \setminus \sigma_i$, $\mu = \mu' \setminus \mu_i$. The corresponding birth move acceptance is then

$$a_{birth} = \frac{p(\theta', I)}{p(I)} \frac{q(m \mid \theta')}{q(m' \mid \theta)} \cdot \frac{q(i \mid k')}{q(* \mid k)} \cdot \frac{1}{q_{\rightarrow}(\sigma_* \mid \sigma)} \cdot 1, \tag{2.44}$$

where $q_{\rightarrow}(\sigma_* \mid \sigma) = p_1(\sigma)$ is directly the prior probability of the new window (2.25), $q(i \mid k') = \frac{1}{k'}$ and $q(* \mid k) = \frac{1}{k}$ are the probabilities of selecting the windows $*, i$.

By removing an existing component from the set (*death*) we propose a decrease of dimension $k \mapsto k' = k - 1$, and choose a window $i \sim \mathcal{U}(k)$ to be removed. With an appropriate change of labeling, the derivation of death move will be the same as for birth, except for the inversion of ratios in (2.44) and corresponding reindexation.

In the basic case of *birth* the new position μ_* is sampled uniformly and the new size parameters are sampled from the prior $\sigma_* \sim p_1(\sigma)$. The jumps detailed below are special cases of *birth* that exploit the structure of the current configuration for predicting values of the new components, which can be generally described as sampling from $\bar{\theta}_* \sim q(\bar{\theta} \mid N)$. We designed them to sample from the marginal distributions of the structural model where possible, which is expected to have a high acceptance probability A resulting in more efficient exploration of the configuration space (mixing).

2.6.3.2 Append

In this case of the *birth* jump we attempt to predict a location for the new component based on the prior information. We first choose uniformly an existing component $i \sim \mathcal{U}(k)$ and

[5] Recall that σ, μ without subscripts are parameter arrays and subscripted $\sigma_\circ, \mu_\circ$ are parameters of a single component. By $(x, x_\circ)$ we mean the element $x_\circ$is appended after the last item in array x.

[6] The $x \setminus x_i$ indicates removal of the i-th element from the array x, i.e. we extend the set operator $\setminus$ for arrays.

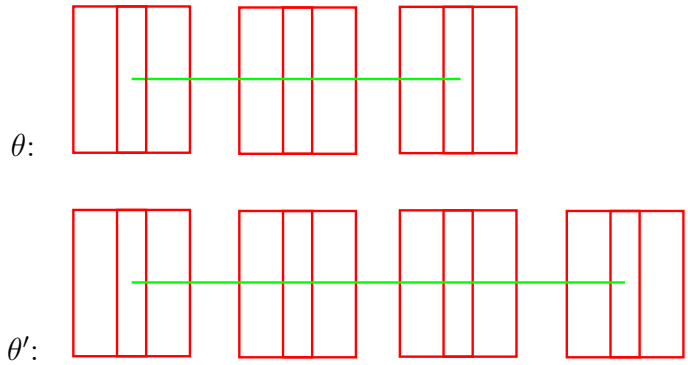


Figure 2.10: *Replicate* extends an array of components following its orientation and spacing.

place the new component relatively to its position according to

$$\mu_* = \mu_i + \rho\nu(\varphi), \tag{2.45}$$

where $\nu(\varphi) = [\sin\varphi, \cos\varphi]$, we sample $\rho \sim p(\varrho_{uv})$ and $\varphi \sim p(\varphi_{uv} \mid l_{uv} = 1)$ from the priors (2.19) and (2.20). Its shape parameters σ_* are sampled relatively to σ_i from the marginal Beta distribution (2.29) of similarity by $\delta \sim p_2(\sigma \mid N)$ and then

$$\sigma_* = \sigma_i \frac{1-\delta}{\delta}. \tag{2.46}$$

We explicitly set the edge $l_{i*} = 1$ and the Jacobian here is $J_\rightarrow = \rho$.

2.6.3.3 Replicate

This jump is similar to *append*, but we predict the new position based on the existing structure i.e. to add a new component to the end of an array (see Fig. 2.10). We uniformly sample an edge (u, v) and set the new window position to

$$\mu_* = \mu_v + \rho_{uv}\nu(\varphi_{uv}), \tag{2.47}$$

where ρ_{uv} and φ_{uv} are taken from the sampled edge. The size is replicated by taking the mean of the two sampled components

$$\sigma_* = \frac{1}{2}(\sigma_u + \sigma_v). \tag{2.48}$$

The Jacobian is here $J_\rightarrow = \rho_{uv}$.

2.6.3.4 Extend

The above introduced proposals have low acceptance when a single new component is added as the first one in a new row or column (in the regular case), because the structure prior puts a low probability on this configuration. Adding two components at a time can be more

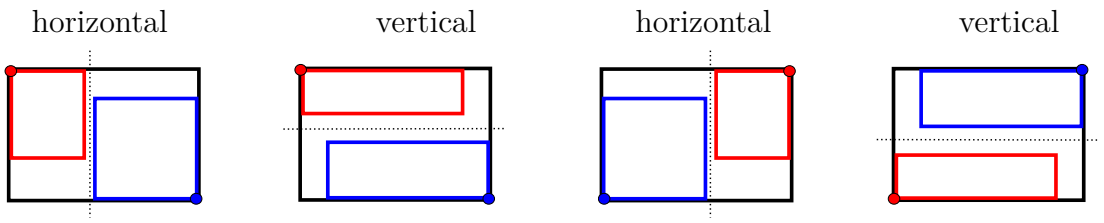


Figure 2.11: For *split* first a diagonal (basic/flipped, indicated by points in corners) is chosen and then the orientation (horizontal/vertical, indicated by dotted line), resulting in four possible scenarios. The *black* rectangle can be split into two rectangles (*red* and *blue*), or inversely *red* and *blue* rectangles can be merged to the *black* rectangle (their common bounding box). In the general non-overlapping case there are four possible scenarios.

successful, so in this case we add two new components $*_1, *_2$ at once and connect them with edges to create a new four-cycle in the graph N. We uniformly sample an edge (u, v) and set the new positions to

$$\mu_{*_1} = \mu_u + \rho_{uv}\nu(\varphi_*), \tag{2.49}$$

$$\mu_{*_2} = \mu_v + \rho_{uv}\nu(\varphi_*), \tag{2.50}$$

where $\varphi_* = \varphi_{uv} \pm \frac{\pi}{2}$ and the sign is chosen uniformly. The size parameters are replicated from σ_u to σ_{*_1} and σ_v to σ_{*_2}. The face is completed by activating edges $l_{u*_1} = l_{v*_2} = l_{*_1*_2} = 1$.

2.6.3.5 Split and Merge

The *split* move proposes increase of dimension $k \mapsto k' = k + 1$, where an existing component is transformed into two new ones. This move is a shortcut for an equivalent sequence of *drift* and *birth* detailed above. *Split* is expected to have higher acceptance than the partial moves combined, because the intermediate configurations have low probability. The same applies to the inverse move *merge* which shortcuts *death* and *drift* by replacing two neighboring components by their bounding box; this merging procedure has impact on the split procedure, because they have to be exactly reversible. There are four splitting scenarios corresponding to the relative position of the two split or merged components in Fig. 2.11 and we need to sample them all in order to have inverse *split* for any *merge* move and vice versa.

To simplify the calculations we will work with the component rectangles represented by upper-left and lower-right corners B (bounding box), which can be obtained from the location and size parameters $\bar{\theta}_i = (\mu_i, w_i, h_i, t_i)$ using

$$B(\bar{\theta}_i) = B_i = \left[\mu_{i1} - \frac{w_i}{2}, \mu_{i2} - \frac{h_i}{2}, \mu_{i1} + \frac{w_i}{2}, \mu_{i2} + \frac{h_i}{2}\right], \tag{2.51}$$

$$B(\bar{\theta}) = (B_1, \ldots, B_k). \tag{2.52}$$

We choose and fix the component $v \in \{1, \ldots, k\}$ to be split, the split direction (horizon-

tal/vertical) and sample the split factors $s_{ij} \in (0, 1)$, which describe locations of the two split rectangles relative to the original rectangle (bounding box), as shown in Fig. 2.12. They are sampled from the beta distribution as the communicating variables

$$u_{\rightarrow} = [\, s_{11}\, s_{12}\, s_{21}\, s_{22}\,] = \mathbf{s}.$$

The beta pdf parameters are chosen according to the given split scenario, i.e. for the horizontal scenario

$$s_{11}, s_{12} \sim \mathrm{Be}(\beta_{s1}, \beta_{s1}),\ \ s_{21} \sim \mathrm{Be}(1, \beta_{s2}),\ \ s_{22} \sim \mathrm{Be}(\beta_{s2}, 1).$$

The corresponding dimension matching function is then

$$f_{\rightarrow}(B, u_{\rightarrow}) = f_{\rightarrow}\Big(B, [s_{11}\, s_{12}\, s_{21}\, s_{22}]\Big) = \Big(\{B, B^*\}, [\,]\Big) = \Big(B', [\,]\Big), \tag{2.53}$$

which in the basic horizontal scenario modifies $B_v = \Big[b_{11}\, b_{12}\, b_{21}\, b_{22}\Big]$ into

$$B'_v = [b_{11} + s_{12}w_v,\ b_{12} + s_{21}h_v,\ (1 - s_{12})w_v,\ (1 - s_{21})h_v], \tag{2.54}$$

and inserts new B^* into the set of components

$$B^* = [b_{11}, b_{12},\ s_{11}w_v, s_{22}h_v].$$

The case for flipped or vertical orientation is derived analogically. The parameters B' for other components than v are copied from B and the Jacobian

$$J_{\rightarrow} = \left|\frac{\partial(B', B^*)}{\partial(B, B^*)}\right| = w_v^2\, h_v^2 \tag{2.55}$$

is calculated given v from the variables that actually change: $B_v \mapsto B'_v$. The other scenarios yield the same result.

The inverse move is *merge*, for which we have no communication variable $u_{\leftarrow} = [\,]$ (it is deterministic), and choose the two neighboring components $B_v, B^\dagger \in B'$ to be merged into one. To establish reversibility, we define inverse matching function as

$$f_{\leftarrow}(B', u_{\leftarrow}) = f_{\leftarrow}\Big(\{B_{\neg v}, B'_v, B^\dagger\}, [\,]\Big) = \Big(\{B\}, \mathbf{s}\Big) \sim \Big(B, u_{\rightarrow}\Big), \tag{2.56}$$

where $B^\dagger$ is the removed component and B_v is the merged component, $B = \{B' \setminus B^\dagger\}$. The split configuration is detected and ratios $\mathbf{s}$ are calculated from the affected component pair $B_v, B^\dagger$, inversely to (2.54). In the split move acceptance we now have $a_u = \frac{1}{q_{\rightarrow}(\mathbf{s})}$, where

$$q_{\rightarrow}(\mathbf{s}) = p(s_{11})\, p(s_{12})\, p(s_{21})\, p(s_{22}) \tag{2.57}$$

is the prior probability of the split and $\alpha_q = \frac{k}{k+1}$ reflects component selection.

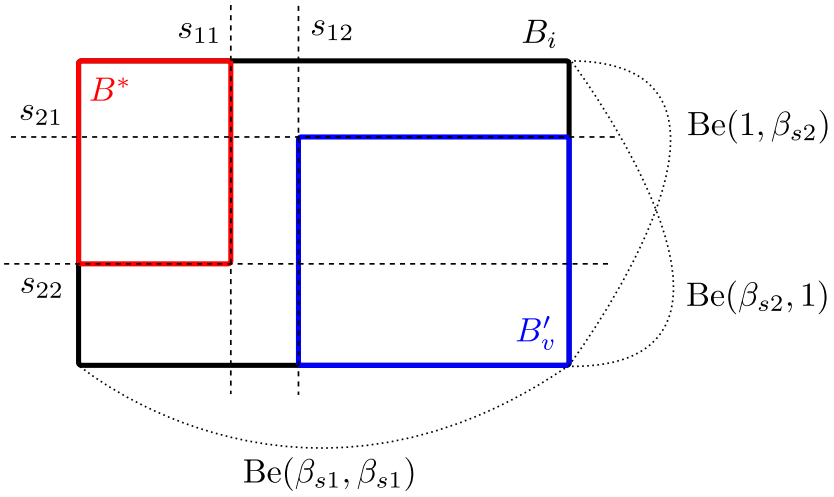


Figure 2.12: Split bounding box B_i (black) with relative split locations $\mathbf{s}$ (dashed) for horizontal scenario and proposal Beta pdfs (dotted, approx.).

For *merge*, where $k \mapsto k' = k - 1$, the merged component rectangle B'_v is a bounding box of the merged components $B_v, B^\dagger$ and the Jacobian is now

$$J_{\rightarrow} = \left| \frac{\partial(B', \mathbf{s})}{\partial(B)} \right| = \frac{1}{w_v^2 \, h_v^2}. \tag{2.58}$$

Again, with appropriate change of labeling, the derivation of *merge* move is the same as for *split*, except for the inversion of ratios, i.e. $a_q = \frac{k+1}{k}$ and $a_u = q_{\rightarrow}(\mathbf{s})$, where the corresponding split factors $\mathbf{s}$ must be calculated from the input configuration. The pair of components to merge is sampled from the edge proposal $q(u, v \mid \bar{\theta})$ in Sec.2.6.2.4 to components close to each other.

2.6.4 Convergence and Complexity

The overview of implemented proposals is given in Tab. 2.2.

We have found that the typical necessary number of MCMC samples (and classifier calls) is proportional to image size in pixels $|I|$ (from 30% for easy instances to 200% for the difficult ones). As a result, we fixed the number of samples in our current method to a pessimistic estimate, but our experiments suggest that significantly shorter sampling time could be achieved with a suitably designed stopping condition (see Fig. 2.13). Another option is to use a more efficient sampling scheme, i.e. Duane et al. (1987) for the continuous part or Barbu and Zhu (2005) for the discrete variables (labels).

2.7 Experimental Results

We have performed a number of experiments with the implementation of window detection in facades of various styles to demonstrate the universality of our approach. We have run

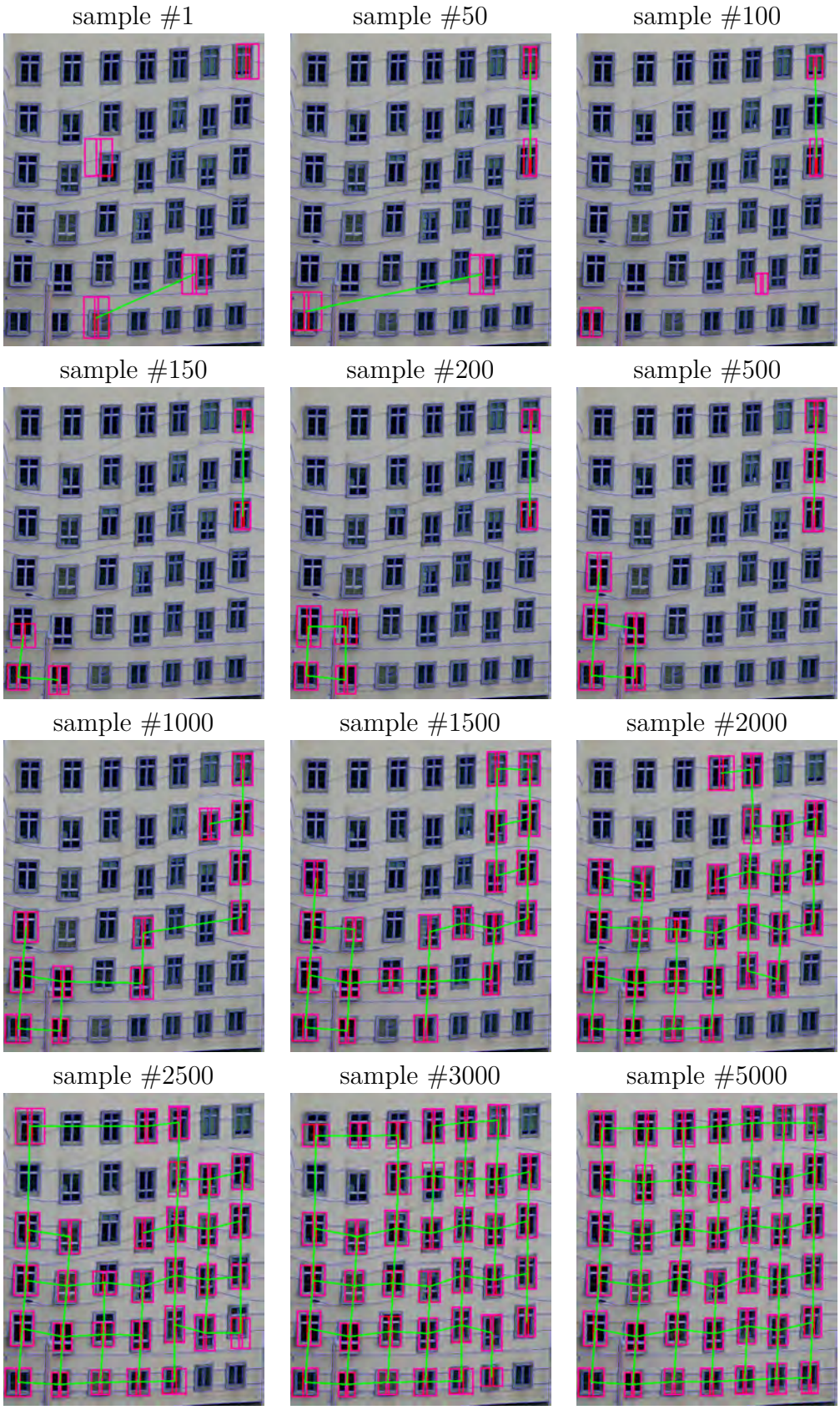


Figure 2.13: Inference progress of the proposed RJMCMC sampler. Detected windows are shown in red, neighborhood edges in green and image edges are emphasized in blue.

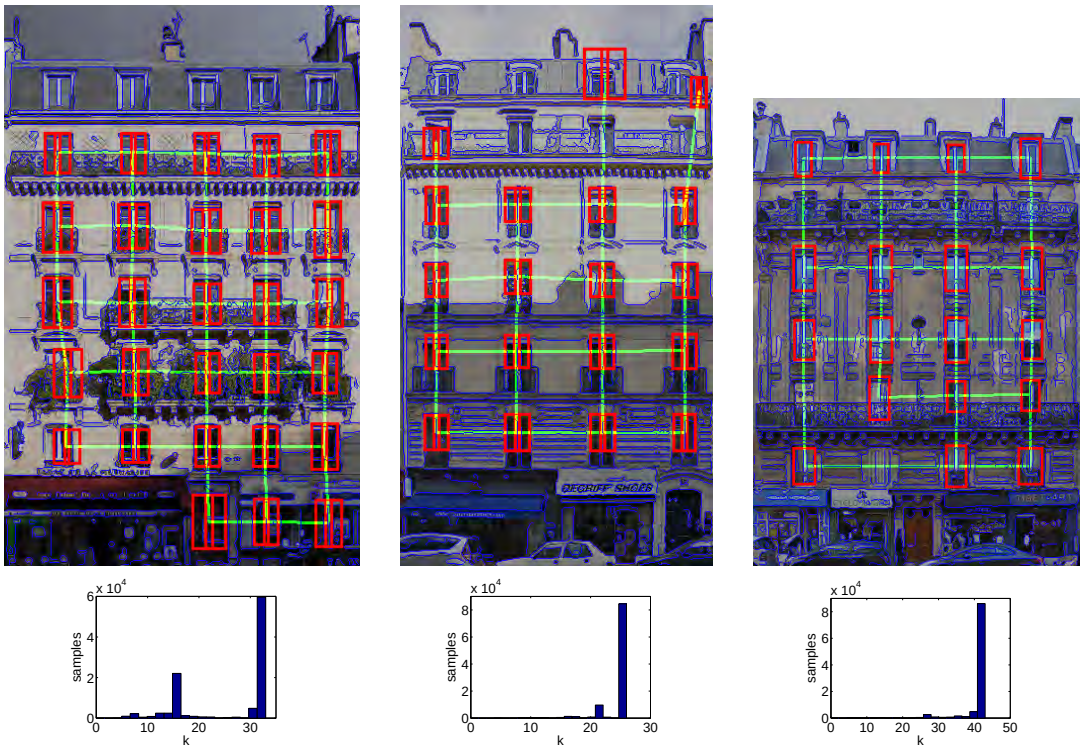
Parameter		*Move*	*Proposal pdf and parameters*	
μ	$\bar{\theta}$	drift	$\mathcal{N}$	σ_Δ
σ, μ	$\bar{\theta}$	resize	$\mathcal{N}$	σ_Δ
μ	$\bar{\theta}$	flip	-	-
σ	$\bar{\theta}$	resample	$p_1(\sigma)$	see (2.24)
σ, μ	$\bar{\theta}$	inherit	$q(\bar{\theta}_i \mid \bar{\theta}, N)$	-
N	$\dot{\theta}$	switch edge	$q(u, v \mid \bar{\theta})$	-
B	$\dot{\theta}$	switch color	$q_b(i \mid N)$	$q_b = 0.9$
k	+1	birth	$\mathcal{U}(\mu)$, $p_1(\sigma)$	see (2.24)
k	-1	death	$\mathcal{U}(k)$	-
k	+1	append	$p(\varrho_{uv})$, $p(\varphi_{uv})$, $p_1(\sigma)$	(2.19), (2.20)
k	+1	replicate	$\mathcal{U}(N)$	-
k	+2	extend	$\mathcal{U}(N)$	-
k	$1 \to 2$	split	Be	$\beta_{s1} = 2$, $\beta_{s2} = 10$
k	$2 \to 1$	merge	$q(\bar{\theta}_i \mid \bar{\theta}, N)$	-

Table 2.2: List of proposals and their typical acceptance ratios. Reversible jumps changing dimension are in the bottom part of the table.

the Markov Chain for fixed 5×10^5 iterations in our experiments, which roughly equals to visiting all pixels in the analyzed images. With our Matlab implementation, the running time was under one minute on a standard 2 GHz CPU.

The only public dataset known to us that allows quantitative comparison in this area has been provided by Teboul et al. (2010). The dataset consists of 30 rectified and annotated images of facades from a street in Paris, which share attributes of Haussmannian style but differ in illumination conditions. We have trained our model on 20 of them and 10 were used for testing. Direct comparison is not possible, because they segment facade pixels into eight different classes of components and our window detector defines only two (window/non-window). To deal with this issue, we have used a similar reduction as in Tyleček and Šára (2011a) and merged the columns of confusion matrix given in Teboul et al. (2010) into two, treating all original classes other than *window* as our background (non-window).

The results in Tab. 2.3 for *window* and *wall* suggest that the proposed method is performing better in the terms of high specificity when compared to the procedural segmentation (PS) framework (Teboul et al., 2010) see Fig. 2.15. We attribute this to the extended color model model with Gaussian mixtures in the HSV color space (which is less sensitive to the illumination changes), on the other hand, it resulted in a small drop in sensitivity to the window class. The new bipartite structural model with parameters learned from the annotations also contributed to the results, it is able to support windows completing the structure even where the data response is low. This allows us to achieve good results even when the illumination varies and partial occlusion of windows is present, as shown in Fig. 2.7.



$H = 1.3705$ | $H = 0.6632$ | $H = 0.6920$

a) Monge No. 13 | b) Monge No. 43 | c) Monge No. 50

Top row: Visualization of selected results from Parisian dataset (Teboul et al., 2010), facade a) is occluded by plants, in facade b) a cast shadow is present. False positive windows are also window-like regions: They have good response from both classifiers and match with the neighbors. *Bottom row:* Posterior histograms for complexity k.

The difference between RNG and SBG is in favor of the latter particularly due to less false positive detections as a result of using less restrictive graph prior which allowed to find better balance between the parts of the probability model (Sec. 2.4.4).

Posterior histograms shown in Fig. 2.7 for complexity k demonstrate different difficulty of the images, which is quantified by estimated entropy H. In the case of a) there is another less probable interpretation for $k = 15$ (missing some rows of windows), resulting in higher H.

To prove our framework is not limited to a particular style, we demonstrate results on modern buildings and even hand drawn images in Fig. 2.16 and Fig. 2.14. Note the appearance of edges in Fig. 2.16a) connecting the 'shifted' middle column, which was not possible in Tyleček and Šára (2011a) due to the RNG constraint. The shape parameter t in Fig. 2.16b) which was fixed in Tyleček and Šára (2011a) is now inferred along with the other parameters of the model.

Finally, we have made experiments with loosely regular facade of *Dancing House* shown in Fig. 2.14a), where window alignment shows significant deviation from the grid structure and we were successful in correctly locating all windows lying on the major plane as well as their neighborhood.

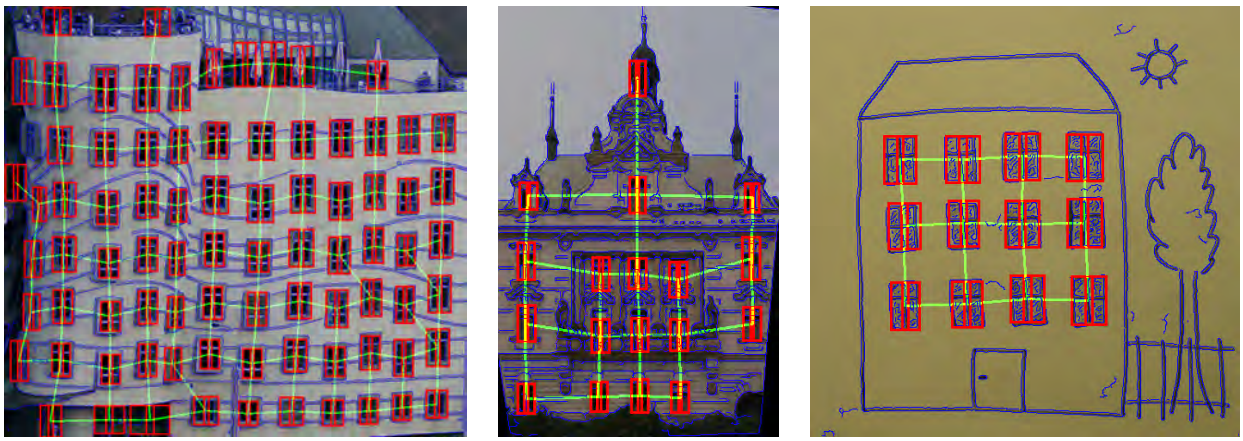
Ground Truth		*PS*		*RNG*		*SBG*	
class	area	hit	miss	hit	miss	hit	miss
window	11	81	19	83	17	76	24
wall	48	83	17	84	16	98	2
balcony	12	72	28	*60*	*40*	*89*	*11*
door	1	71	29	*65*	*35*	*100*	*0*
roof	4	80	20	*51*	*49*	*95*	*5*
chimney	1	0	100	*83*	*17*	*96*	*4*
sky	7	94	6	*99*	*1*	*100*	*0*
shop	14	95	5	*60*	*40*	*99*	*1*
other	2	0	100	*61*	*39*	*96*	*4*
area-weighted		81	19	77	23	93	7

Table 2.3: Quantitative results on the Parisian dataset (Teboul et al., 2010) shown as percentage of pixels from each class specified in a row. The area is the percentage of pixels of a given class in the whole test set. PS stands for Procedural Segmentation (Teboul et al., 2010), RNG for Relative Neighborhood Graph (Tyleček and Šára, 2011a), SBG for Softly Bipartite Graph Tyleček and Šára (2012). Percentage in italics indicate remapping to window/wall classes described in Tyleček and Šára (2011a).

2.8 Conclusion

We have presented a recognition framework that uses a weak structure model to locate components in images, and demonstrated its potential in the task of window detection in facades. Our experiments have demonstrated that structural regularity given by pair-wise parameter constraints can efficiently guide a stochastic process that estimates component locations and neighborhood at the same time. We have shown that the conjunction of a weak non-specific classifier and a weak structural model can lead to performance that would be hardly achievable by a well-trained specific classifier alone.

In practice we have faced difficulties to tune model hyperparameters (Tab.2.1) with proposal parameters (Tab. 2.2) for overall balanced performance. Although the approach described in 2.4.4 is useful for this task, a limited set of tentative hyperparameter values must be established manually and this choice may be suboptimal. In the following chapters we will address this issue.



a) Irregular facade b) Sparse structure c) Hand-drawn

Figure 2.14: Results on non-standard facade images.

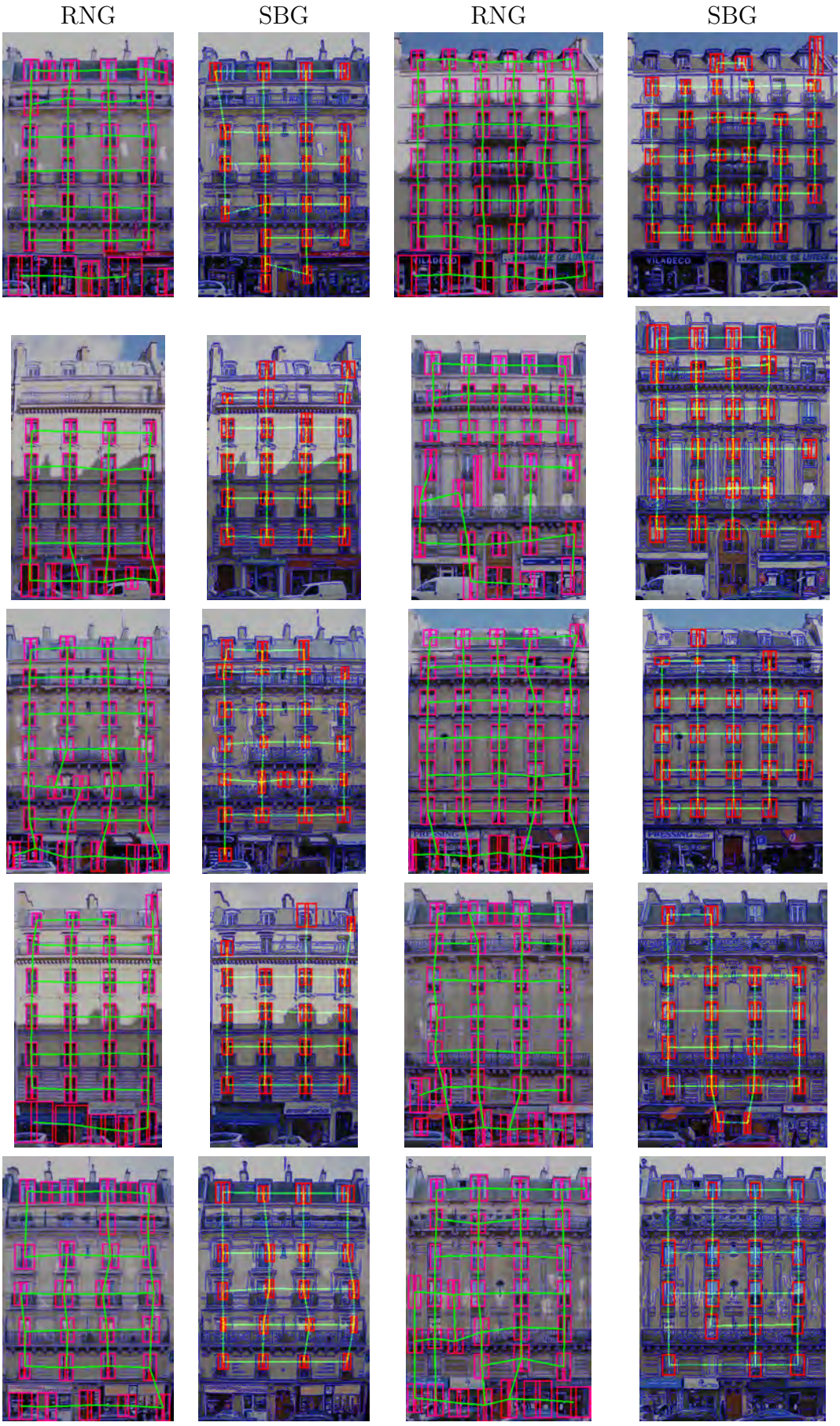


Figure 2.15: Results of the proposed method on the ten test images in the Parisian dataset with RNG and SBG structure priors.

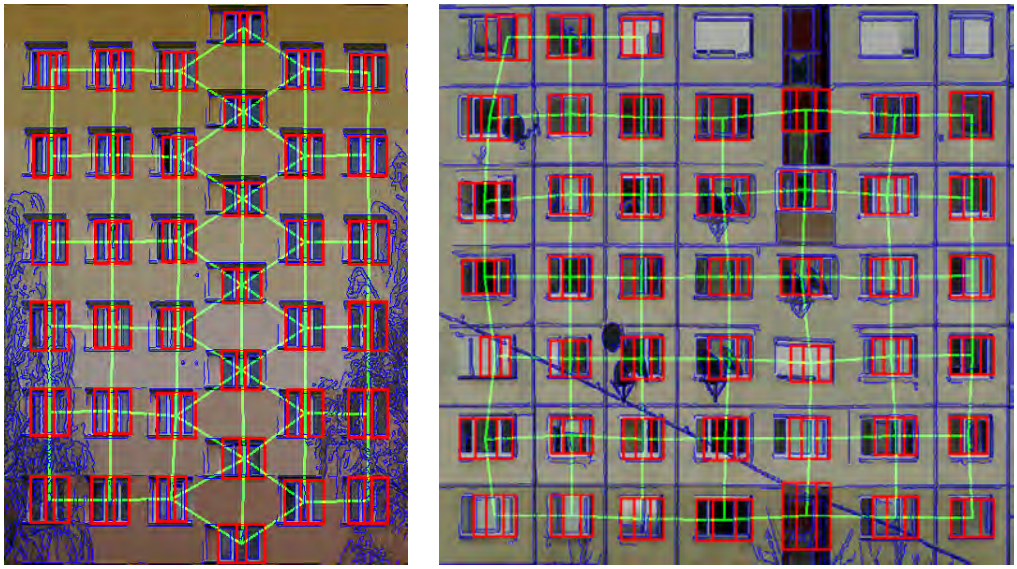


Figure 2.16: Interpreted facades of modern buildings.

Chapter 3

Spatial Pattern Templates

"The design of a temple depends on symmetry, the principles of which must be most carefully observed by the architect."

Marcus Vitruvius (80-25 BC)

3.1 Introduction

While previously some of hyperparameter values were assigned empirically, in this chapter a method which is able to learn the structure of relations between components will be proposed to get around the difficulties encountered in the previous approach (Sec. 2.8).

The recent development in the areas of object detection and image segmentation is centered around the incorporation of contextual cues. Published results confirm the hypothesis that modeling relations between neighboring pixels or segments (*superpixels*) can significantly improve recognition accuracy for structured data. The first choice one has to make here is to choose the neighbor relation, or in other words, which primitive elements participate in constraints on labels. The constraints are usually specified with a formal language of spatial arrangements. A common choice for the relation is the adjacency of element pairs in the image plane, such as 4 or 8-neighborhood of pixels in a grid, which supports the language model (Čech and Šára, 2009). This can be extended in various directions: In 'depth' when more concurrent segmentations[1] are overlaid to handle multiple scales, or in cardinality when we connect more elements together. Generally speaking, in this chapter we will take a closer look on this design process and introduce a concept called *Spatial Pattern Templates* (SPT).

A convenient framework to embed such patterns into are probabilistic graphical models, where image elements correspond to nodes and edges (or higher-order cliques) to the relations among them. In such a graph, our pattern templates correspond to cliques or factors, as they describe how a given joint probability factorizes. We choose CRF (Sec. 1.3.2.3, Lafferty et al. (2001)) as a suitable model, which allows us to concentrate on the element relations

[1] Segmentation is a partitioning of an image into segments (compact subsets of pixels in the image).

and not to care much about how the data are generated. Specifically, we propose pattern templates to deal with regular segmentations of translation-symmetric objects and call them *Aligned Pairs (AP)* and *Regular Triplets (RT)*.

We identify regular segmentations as those where object geometry, shape or appearance exhibit translation symmetry, which manifests in alignment and similarity. Such principles often apply to images with man-made objects, even though such phenomena are also common in the nature. Urban scenes have some of the most regular yet variable segmentations and their semantic analysis is receiving more attention nowadays, as it can aid other computer vision tasks such as image-based urban reconstruction. We design our method with this application in mind, specifically targeting parsing of facade images (a multi-class labeling problem).

In this task, we exploit the properties of largely orthogonal facade images. We start by training a classifier to recognize the patches given by unsupervised segmentation. Based on the initial segments we build a CRF with binary relative location potentials on AP and ternary label consistency potentials on RT. For intuition, this can be seen as a process where all segments jointly vote for terminal labels of the other segments, with voting scheme following the chosen spatial patterns. The concept of template design, its embedding in the CRF and implementation for regular objects with *Regular Triplets* and *Aligned Pairs* are the contributions of this chapter.

3.2 Related Work

3.2.1 Contextual Models

Relative location prior on label pairs is used in Gould et al. (2008) for multi-class segmentation. Every segment votes for the label of all other segments based on their relative location and classifier output. Ideally, such interactions should be modeled with a complete graph CRF, where an edge expresses the joint probability of the two labels given their relative location, but this would soon make the inference intractable with the growing number of segments. Instead Gould et al. (2008) resort to a voting scheme and use CRF with pairwise terms for directly adjacent segments only. In our approach, we include the discretized relative location prior in a CRF but limit the number of interactions by choosing a suitable pattern template.

CRFs are popular for high-level vision tasks also thanks to the number of algorithms available for inference and learning (Nowozin et al., 2010). However, useful exact algorithms are only known for a specific class of potential functions (obeying *submodularity*). Kohli et al. (2009) fit in this limitation with a robust version of a generalized Potts model, which softly enforces label consistency among any number of elements in a high order clique (pixels in segments). We can use this model for RT, but because the pairwise relative location potentials may have arbitrary form, we cannot apply the efficient α-expansion optimization

used in Kohli et al. (2009).

3.2.2 Structure Learning

A number of methods for learning general structures on graphs have been recently developed (Galleguillos et al., 2008; Schmidt et al., 2008; Schmidt and Murphy, 2010). They learn edge-specific weights in a fully connected graph, which is directly tractable only when the number of nodes n is small (10 segments and 4 spatial relations in Galleguillos et al. (2008)) due to edge number growing with $\mathcal{O}(n^2)$. Scalability of the approach has been extended by Schmidt et al. by block-wise regularization for sparsity (Schmidt et al., 2008) (16 segments) and subsequently also for higher-order potentials with a hierarchical constraint (Schmidt and Murphy, 2010) (30 segments). Since we deal with ≈ 500 segments, this approach cannot be directly applied and, as suggested in Schmidt et al. (2008), a restriction on the edge set has to be considered. The SPT can be here seen as a principled implementation of this restriction to keep the problem tractable.

3.2.3 Facade Parsing

In contrast to the state of the-art method (Sec. 1.2.5) by Martinovic et al. (2012) our method accommodates the general assumption of regularity in a principled and general way as a part of the model, which is based on the CRF and can benefit from the joint learning and inference.

3.3 Spatial Pattern Template Model

Initially we obtain a set of segments X in the input image with a generic method such as (Felzenszwalb and Huttenlocher, 2004), tuned to produce over-segmentation of the ground truth objects such as windows, wall, door etc. in Fig. 3.7b. The image parsing task is to assign labels $Z = (z_1, \ldots, z_n)$, $z_i \in C$, of given semantic classes $C = \{c_1, \ldots, c_k\}$ to given image segments $X = (x_1, \ldots, x_n)$, $x_i \subset \operatorname{dom} I$ in an image I. With segments corresponding to nodes in a graph and labels Z being the node variables, we construct a CRF with potentials taking the general form of

$$p(Z \mid \theta, X, \mathcal{Q}) \propto \prod_{q \in \mathcal{Q}} \exp\left(- \sum_{j \in \phi(q)} \theta_j \, p_j(\mathbf{z}_q \mid \mathbf{x}_q) \right), \tag{3.1}$$

where $\mathcal{Q}$ is the set of cliques, p_j are potential functions from a predefined set $\phi(q)$ defined for a clique q. The p_j is a function of all parameters z_i, x_i in the clique q joined together in vectors $\mathbf{z}_q, \mathbf{x}_q$ and the output is weighted by θ_j. The design of a specific CRF model now lies in the choice of cliques $\mathcal{Q}$ defining a topology on top of the segments, and the choice

of their potential functions p_j, which act on all node variables in the clique and set up the probabilistic model.

The analogy to the other models is suggested by the notation following Sec. 1.3.1. Primitives are segments X and their assignment to classes C is represented by labels Z. Classes can be seen as 'semantic components' opposed to spatial components. Both primitives X, classes C with their number k need to be fixed in (3.1) for inference and learning of (hyper)parameters θ_j. This is the downside of bypassing the problem of hyperparameter learning encountered in the previous Chapter 2.

3.3.1 Spatial Templates for Data-dependent Topology

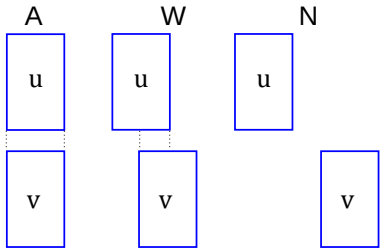
As a generalization of the *adjacency*, used i.e. in Yang and Förstner (2011), we can think of other choices for the graph topology that may suit our domain by including interactions between distant image elements, which are 'close' to each other in a different sense. As mentioned in Sec. 3.2, the scale of the problem does not allow us to reach complete connectivity. To allow dense connectivity while keeping the problem tractable, we need to restrict the number of cliques (edges). We describe this restriction with a *template* and, with the geometrical context in mind, we limit ourselves to *spatial* templates, which assign segments to cliques based on their geometrical attributes (shape, location). In principle other attributes (appearance) could be used in the template too. The meaning of this representation is to provide a systematic procedure for automatic learning of which interactions are the most efficient ones for the recognition task at hand.

In order to describe the process of designing a complex data dependent topology for a CRF, we first have to decompose the process behind clique template design into individual steps:

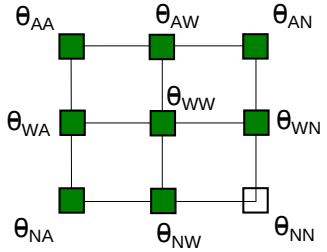
1. The first step is the specification of **core attribute relation functions** $\delta_i : A^n \to \mathbb{R}$ based on the domain knowledge. The relations act on easily measurable attributes A of n-tuples of segments.
   Example: Positions of two points in a plane as attributes $A_x, A_y \in \mathbb{R}^2$ and their signed distances in directions x and y as the relations δ_x, δ_y.

2. The ranges of relations δ_i are **discretized** to ordered sets Δ_i and $d_i : A^n \to \Delta_i$ becomes the discrete counterpart of function δ_i.
   Example: The signed distance is divided into three intervals, $\Delta_x = \{\text{left}, \text{equal}, \text{right}\}$, $\Delta_y = \{\text{below}, \text{equal}, \text{above}\}$.

3. In the next step the Cartesian product of m relation ranges Δ_i gives domain $D = \Delta_1 \times \cdots \times \Delta_m$, where subsets define logical **composite relations** (*and, or,* $=$).
   Example: Three intervals on two axes give 3^2 combinations in $D_{xy} = \Delta_x \times \Delta_y$, which can represent relations such as $((\Delta_x = \textit{left})\,\textit{and}\,(\Delta_y = \textit{below}))$, another example is $(\Delta_x = \Delta_y)$.



a) Discretized horizontal alignment relation Δ_a (**A**ligned, **W**eakly aligned, **N**ot aligned).



b) Template domain $D_{AP} = \Delta_a \times \Delta_a$, green indicates $\omega_{AP} = 1$.

Figure 3.1: Illustration of AP construction on a simplified case with only three possibilities in Δ_a = (A,W,N) for horizontal alignment and the same for vertical alignment (not shown). The weights θ associated with the template are subject to learning, except for the θ_{NN} (not aligned in any direction) for which $\omega_{AP}(N, N) = 0$, i.e. it is purposely excluded by the designer.

4. The **spatial template** is a subset $\Omega \subset D$ representing a concrete relation. The template is specified by an indicator function $\omega : D \to \{0, 1\}$ representing the allowed combinations.
   Example: For alignment in one direction we set $\omega_{xy} = 1$ *when* $d_x =$ equal *or* $d_y =$ equal, *otherwise* $\omega_{xy} = 0$.

The template design may be viewed as a kind of declarative programming framework for model design, a representation that can incorporate the specific knowledge in a generic way with combinations of core relations δ_i. Each spatial template is related to one potential function p_j in (3.1).

In summary, the result of this process describes which subsets of segments S labeled L should be jointly modeled in a graphical model; which of these are effective is subject to learning. Figure 3.3 shows how the segments correspond to nodes and their subsets define factors in $p(Z \mid X)$. In this work we introduce two templates suitable for regular segmentations.

3.3.1.1 Aligned Pairs (AP)

First template *Aligned Pairs* extend the basic adjacency relation by allowing also more distant connections between segments which are not directly adjacent. Out of all pairs of segments u, v we choose only those which are aligned either vertically or horizontally. It is useful to connect non-adjacent segments when the labels in such pairs follow some pattern, i.e. windows are aligned with some free wall space in between, sky is above roof, windows are inside wall etc.

The specification follows the spatial template design steps, a simplified illustration is provided in Fig. 3.1:

1. Based on the position attribute we choose horizontal and vertical **alignment** δ_h, δ_v with $\delta_h : (x_u, x_v) \to \mathbb{R}$ and $\delta_h = 0$ when the segments are exactly aligned, otherwise

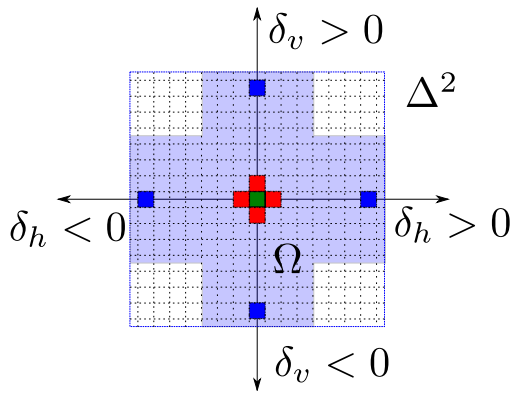


Figure 3.2: Spatial template Ω is a subspace in the domain D_{AP} given by relation functions δ_h, δ_v. The center corresponds to the exact alignment in both axes. If segment u (green) is located in the center, other squares (red for *adjacency*, blue belong to *Aligned Pairs*) correspond to discrete relative positions of segment v.

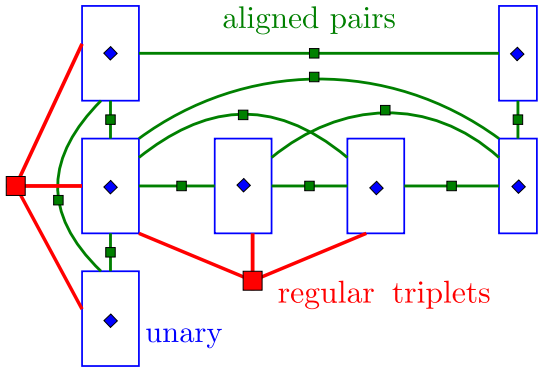


Figure 3.3: Factor graph for regular SPT. Segments X are shown as blue rectangles x_i (i.e. corresponding to *window* frames), factors are solid squares. *Aligned Pairs* connect only segments in mutual relative position specified by the template in Fig. 3.2. *Regular Triplets* then combine two aligned and equally spaced pairs together.

according to Fig. 3.2 (analogically δ_v for vertical).

2. Quantized functions $d_h, d_v : \mathbb{R}^4 \to \Delta_a$ evaluate locations of the two segment bounding boxes in both horizontal (d_h) and vertical (d_v) direction. The possible discrete values $\Delta_a \subset \mathbb{Z}$ for relative position and length of the two intervals are ordered according to Fig. 3.4, which is a pictorial representation of a set of inequality conditions, i.e. the identity $\Delta_a = 0$ is tested with $a = u \wedge b = v$, the left adjacency $\Delta_a = -6$ is tested with $u < a \wedge a = v$ and so on; a few more cases are described below. The values beyond ± 6 include the relative free space, i.e. on the right $\Delta_a = 6 + \lceil (u - b)/(b - a) \rceil$.

3. Combinations of horizontal and vertical alignment are then represented by joining d_h, d_v in a discrete domain $D_{AP} = \Delta_a \times \Delta_a$ limited by maximum distance.

4. Finally we specify the AP template with $\omega_{AP} = 1$ in the blue region in Fig. 3.2.

Note that *adjacency* (4-neighborhood) is a special case of AP when we specify $\omega_{AP} = 1$ only for four specific values in D_{AP} (directly above/under/left/right, red squares in Fig. 3.2, $|d_h| = 6 \wedge d_v = 0$ or $d_h = 0 \wedge |d_v| = 6$). Similarly values of $|d_h| \leq 5$ together with $|d_v| \leq 5$ correspond to *overlap* or *nesting* of segments.

3.3.1.2 Regular Triplets (RT)

In this template we combine two *Aligned Pairs* in a triplet u, v, w with regular spacing, in which the v is the shared segment. Including triplets allows to express a basis for repetitive structures (rows, columns) of primitive objects of the same label (*window*, *balcony*).

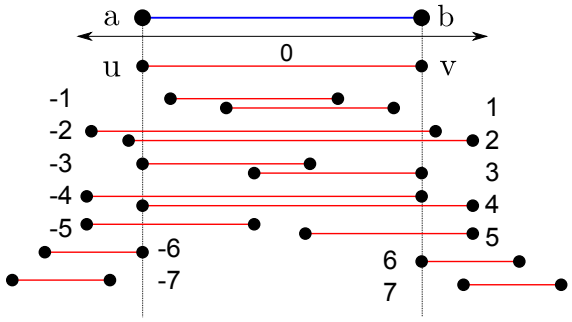


Figure 3.4: Given interval (a, b) the figure shows the values Δ_a of alignment relation function d_a for a set of intervals (u, v), ranging from 0 (aligned) to ± 7 (no overlap). More free space between intervals corresponds to higher absolute values $(8, 9, 10, \dots)$ in Δ. Positions are considered equal within 10% tolerance of the interval length.

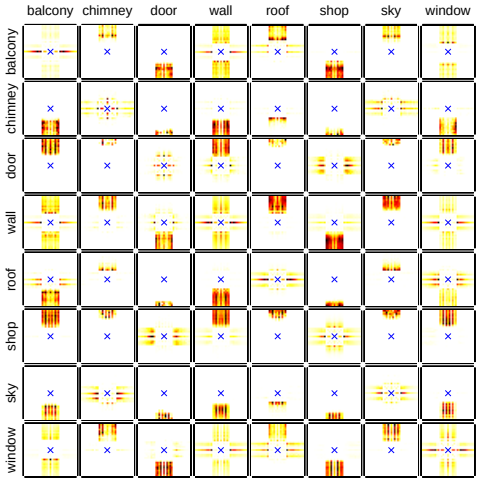


Figure 3.5: Discrete relative co-occurrence location histogram $p(z_u, z_v, d_h, d_v)$ for label pairs in the *ECP-Monge* dataset. It holds information such as 'sky is usually above windows' or 'balconies are aligned vertically with windows'. Dark colors correspond to high frequency, blue cross marks $d_h = d_v = 0$ (equality).

1. In addition to position alignment δ_h, δ_v defined for AP we introduce ternary relation functions for **size similarity** $\delta_s : (x_u, x_v, x_w) \to \mathbb{R}$ (relative difference in size of segments) and **regular spacing** $\delta_r : (x_u, x_v, x_w) \to \mathbb{R}$ (relative difference in free space between segments).

2. Based on them we define binary function $d_s : (x_u, x_v, x_w) \to \{0, 1\}$ to be 1 when $|\delta_s| < 0.1$ and similarly $d_r : (x_u, x_v, x_w) \to \{0, 1\}$ to be 1 when $|\delta_r| < 0.1$.

3. All functions $d_h(x_u, x_v)$, $d_v(x_u, x_v)$, $d_h(x_v, x_w)$, $d_v(x_v, x_w)$, $d_s(x_u, x_v, x_w)$ and $d_r(x_u, x_v, x_w)$ are then joined in a six-dimensional domain $D_{RT} = \Delta_a^4 \times \{0, 1\}^2$.

4. Finally we specify $\omega_{RT} = 1$ in the subspace of D_{RT} where $d_s = 1, d_r = 1$ and values of d_h, d_v indicate that the three segments are pair-wise aligned in the same direction (horizontal or vertical).

3.3.2 Probabilistic Model for Label Patterns

Given the fixed set of segments X, we will now make use of the SPT topology to model regular contextual information with a CRF for the graphical model.

For clarity we rewrite (3.1) in a convenient form

$$p(Z \mid X) \propto \prod_{u \in S} \exp(p_1(\nu_u)) \times \prod_{(u,v) \in AP} \exp(p_2(\nu_u, \nu_v)) \times \prod_{(u,v,w) \in RT} \exp(p_3(\nu_u, \nu_v, \nu_w)), \quad (3.2)$$

where $\nu_i = (z_i \mid x_i)$ are variables related to node i and p_1, p_2, p_3 are unary, pair-wise (AP) and ternary (RT) potential functions (factors) respectively. We will now discuss features used in these factors.

3.3.2.1 Unary Potentials

The $p_1(\nu_i) = \log p(z_i \mid x_i)$ are outputs of a multi-class classifier evaluated on the features for an image patch x_i of the segment x_i. The feature vector $f(x_i)$ is extracted from the image data by appending histogram of gradients (HoG), color (HSV), relative size, position, aspect ratio and 2D auto-correlation function.

3.3.2.2 Pairwise Potentials

Pairwise potentials for AP are restrictions on the template learned for concrete label pairs. They are based on a discretized version of the relative location distribution (Gould et al., 2008), similar form is used in (Tighe and Lazebnik, 2011) for *adjacency*. It is the statistical function

$$p_2(\nu_u, \nu_v) = w_{2,d_h,d_v} \ \log p(z_u, z_v \mid d_h, d_v), \quad (3.3)$$

where d_h are the values of horizontal alignment $d_h(x_{u1}, x_{v1})$ analogically d_v for vertical. As suggested in the specification of AP, they are computed by comparing the two segment locations: Their bounding boxes in the specified dimension (horizontal) are two intervals and a value d_h is assigned following Fig. 3.4 . The **pattern** of labels z_u, z_v is the empirical distribution in the given relative locations d_h, d_v computed as the second order co-occurrence statistics of the labels for pairs of segments observed in a training set. The co-occurrence frequencies are obtained from a training set for each pair of class labels and are accumulated for all values in the spatial template domain Ω_{AP}. Figure 3.5 shows the resulting histograms of AP in Fig. 3.2.

3.3.2.3 Ternary Potentials

Ternary potentials model regularity by encouraging some labels in RT to have the same value (i.e. *window*) in

$$p_3(\nu_u, \nu_v, \nu_w) = \begin{cases} w_{3,c} & \text{if } z_u = z_v = z_w = c, \\ w_{3,0} & \text{otherwise,} \end{cases} \quad (3.4)$$

which is a generalized Potts model (Kohli et al., 2009) and $w_{3,c}$ is a learned class-specific parameter. We do not use the complex ternary co-occurrence statistic with this potential

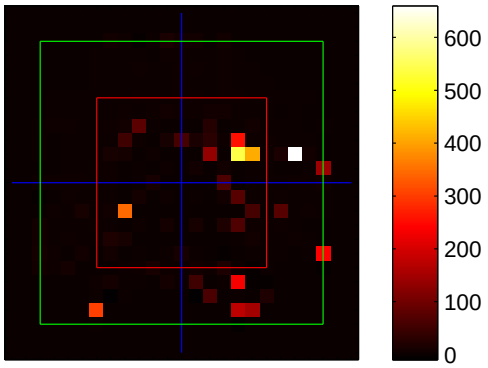


Figure 3.6: Result of AP parameter learning θ_2, green box covers the domain $\Omega_{AP} = d_h \times d_v$ depicted in Fig. 3.2. Large values (bright) correspond to important spatial relations while small values (dark) indicated relations which can be ignored when constructing a CRF for inference.

because there is not enough data for its training. To facilitate efficient learning, we convert ternary potentials into pairwise by adding a hidden variable for each ternary factor p_3.

3.3.3 Piece-wise Parameter Learning

The goal of parameter learning is to maximize (3.1) w.r.t. potential parameters (weights) θ.

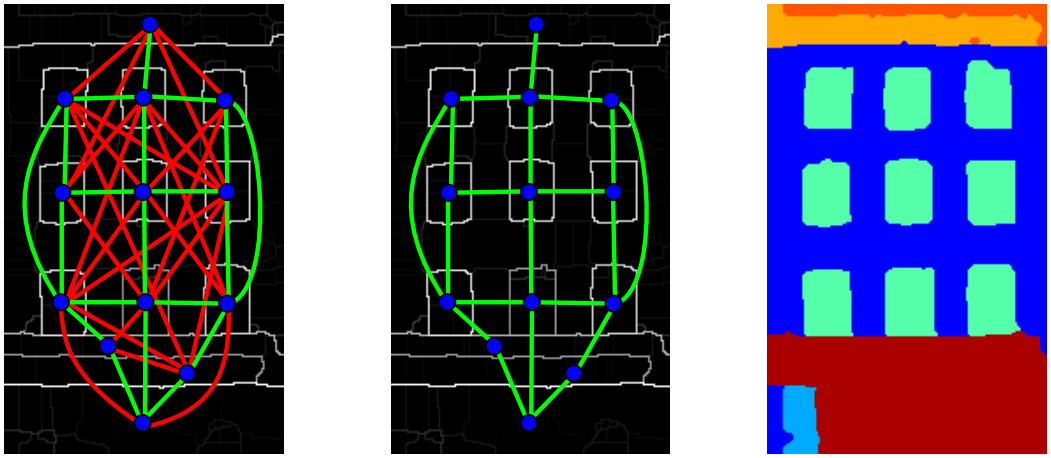
The unary potential classifiers are trained independently to reduce the number of free parameters in the joint CRF learning process. For binary potentials (including the reduced ternary potentials) we use pseudo-likelihood learning procedure to obtain values of the potential weights θ. This process corresponds to structure learning within the domain Ω_{AP} limited by the SPT topology, resulting in $\theta_2 \mapsto 0$ where the relation does not contribute to the discriminative power of the CRF (See Fig. 3.6). In practice this amounts to learning ~ 200 parameters based on likelihood in 50 sampled images, each of them with approximately 500 label variables, 3000 pair and 100 triplet factors. The training process takes several hours to complete (8 cores, 2 GHz) using Mark Schmidt's UGM library [2].

3.3.4 Inference

The overall inference process is illustrated in Fig. 3.7. The CRF is constructed only for important spatial relations, i.e. red edges shown in d) are not included. Because some of our potentials have a general form, exact CRF inference is not possible and we use an approximate algorithm (Kolmogorov, 2006) to compute the marginal distributions of the labels Z in (3.1). Segment labels are assigned to the most probable label to perform MAP estimation of (3.1). The run time around 30 s per image.

[2] www.di.ens.fr/~mschmidt/Software/UGM.html

a) input image → b) segmentation → c) bounding boxes



d) graph construction → e) CRF optimization → f) output labeling

Figure 3.7: Inference process. Only subset of pairwise relations is depicted in the graphs d) and e), ternary factors for rows and columns of windows are omitted for clarity.

3.4 Experimental Results

We have validated our method on two public datasets annotated into 8 classes (like *wall, window, balcony etc.*) and our large facade dataset (Tyleček, 2012).

The public *ECP-Monge* dataset is available from Simon et al. (2011) (we use corrected ground truth labellings from Martinovic et al. (2012)). It contains 104 rectified facade images from Paris, all in uniform Hausmannian style. Next, the public *eTrims* database (Korč and Förstner, 2009) contains 60 images of buildings and facades in various architectural styles (neoclassical, modern and other). We rectified them using vanishing points.

We have compiled a new publicly available larger *CMP Facade* database (Tyleček, 2012) with ~ 400 images of greater diversity of styles and 12 object classes. Its description can be found in Appendix A.

Figure 3.8 shows parsing results for different contextual models, additional results can be found on the dataset website Tyleček (2012). Table 3.1 provides their pixel-wise accuracy and comparison with other methods based on 5-fold cross validation. We have used method (Felzenszwalb and Huttenlocher, 2004) to extract averagely 500 segments (independently on the image resolution) and show it under *SGT*, where ground truth labels of pixels within each segment have been collected and the most frequent label among them selected for the entire segment. The result is the maximum achievable accuracy with this segmentation, inaccurate localization of the segment borders is currently the main limiting factor (we are 4.3% below the limit on *ECP-Monge*).

The main observation is that contextual information improves the accuracy averagely by 20% when statistics on AP is used, and by further 4% when RT are included. The RT help mostly with window and balcony identification, thanks to the statistics of these labels following regular pattern in the dataset. The qualitative improvement is noticeable, even when their effect on the total pixel-wise accuracy is small, which is a sign it is not a very suitable measure. A more sophisticated local classifier make the structural part of the model almost unnecessary, as observed in Martinovic et al. (2012), but such model may be overly reliant on a good training set and perhaps prone to overfitting.

3.5 Conclusion

We have introduced the concept of *Spatial Pattern Templates* for contextual models. The proposed *Aligned Pairs* and *Regular Triplets* templates have been found useful for segmentation of regular scenes by increasing accuracy of facade image parsing. Further we see possible improvement in the quality of the segment extraction to increase accuracy of segment borders.

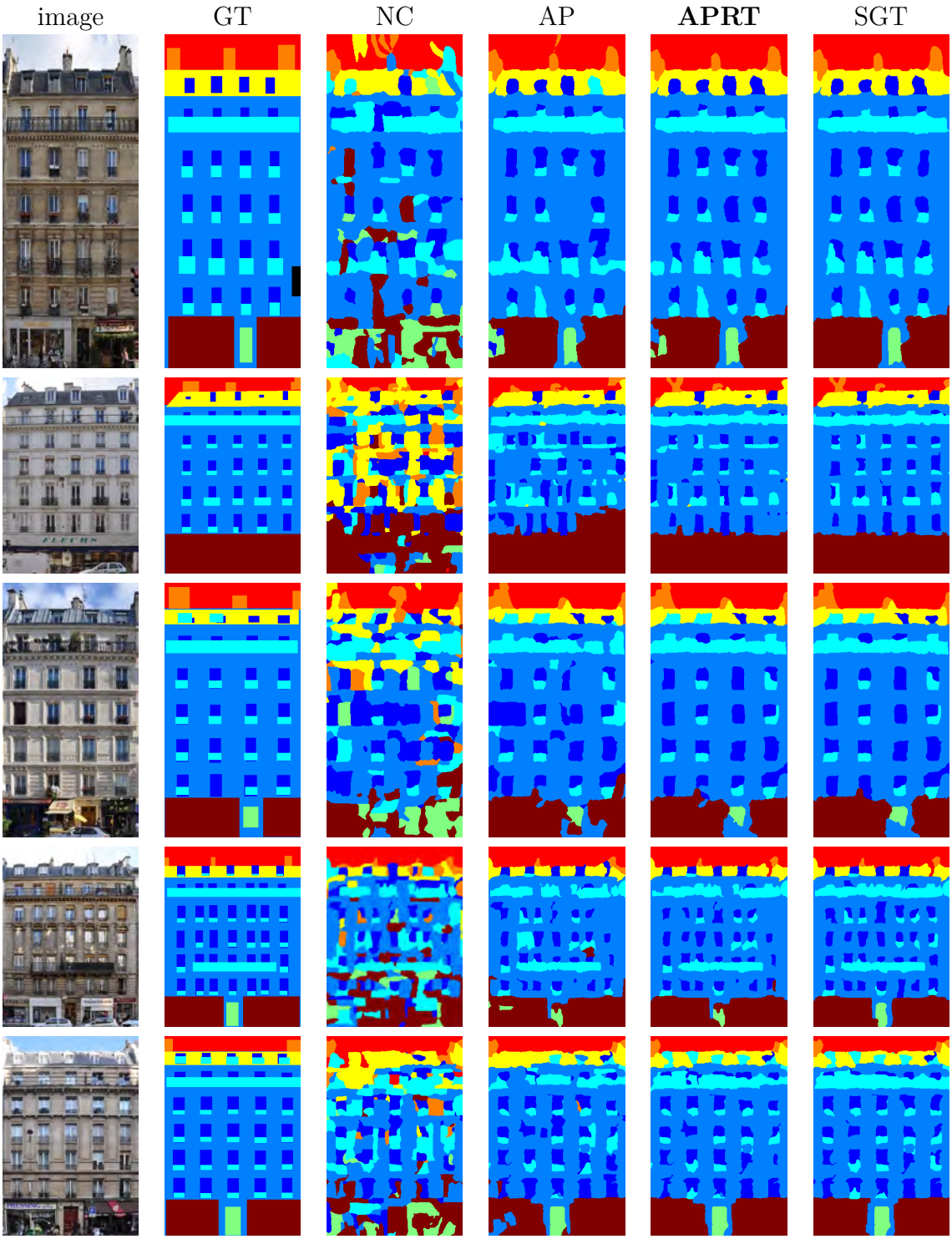


Figure 3.8: Selected visual results on ECP-Monge facade dataset, our result with full model is under *APRT*, (note it cannot be better than *SGT*). See legend in Tab. 3.1 for abbreviations.

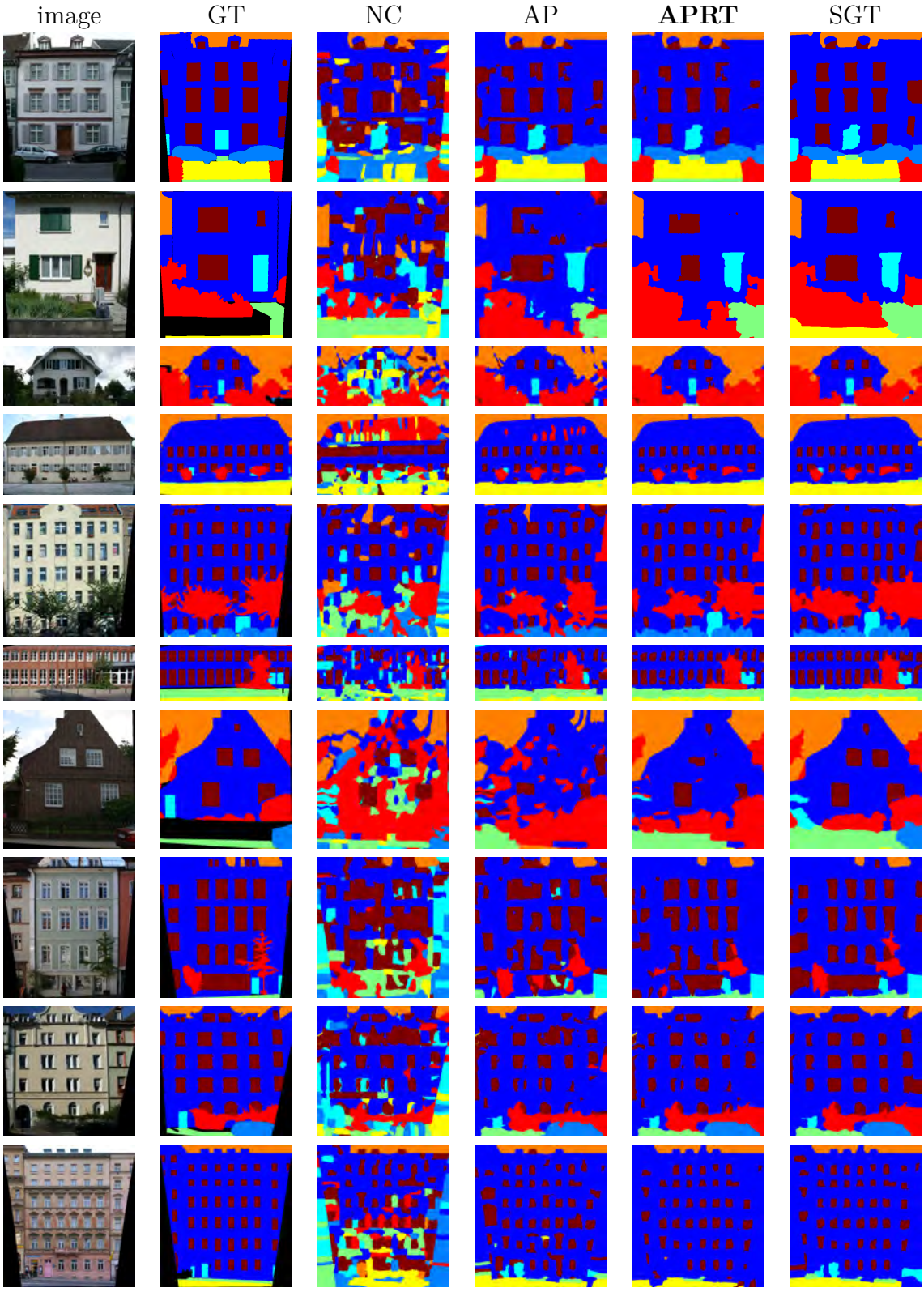


Figure 3.9: Selected visual results on eTrims DB facade dataset, our result with full model is under *APRT*. See legend in Tab. 3.1 for abbreviations.

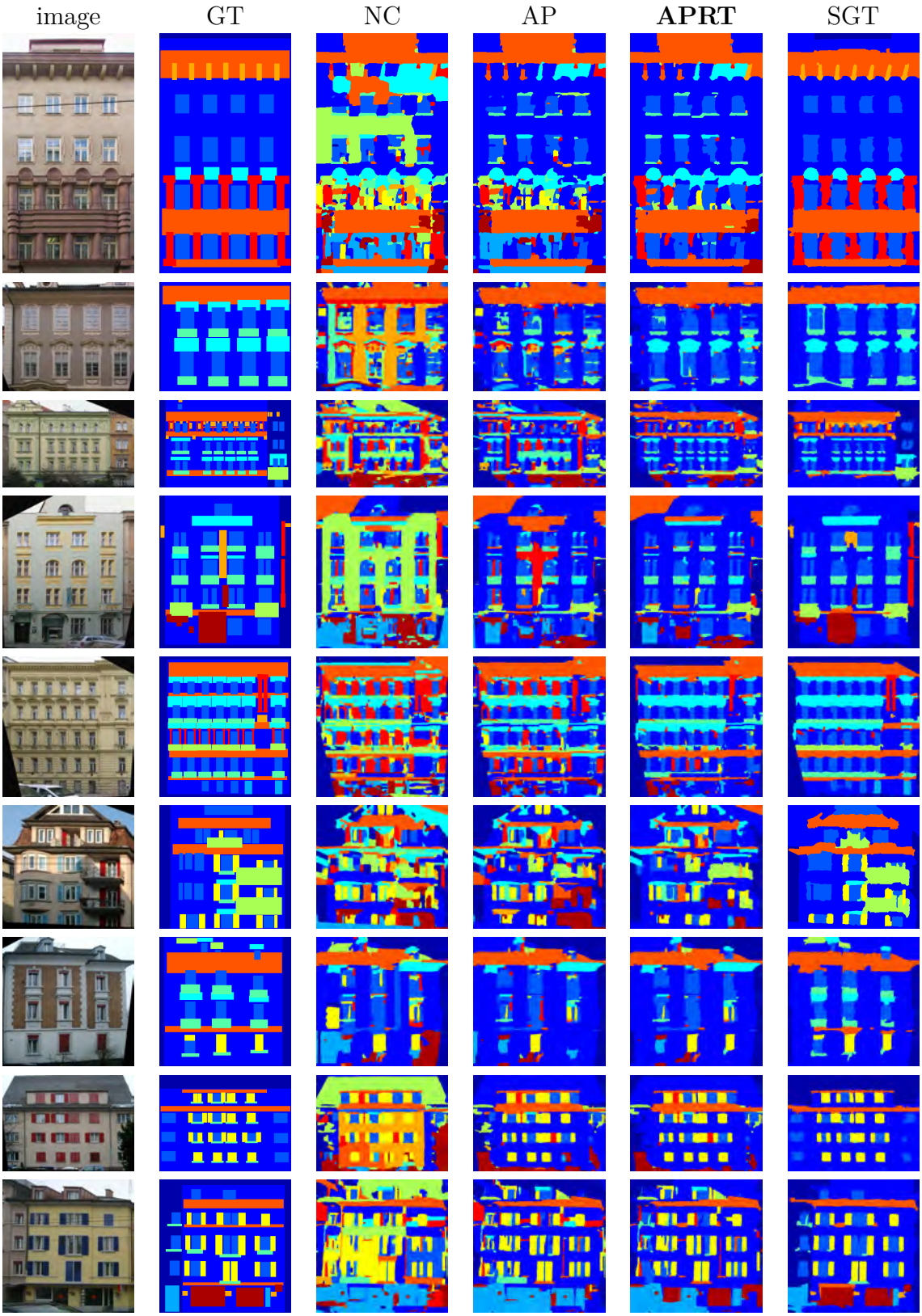


Figure 3.10: Selected visual results on CMP facade dataset, our result with full model is under *APRT*. See legend in Tab. 3.1 for abbreviations.

Method	SPT (proposed)				3L		SG	HCRF
Classifier	*SGT*	SVM			RNN		RF	RDF
Spatial pattern		NC	AP	**APRT**	NC	Adjacency	BSG	HAdj
Prob. model		-	Cooc	**Cooc**	-	Potts	SG	CS-Potts
ECP-Monge (8)	88.5	59.6	79.0	**84.2**	82.6	85.1	74.7	-
eTrims (8)	93.7	56.7	77.4	**82.1**	81.1	81.9	-	65.8
CMP Facade (12)	84.8	33.2	54.3	**60.3**	-	-	-	-

Abbreviations:

SGT Segments with Ground Truth labels,

NC No Context,

AP Aligned Pairs,

RT Regular Triplets,

Cooc Coocurence,

BSG Binary Split Grammar,

HAdj Hiearchical Adjacency,

SVM Support Vector Machine,

RNN Recursive Neural Network,

RF Randomized Forest,

SG Shape Grammar (Simon et al., 2011) ,

3L Three Layers (Martinovic et al., 2012),

HCRF Hierarchical CRF (Yang and Förstner, 2011).

Table 3.1: Pixel-wise accuracy comparison on facade datasets (number of classes in brackets).

Chapter 4

A Bayesian Model for Multiple Reflection Symmetry Detection

"Symmetry is what we see at a glance; based on the fact that there is no reason for any difference..."

Blaise Pascal (1669)

4.1 Introduction

Reflection symmetry[1] is a geometric property of a single object in an image, which typically cannot be further subdivided into distinguishable parts or components of the same kind, such as a car in Fig. 4.1a. We will say such an object is *integral*.

Exceptions from the *integrality* property are reflections of objects otherwise not necessarily symmetric (like in a mirror or water, Fig. 4.1b, which are composed of two separate parts (original, reflection).We will not address this case specifically.

Integrality makes the reflection detection problem substantially different from the translation symmetry addressed in the previous chapters. The elements repeating in translation make the identification of objects easier: multiple observations of the same objects (tens of them) justify the presence of the symmetry.

In the case of reflection symmetry we cannot rely on the regularity among the object parts and determination of the number of objects becomes more complicated. While there are typically only a few true reflection-symmetric objects in an image, we have to deal with the presence of multiple locally symmetric image patches (i.e. corners, stripes in texture) that are both geometrically and visually perfectly symmetric but not considered objects from the semantic point of view. This calls for a more elaborate method to determine the number of symmetric components rather than thresholding some symmetry measure, which is the common approach of the current state-of-the-art methods (Sec. 1.2.3).

[1] Some authors refer to this type of symmetry as *mirror* or *bilateral*.

a) A reflection-symmetric object. b) Arbitrary object and its reflection.

Figure 4.1: Different types of reflection symmetry (from Liu et al. (2013) dataset).

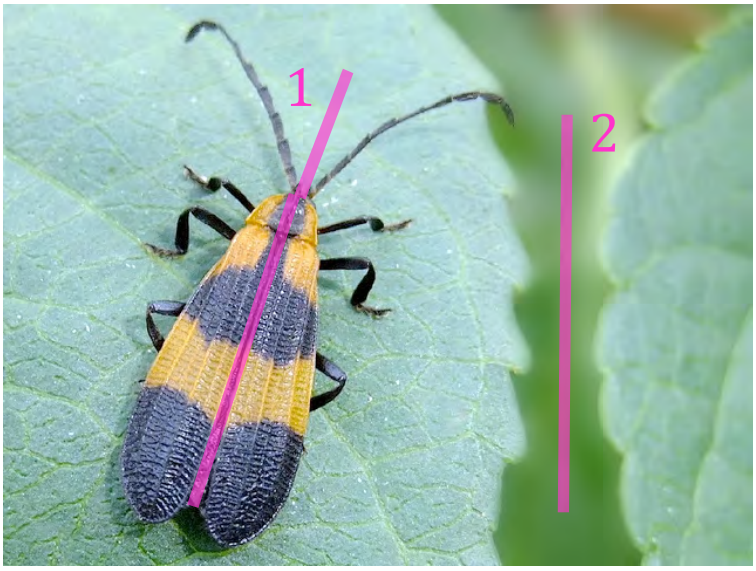


Figure 4.2: Object-background ambiguity in symmetry detection. Which symmetry is of interest? Only one symmetry (1) is annotated in the dataset Liu et al. (2013).

An example in Fig. 4.2 demonstrates ambiguity in symmetry detection. Both major axes of symmetry (the bug and the blank between the leaves) appear similar according to the geometrical quality of the reflection as well as appearance, but from the subjective view of an observer, it is just the bug (1) that is a symmetric object while the blank space (2) is considered background. Without some semantic information on the *objectness* of the symmetric entity it is difficult to distinguish between object and background, and it has been identified as one of the causes of false positive detections in the state-of-the-art methods.

4.1.1 Overview

We propose to employ Bayesian two-level inference (Sec. 1.3.3.4) to determine the number of symmetry instances (components) in a given image. The probabilistic model and the

inference method are as little specific to the application as possible, based on general principles. Unfortunately this does not imply modeling or algorithmic simplicity as judged by the formal modeling machinery required.

The underlying probabilistic model will allow to jointly evaluate properties of a component as a whole during the inference and even relations among the components, not only individual correspondences. A *correspondence* is a basic element in this model, it is linking two local image patches around interest points on either side of a hypothetical symmetry axis.

While considering all possible correspondences in an image is not computationally tractable, we filter them first using standard computer vision methods, i.e. keypoint extraction and image patch descriptor similarity. Specifically, we will pair salient keypoints by measuring their geometry and testing reflection symmetry of the descriptors in Sec. 4.2.

Then we start building our model around the filtered set of *tentative* correspondences that are the primitives of the probability model. Assuming an axis is given, we measure how much each individual correspondence matches it. The quality of the match is given primarily by geometry of the keypoint locations w.r.t. to axis parameters and we will derive the geometric term from a generative model. Additionally we add terms for a set of correspondence *features,* which are discriminative auxiliary functions taking into account keypoint appearance, orientation and scale.

The next level of our assumptions stemming from the integrality will be encoded in model priors for component parameters and shape, i.e. objectness and compactness (Sec. 4.6.1). The top level of our model will consider the component set as a whole; we see it is possible to describe the structure of the set by *grouping* components together. See Fig. 4.3 for breakdown of model elements and features, which will be detailed in the model description (Sec. 4.3).

With a few more general terms (complexity prior etc.) we can proceed with the stochastic inference. It is based on a random walk in the model parameter space, where the complexity and grouping is treated in a special way. We can obtain its marginal distribution (complexity posterior) by sampling configurations of components from the model. Given the maximum posterior complexity we can determine (look-up) the output model and its parameters (Sec. 4.10).

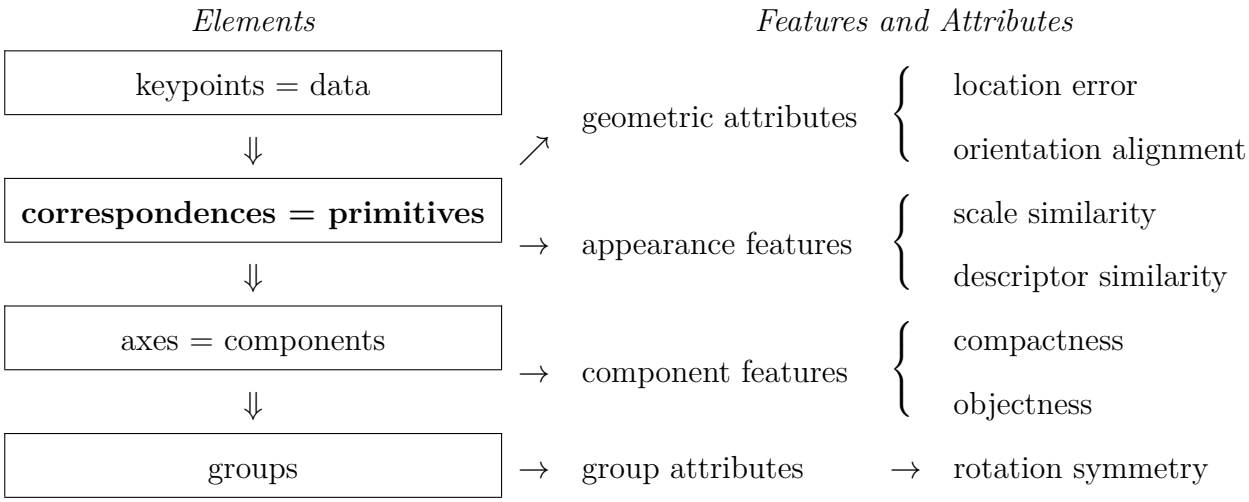


Figure 4.3: Hierarchy of model elements and features.

4.2 Image Features and Geometry

Similarly to other structure estimating methods in computer vision, we work with a set of keypoints which cover regions of interest. For our task of reflection symmetry detection we expect the keypoints to be invariant to reflection and obtain keypoints at similar locations on either side of the symmetry axis that correspond to each other, as in Fig. 4.5. This requires a keypoint detector sufficiently invariant to changes in local appearance of the object (particularly rotation, translation and reflection). To identify corresponding points we need also a descriptor which represents local appearance of an image patch around the keypoint with the same invariance properties as the detector. It will serve for the pre-selection of tentative correspondences and for computing descriptor similarity in the probabilistic model.

4.2.1 Keypoint Detector

The method for reflection symmetry detection (Loy and Eklundh, 2006), considered a baseline method by Rauschert et al. (2011) and described in Sec. 1.2.3, uses a traditional covariant feature detector of keypoints (DoG in Lowe (2004)), where the sparse detections often rely on a small number of corresponding points which can be missed when appearance varies from side to side of the axis. Furthermore, their sparsity does not allow accurate estimation of the center and extent of a symmetric patch.

More stable features are image edges (i.e. from detector by Canny (1986)) or 'contours' (gPb detector by Maire et al. (2008)), which have been used for symmetry detection by Wang et al. (2014). Matching of contour fragments however assumes good edge continuity, which is difficult to obtain when the object is not well separated from its background.

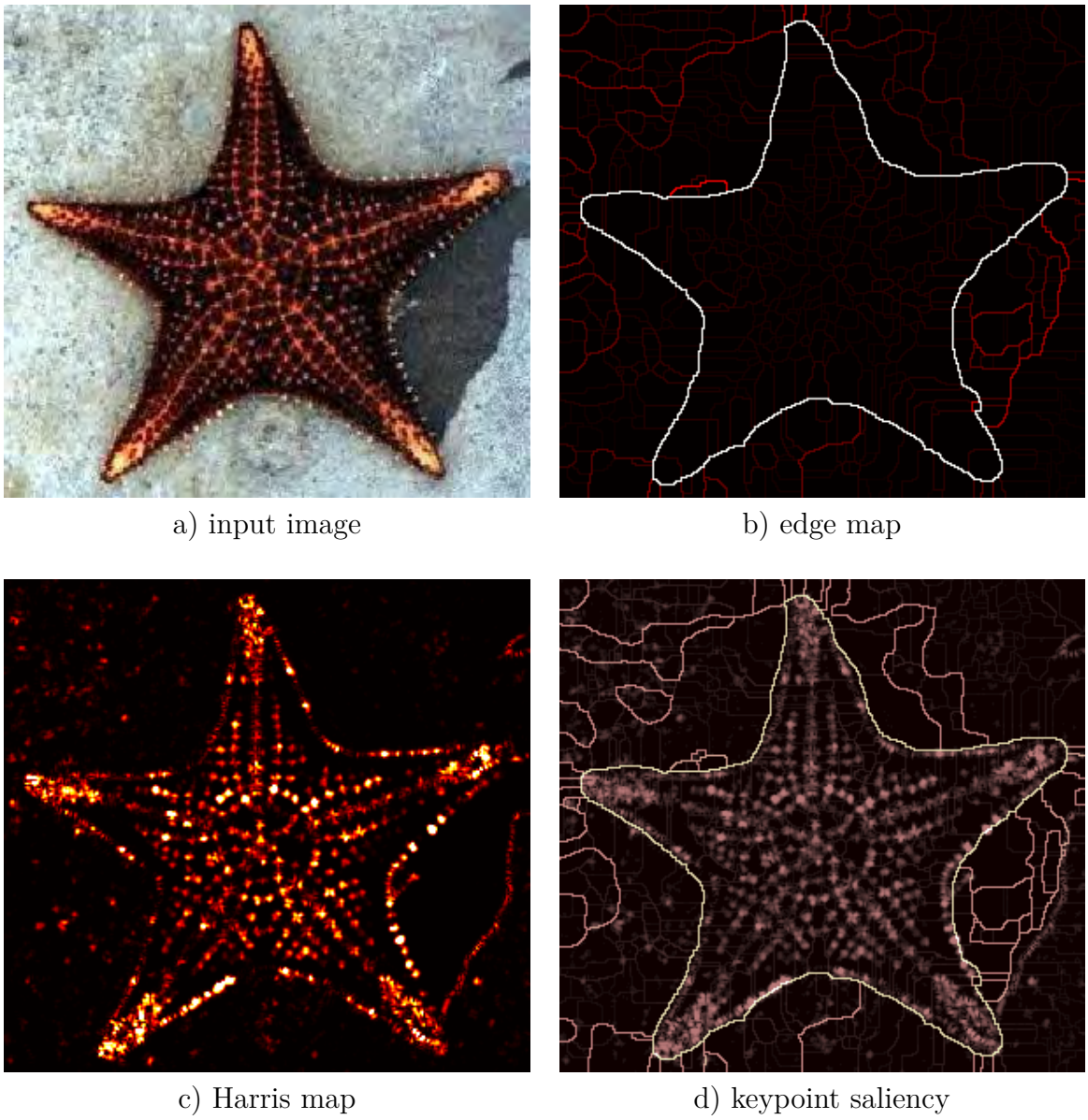
We propose to combine both approaches by augmenting densely sampled contours with covariant detections (Lowe, 2004) for better localization of the samples along the contour curves. Keypoints are sampled from a saliency map, which is a weighted sum of contours (Maire et al., 2008) and cornerness measure (Harris and Stephens, 1988b) shown in Fig. 4.4. In a given image I we select at most $n_k \approx 5000$ maximal points from such saliency map that exceed a given threshold and apply non-maximal suppression to enforce minimum distance between the keypoints. Each obtained keypoint has a set of attributes

$$(\mathbf{y}_i,\ s_i, \phi_i,\ \mathbf{d}_i)\,, \tag{4.1}$$

where we distinguish *geometric* attributes $\mathbf{y}_i$, ϕ_i and *appearance* attributes s_i, $\mathbf{d}_i$ specified as:

$\mathbf{y}_i \in (0,1)^2$ – location relative to image frame,

$\phi_i \in [0, 2\pi)$ – orientation (angle) in radians (image intensity gradient),



a) input image b) edge map

c) Harris map d) keypoint saliency

Figure 4.4: Saliency map for keypoint detection is constructed from edge and corner image features.

$s_i \in (0, 1]$ – scale relative to image frame (descriptor radius),

$\mathbf{d}_i \in [0, 1]^d$ – invariant image descriptor (vector of length d).

For the geometric location we will have a generative model while appearance attributes will be treated as discriminative features (see Sec. 1.3.1.6 for difference between attributes and features).

4.2.2 Reflection Geometry

Reflection symmetry is given by an axis defined with two vectors $(\mu, \mathbf{u})$ in

$$\mathbf{y} = \mu + \lambda \mathbf{u}, \; ||\mathbf{u}|| = 1, \tag{4.2}$$

where μ is a point on the axis, $\mathbf{u} = (\cos\varphi, \sin\varphi)$ is a directional vector corresponding to axis direction $\varphi \in [0, \pi)$ and $\lambda \in \mathbb{R}$ is a parameter (local coordinate of points on the axis). We further restrict the reflection to the strip delimited by two endpoints

$$\mu^1 = \mu + \lambda_h \mathbf{u}, \tag{4.3}$$

$$\mu^2 = \mu - \lambda_h \mathbf{u} \tag{4.4}$$

on an axis segment with half-length λ_h (see Fig. 4.5). Then

$$\mu = \frac{1}{2}(\mu^1 + \mu^2) \tag{4.5}$$

in (4.2) corresponds to their midpoint. The unit normal vector $\mathbf{v}$ is defined with $\mathbf{v} \perp \mathbf{u}$ and oriented counter-clockwise as in Fig. 4.5.

A single correspondence in the same sense as in RANSAC is a minimal sample of primitives for proposing an axis $\mathbf{y}_j(\lambda)$ perpendicular to the line connecting points $\mathbf{y}_{i_1}$, $\mathbf{y}_{i_2}$ running through their midpoint $\mu_j = \frac{1}{2}(\mathbf{y}_{i_1} + \mathbf{y}_{i_2})$

$$\mathbf{y}_j(\lambda) = \mu_j + \lambda \frac{(\mathbf{y}_{i_1} - \mathbf{y}_{i_2})^\top}{\|\mathbf{y}_{i_1} - \mathbf{y}_{i_2}\|} = \mu_j + \lambda \left(\cos\varphi_j, \sin\varphi_j\right), \tag{4.6}$$

where $\lambda \in \mathbb{R}$. This proposal will be later used to initialize the parameters $\bar{\theta}_j = (\mu_j, \varphi_j)$ of a new component in inference.

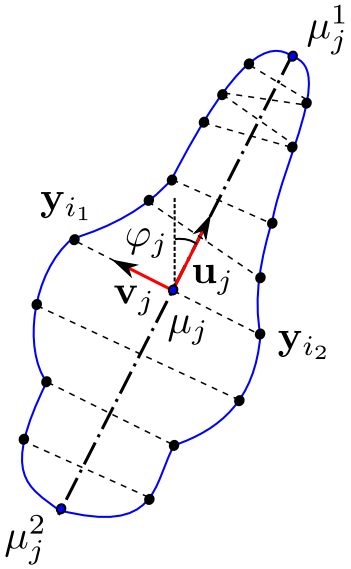


Figure 4.5: Sketch of a symmetric object (blue outline) with keypoints (black dots), correspondences (dotted lines) and a symmetry axis (dot and dash).

4.2.3 Descriptors

Each correspondence i comes with a descriptor similarity value comparing appearance of image patches around two keypoints i_1, i_2. Assuming translation invariance of descriptors we specify a descriptor similarity function $D_s : [0,1]^{d\times d} \to [0,1]$ specified for a given i by

$$d_i = D_s\Big(\mathbf{d}_{i_1}, \mathbf{d}_{i_2};\, \varphi_i\Big), \tag{4.7}$$

with respect to a reflection symmetry axis oriented by angle $\varphi_i \in [0,\pi)$ proposed by the correspondence itself (4.6). It has the form of

$$D_s(\mathbf{a}, \mathbf{b};\, \varphi) = \Big\| \mathbf{a} - f_r(\mathbf{b};\, \varphi)\Big\|, \tag{4.8}$$

where $\mathbf{a}, \mathbf{b} \in [0,1]^d$ are descriptor vectors and $f_r(\varphi) : [0,1]^d \to [0,1]^d$ rotates and mirrors a descriptor. Value $D_s = 0$ corresponds to exact reflection symmetry. We implement this function using a steerable circular descriptor Daisy (Tola et al., 2010) similar to R-HOG (Dalal and Triggs, 2005). Daisy has a slight advantage, its similarity evaluation follows directly the proposed reflection geometry via parameter φ[2]. It does not rely on implicit orientations φ_i locally estimated from descriptor such as in SIFT (Lowe, 2004)), then f_r only mirrors as descriptors are rotated implicitly. In both cases we make use of a descriptor mirroring function, which prevents us from the need to extract the underlying image patch, mirror it and compute a new descriptor.

The reflection similarity measured in d_i compares regions around the two keypoints, but the appearance of the central region between them can still be arbitrary. To verify the symmetry of the central region with the assumption of integrality we can evaluate a large scale descriptor for the midpoint, with orientation given by the correspondence and region size given by the distance between keypoints, see Fig. 4.6. This is in spirit similar to validation step in (Patraucean et al., 2013), where candidates produced by the baseline method (Loy and Eklundh, 2006) are filtered to reduce false positives. The candidates are validated by rotating the image according to the axis and calculating symmetry error of image gradient orientations densely in all pixels; candidates with high error are discarded. Our approach can be seen as a sparse approximation of the gradient symmetry error more efficiently calculated using descriptors.

The reflection self-similarity of this descriptor $\mathbf{m}_i$ evaluated for a correspondence $i = (i_1, i_2)$ is denoted

$$m_i = D_s\Big(\mathbf{m}_i, \mathbf{m}_i;\, \varphi_i\Big). \tag{4.9}$$

In practice we calculate m only for selected tentative correspondences (Sec. 4.2.4) rather than all keypoint pairs due to the computational cost of $\mathcal{O}(n^2)$.

[2] Because of discretization of the descriptor w.r.t angle in practice an arbitrary rotation is approximated by linear interpolation.

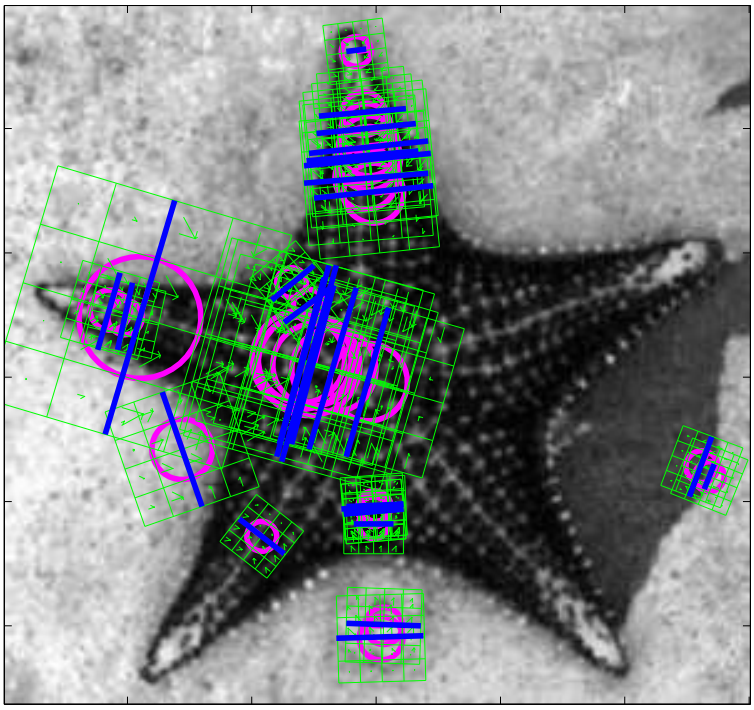


Figure 4.6: Large scale descriptors (green frame with bins) around a correspondence midpoint (center of the cyan circle) is used to evaluate the symmetry of the image region between the keypoints (connected by the blue line segment).

4.2.4 Primitive Elements

The *primitive element*[3] in reflection symmetry is a correspondence x_i between two keypoints i_1 and i_2 with locations $\mathbf{y}_{i_1}$, $\mathbf{y}_{i_2}$ in a plane (Fig. 4.5). Let

$$x_i = (\underbrace{\mathbf{y}_{i_1},\, s_{i_1}, \phi_{i_1},\, \mathbf{y}_{i_2},\, s_{i_2}, \phi_{i_2},}_{\text{attributes}}\ \underbrace{d_i,\, m_i}_{\text{features}}\ ) \tag{4.10}$$

be a concatenation of the two keypoint variables. As mentioned in Sec. 1.3.1.5, attributes are directly part of the data representation X while features are additional functions taking into account the original image I.

In order to reduce the number of primitive elements entering inference, we pick up only the prospective pairs from the set of all keypoint pairs $\mathcal{X}$. The measure used for this purpose is the probability density of correspondence attributes (scale, orientation, descriptors) given the axis parameters proposed by the correspondence itself; it will be described in Sec. 4.4.2 as a part of the probabilistic model.

The most effective strategy found is a variant of non-maximum suppression, which can be described as a greedy selection of the best correspondences. In the greedy scheme correspondences *close* to the currently best correspondence are removed together from the set along with the best one. The closeness is characterized as follows: Each two keypoints generate a line (different correspondences can generate the same line). Let us consider a

[3] A minimal data structure participating in the inference; see Sec. 1.3.1.1.

space of lines parameterized with polar coordinates (ϱ, φ), where $\varrho \in [0, \sqrt{2}]$ is the distance of the line from the origin in the unit image frame and $\varphi \in [0, 2\pi)$ the line orientation. The distance between the two lines i and j is then measured by

$$\delta_{ij} = \left\| \frac{1}{\sqrt{2}}(\varrho_i - \varrho_j),\, \sin(\varphi_i - \varphi_j) \right\|. \tag{4.11}$$

Finally correspondences with the distance $\delta_{ij} \leq \delta_0$ under a given threshold are considered close to each other.

With this strategy we can allow multiple distinct axes to share a keypoint while there is a low chance of a miss when we put a fixed limit on the maximum number of selected *tentative correspondences*. The tentative correspondence set will be denoted $X = \{x_1, \ldots, x_n\} \subset \mathcal{X}$.

4.3 Probabilistic Model

The parameters of this model follow the general definition from Introduction (Sec. 1.3.1.5), where different 'flavors' of parameters θ are distinguished and we give here only their brief review. The complexity k gives the number of components for which there are common configuration $\dot{\theta}$ and shape parameters $\hat{\theta}$. Grouping parameters $\breve{\theta}$ are associated with $\breve{k}$ groups. Components with parameters $\bar{\theta}$ are allocated to groups in the grouping field $\breve{Z}$ and finally primitive data elements X are allocated to components in configuration field Z. The hyperparameters denoted by ξ are fixed during inference and we also distinguish their flavors $\bar{\xi}, \hat{\xi}, \breve{\xi}$.

The two-level inference method (Šára, 2014) assumes a probability model structured in a way similar to Richardson and Green (1997) as

$$p(X, Z, \bar{\theta}, \breve{Z}, \breve{\theta}, \breve{k}, \hat{\theta}, \dot{\theta}, k) = p(X, Z \mid \bar{\theta}, \hat{\theta}, \dot{\theta}, k)\, p(\bar{\theta}, \breve{Z} \mid \breve{\theta}, \breve{k}, \hat{\theta}, k)\, p(\breve{\theta}, \breve{k} \mid k)\, p(\hat{\theta})\, p(\dot{\theta} \mid k)\, p(k)\,, \tag{4.12}$$

with the following terms (from left to right) and their function w.r.t. to reflection symmetry detection:

Data clustering model $p(X, Z \mid \bar{\theta}, \hat{\theta}, \dot{\theta}, k)$, which describes the reflection geometry and assignment of correspondences to components based on keypoint matching,

Component model $p(\bar{\theta}, \breve{Z} \mid \breve{\theta}, \breve{k}, \hat{\theta}, k)$, which describes component properties (integrality) and their relations (hierarchy, grouping),

Group prior $p(\breve{\theta}, \breve{k} \mid k)$, which describes group parameters $\breve{\theta}$, i.e. the center of a component cluster, and the number of groups $\breve{k}$,

Shape prior $p(\hat{\theta})$, which describes shape parameters $\hat{\theta}$ common to all components, i.e. the typical size of a component (variance of correspondences w.r.t. symmetry axis),

Data clustering prior $p(\dot{\theta} \mid k)$, which is a regularizer preferring components of similar inlier count,

Complexity prior $p(k)$, which represents a weak constraint on the number of components.

The following sections will either describe these terms specific to symmetry detection or keep their default form as specified in Šára (2014). For the sake of readability we omit hyperparameters ξ from the high-level model description and present them only in the detailed specification of individual terms. The terms are summarized in Tab. 4.1 and the model parameter structure can be found in Fig. 4.7.

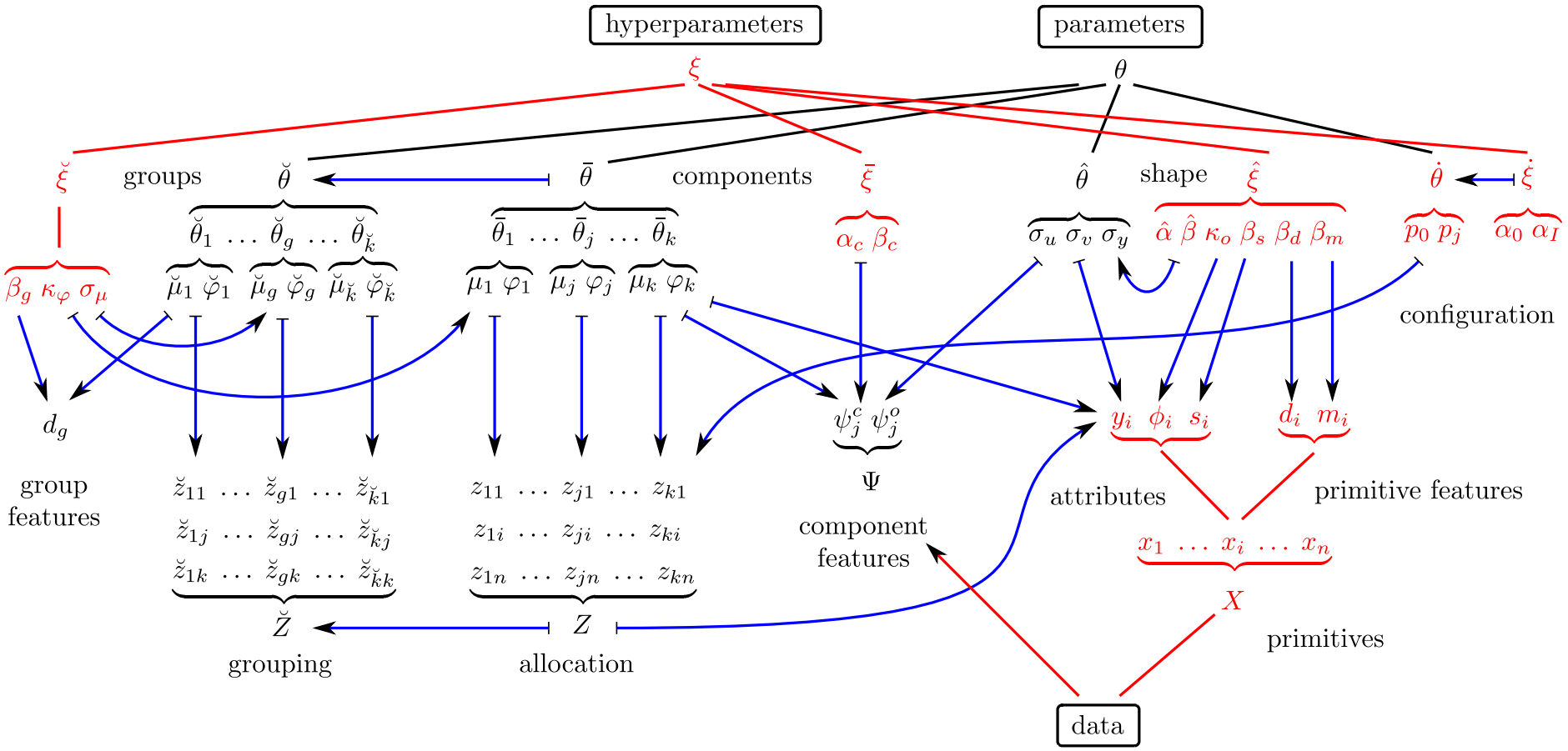


Figure 4.7: Breakdown of model parameters and their dependencies (blue arrows $x \to y$ when $p(y \mid x)$). Some dependencies have been joined together for clarity (T-shaped arrows $\mathbf{x} \mapsto \mathbf{y}$). Parameters fixed during inference are in red.

Term	*Param.*	*Description*	*Pdf*	*Hyperparameters*	*Eq.*
$p(k)$	k	complexity prior	B	p_k	(4.71)
$p(\breve{k} \mid k)$	$\breve{k}$	group complexity	B	$p_{\breve{k}}$	(4.73)
$p(\hat{\theta})$	$\hat{\theta}$	shape prior	IG	$\alpha_u, \beta_u,\ \alpha_v, \beta_v,\ \alpha_y, \beta_y$	(4.50)
$p(\breve{\theta} \mid \breve{k})$	$\breve{\theta}$	group prior	Be	β_g	(4.68)
$p(\dot{\theta} \mid k)$	$\dot{\theta}$	configuration prior	Dir	α_0, α_I	(4.70)
$p(y_i \mid \bar{\theta}_j, \hat{\theta})$	$\bar{\theta}$	data geometry	$\mathcal{N}$	$\sigma_u, \sigma_v, \sigma_y$	(4.22)
$p_0(y_i)$	$\bar{\theta}$	universal data	$\mathcal{N}$	σ_0	(4.46)
$p(\phi_i \mid \varphi_j)$	$\bar{\theta}$	orientation sym.	$\mathcal{N}_c$	κ_o	(4.28)
$p(s_i)$	$\bar{\theta}$	scale prior	Be	β_s	(4.37)
$p(d_i, m_i)$	$\bar{\theta}$	descriptor symmetry	Be	$\beta_d,\ \beta_m$	(4.39)
$p(\psi_j^c \mid \bar{\theta}_j, \hat{\theta})$	$\bar{\theta}$	compactness	Be-B	$\alpha_c,\ \beta_c$	(4.57)
$p(\psi_j^o \mid \bar{\theta}_j, \hat{\theta})$	$\bar{\theta}$	objectness	-	-	(4.58)
$p(\bar{\theta}_j \mid \breve{Z}, \breve{\theta}, \hat{\theta})$	$\bar{\theta}$	rotation group	Be, N	$\sigma_\mu, \kappa_\varphi$	(4.60)

Table 4.1: List of model parameters and their distributions. Hyperparameters fixed during inference are in red.

4.4 Data Clustering Model

Following Sec. 1.3.1.3 we assume each individual primitive $i = 1, \dots n$ has parameters x_i and a component allocation vector (Dirac distribution) $z_i = (z_{ji})$, $j = 0, 1, \dots, k$, where $z_{ji} \in \{0, 1\}$ are binary allocation variables ($z_{ji} = 1$ when primitive i is assigned to component j). Primitives assigned to the background component $j = 0$ are *outliers* while primitives assigned to some other components $j = 1, \dots, k$ are *inliers*. The partitioning of the set of primitives X into $k + 1$ sets Z_j, $j = 0, \dots, k$ will be called a *configuration* $Z : \bigcup_{j=0}^{k} Z_j$.

Probability density of observing a data instance X allocated to components by a binary *configuration field* $Z = \{z_i;\, i = 1, \dots, n\}$ is then given by the joint distribution

$$p(X, Z \mid \bar{\theta}, \hat{\theta}, \dot{\theta}, k) = \prod_{i=1}^{n} \sum_{j=0}^{k} z_{ji}\, p_j\, p(x_i \mid \bar{\theta}_j, \hat{\theta}) \tag{4.13}$$

where $p(x_i \mid \bar{\theta}_j, \hat{\theta})$ is the correspondence data term. The parameter p_j controls component membership and it is defined by

$$\dot{\theta} = (p_0, p_1, \dots, p_k), \quad p_j > 0, \qquad \sum_{j=0}^{k} p_j = 1. \tag{4.14}$$

More details on this construction can be found in Šára (2014).

The data term of matching a correspondence x_i (keypoint pair) with a given axis $\bar{\theta}_j = (\mu_j, \varphi_j)$ is then calculated as

$$p(x_i \mid \bar{\theta}_j, \hat{\theta}) = p(y_i, \phi_i \mid \bar{\theta}_j, \hat{\theta})\, p(s_i, d_i, m_i \mid \bar{\theta}_j, \hat{\theta}), \tag{4.15}$$

where *geometric symmetry* $p(y_i, \phi_i \mid \cdot)$ evaluates how locations of corresponding keypoints $y_i = (\mathbf{y}_{i_1}, \mathbf{y}_{i_2})$ and orientations $\phi_i = (\phi_{i_1}, \phi_{i_2})$ match a given axis and $p(s_i, d_i \mid \cdot)$ is *appearance symmetry*, where $s_i = (s_{i_1}, s_{i_2})$ are descriptor scales and d_i, m_i are descriptor symmetry features.

We will parameterize the per-primitive data model 4.15 in a way that is suitable for an efficient implementation. Let us write it as a scaled exponential-family distribution, which means it can be written as

$$p(x_i \mid \bar{\theta}_j, \hat{\theta}) = \exp\left[\, \sum_{w=1}^{W} \eta_j^w(\bar{\theta}_j, \hat{\theta})\, T_i^w(x_i) \,\right], \tag{4.16}$$

where η_j^w are *natural parameters* and T_i^w are *sufficient statistics* of the exponential-class model[4]. For simplicity of exposition we assumed that the components are homogeneous (η_j^w, T_i^w, are the same functional forms for all components). The parameters $\bar{\theta}_j, \hat{\theta}$ are subject to inference, whereas statistics $T_i^w(x_i)$ are fixed for a given problem instance, hence they can be

[4]This homogeneous form is somewhat non-standard by including the partition function in the parameter set. Nevertheless, we use the terms 'natural parameters' and 'sufficient statistics', even if this is not precise.

precomputed.

We proceed with geometric symmetry $p(y_i, \phi_i \mid \bar{\theta}_j, \hat{\theta})$ and appearance $p(s_i, d_i \mid \bar{\theta}_j, \hat{\theta})$ will be detailed later (Sec. 4.4.2).

4.4.1 Geometric Symmetry

We replace the geometric symmetry measure chosen mostly arbitrarily in existing works by a derivation of a distribution for the symmetry in data from a generative model for reflection symmetry. We proceed in a generic way and from the first principles, starting with

$$p(y_i, \phi_i \mid \bar{\theta}_j, \hat{\theta}) = p(y_i \mid \bar{\theta}_j, \hat{\theta})\, p(\phi_i \mid \varphi_j), \tag{4.17}$$

where the locations y_i and orientations ϕ_i are independent.

4.4.1.1 Location Symmetry

Let us assume the corresponding keypoints located at $\mathbf{y}_{i_1}, \mathbf{y}_{i_2}$ are noisy observations of an underlying perfectly symmetric keypoint pair with unknown positions $\mathbf{y}, \mathbf{y}'$ constrained by a given axis $\bar{\theta}_j = (\mu_j, \varphi_j)$ in

$$\mathbf{y}' = f_r(\mathbf{y};\, \mu_j, \varphi_j) = \mu_j + \mathbf{R}_j(\mathbf{y} - \mu_j), \tag{4.18}$$

where $\mathbf{R}_j = \mathbf{I} - 2\mathbf{u}_j\mathbf{u}_j^\top$ is a Householder reflection matrix and we will use the prime symbol as a shortcut for this reflection function f_r in the following text. The distribution of the perfect pair's location is

$$p(\mathbf{y}, \mathbf{y}' \mid \mu_j, \varphi_j) = \mathcal{N}\left(\mathbf{F}_j(\mathbf{y} - \mu_j);\, \mathbf{0}, \Sigma_j\right), \tag{4.19}$$

where $\Sigma_j = \mathrm{diag}(\sigma_{\hat{u}}^2,\, \sigma_{\hat{v}}^2)$ and $\mathbf{F}_j = [\mathbf{u}_j\, \mathbf{v}_j]$ is a rotation matrix composed of vectors $\mathbf{u}_j, \mathbf{v}_j$. The $\mathbf{y}'$ is not a free variable as it is fully determined by $\mathbf{y}$ given μ_j, φ_j.

The deviation of observed $(\mathbf{y}_{i_1}, \mathbf{y}_{i_2})$ from predicted $(\mathbf{y}, \mathbf{y}')$ is described by $p(\mathbf{y}_{i_1}, \mathbf{y}_{i_2} \mid \mathbf{y}, \mathbf{y}')$, where the noise term for a single observed $\mathbf{y}_{i_1}$ given a perfect location $\mathbf{y}$ is

$$p(\mathbf{y}_{i_1} \mid \mathbf{y}) = \mathcal{N}(\mathbf{y}_{i_1};\, \mathbf{y}, \sigma_y^2), \tag{4.20}$$

Because we don't know which of the $\mathbf{y}_{i_1}, \mathbf{y}_{i_2}$ 'belongs' to the observed $\mathbf{y}_1$ or $\mathbf{y}_2$, we marginalize this model over both possibilities in

$$p(y_i \mid \bar{\theta}_j, \hat{\theta}) = \int \left(p(\mathbf{y}_{i_1} \mid \mathbf{y})\, p(\mathbf{y}_{i_1} \mid \mathbf{y}') + p(\mathbf{y}_{i_1} \mid \mathbf{y}')\, p(\mathbf{y}_{i_1} \mid \mathbf{y})\right)\, p(\mathbf{y}, \mathbf{y}' \mid \mu_j, \varphi_j)\, \mathrm{d}\mathbf{y}. \tag{4.21}$$

The resulting geometric data term is a multinomial normal distribution in transformed coordinates and the pdf is specified as

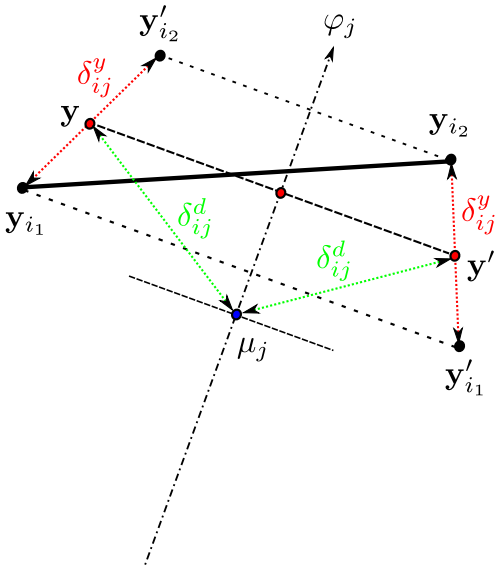


Figure 4.8: Geometry for the measure of reflection error (red) and distance (green).

$$
\begin{aligned}
p(y_i \mid \bar{\theta}_j, \hat{\theta}) &= p(\mathbf{y}_{i_1}, \mathbf{y}_{i_2} \mid \mu_j, \varphi_j, \sigma_u, \sigma_v, \sigma_y) = \mathcal{N}\left(\delta_{ij}(y_i, \bar{\theta}_j);\, \mathbf{0}, \hat{\Sigma}\right), \qquad (4.22)\\
\delta_{ij}(y_i, \bar{\theta}_j) &= \Big[\underbrace{\frac{\mathbf{y}_{i_1} - \mu_j - \mathbf{R}_j(\mathbf{y}_{i_2} - \mu_j)}{2}}_{\text{reflection error } \delta^y_{ij}}, \underbrace{\frac{\mathbf{y}_{i_1} + \mu_j + \mathbf{R}_j(\mathbf{y}_{i_2} - \mu_j)}{2}}_{\text{distance } \delta^d_{ij}} \Big],
\end{aligned}
$$

where $\hat{\Sigma} = \mathrm{diag}(\sigma_y^2, \sigma_y^2, \sigma_u^2, \sigma_v^2)$ and $\sigma_u^2 = \sigma_y^2 + 2\sigma_{\hat{u}}^2$, $\sigma_v^2 = \sigma_y^2 + 2\sigma_{\hat{v}}^2$ are the natural coefficients describing the component size and shape. The transformed coordinates δ_{ij} have intuitive meaning, as illustrated in Fig. 4.8. The first vector

$$
\delta^y_{ij} = \frac{\mathbf{y}_{i_1} - \mathbf{y}'_{i_2}}{2} = \frac{\mathbf{y}'_{i_1} - \mathbf{y}_{i_2}}{2} \qquad (4.23)
$$

is a reflection error, which measures distance between one keypoint and reflection of the other. The second vector

$$
\delta^d_{ij} = \frac{\mathbf{y}_{i_1} + \mathbf{y}'_{i_2}}{2} = \frac{\mathbf{y}'_{i_1} + \mathbf{y}_{i_2}}{2} \qquad (4.24)
$$

is a distance of the correspondence from midpoint or location relative to the axis frame.

Let us now use the following notation to simplify the exponential parametrization:

$\underline{\mathbf{a}} = [\mathbf{a}, 1]$ is a homogeneous representation of a vector $\mathbf{a}$,

$\mathbf{A} : \mathbf{B} = \mathrm{vec}(\mathbf{A})^\top \mathrm{vec}(\mathbf{B})$ is a double inner product of tensors $\mathbf{A}$ and $\mathbf{B}$ (dot product of vectorized matrices).

Since for every symmetric matrix $\mathbf{Q}$ the quadratic form $\mathbf{x}^\top \mathbf{Q} \mathbf{x}$ can be written as $\mathbf{Q} : (\mathbf{x}\mathbf{x}^\top)$, the natural parametrization (4.16) of multinomial normal pdf (4.22) can be compactly written

as

$$\begin{aligned}\log p(y_i \mid \bar{\theta}_j, \hat{\theta}) &= \eta_j^p(\bar{\theta}_j, \hat{\theta}) : \underbrace{\left(\underline{\mathbf{y}_i^p}\,\underline{\mathbf{y}_i^{p\top}}\right)}_{T_i^p} + \eta_j^m(\bar{\theta}_j, \hat{\theta}) : \underbrace{\left(\underline{\mathbf{y}_i^m}\,\underline{\mathbf{y}_i^{m\top}}\right)}_{T_i^m}, \\ \mathbf{y}_i^p &= \frac{1}{2}(\mathbf{y}_{i_1} - \mathbf{y}_{i_2}), \\ \mathbf{y}_i^m &= \frac{1}{2}(\mathbf{y}_{i_1} + \mathbf{y}_{i_2}),\end{aligned} \tag{4.25}$$

where T_i^p, T_i^m are fixed primitive statistics computed from the original locations $y_i = (\mathbf{y}_{i_1}, \mathbf{y}_{i_2})$. The natural parameters for a given component j are

$$\begin{aligned}\eta_j^p(\bar{\theta}_j, \hat{\theta}) &= \begin{bmatrix} -\mathbf{S}_j^p & \mathbf{0} \\ \mathbf{0} & -\log(2\pi\sigma_u\sigma_y) \end{bmatrix} + \mathbf{U}(\sigma_0), \\ \eta_j^m(\bar{\theta}_j, \hat{\theta}) &= \begin{bmatrix} -\mathbf{S}_j^m & \mathbf{S}_j^m\mu_j \\ \mu_j^\top\mathbf{S}_j^m & -\mu_j^\top\mathbf{S}_j^m\mu_j - \log(2\pi\sigma_v\sigma_y) \end{bmatrix} + \mathbf{U}(\sigma_0), \\ \mathbf{U}(\sigma_0) &= \operatorname{diag}\left(\frac{1}{2\sigma_0^2}, \frac{1}{2\sigma_0^2}, \log(2\pi\sigma_0^2)\right),\end{aligned} \tag{4.26}$$

where $\mathbf{U}(\sigma_0)$ accounts for the universal model and $\mathbf{S}_j^m = \mathbf{F}_j\mathbf{L}_m^2\mathbf{F}_j^\top$, $\mathbf{S}_j^p = \mathbf{F}_j\mathbf{L}_p^2\mathbf{F}_j^\top$ are projections of diagonal precision matrices

$$\begin{aligned}\mathbf{L}_p &= \frac{1}{\sqrt{2}}\operatorname{diag}(\sigma_u^{-1}, \sigma_y^{-1}), \\ \mathbf{L}_m &= \frac{1}{\sqrt{2}}\operatorname{diag}(\sigma_y^{-1}, \sigma_v^{-1}).\end{aligned} \tag{4.27}$$

4.4.1.2 Orientation Symmetry

We model orientations ϕ_i by combining two circular normal (von Mises) pdfs in

$$\begin{aligned}p(\phi_i \mid \varphi_j;\, \kappa_o) &= \underbrace{\mathcal{N}_c(\phi_{i_1} + \phi_{i_2};\, 2\varphi_j, \kappa_o)}_{\text{symmetry}}\,\underbrace{\mathcal{N}_c(\phi_{i_1};\, \phi_{i_2} + \pi, \kappa_o)}_{\text{oppositeness}} = \\ &= \frac{1}{4\pi^2 I_0(\kappa)}\exp\left[-2\kappa_o\sin(\varphi_j - \phi_{i_1})\,\sin(\varphi_j - \phi_{i_2})\right],\end{aligned} \tag{4.28}$$

where κ_o is the concentration parameter. The symmetry term in (4.28) models condition on keypoint orientations ϕ_{i_1}, ϕ_{i_2} to be symmetric according to the given axis φ_j and the distribution has a mode at

$$\varphi_j = \frac{\phi_{i_1} + \phi_{i_2}}{2}. \tag{4.29}$$

The prior term in (4.28) prefers correspondences with opposite keypoint orientations to avoid ambiguous straight edge correspondences. The distribution has a mode at

$$\phi_{i_1} = \phi_{i_2} + \pi. \tag{4.30}$$

An image of a linear object (e.g. pole, bar or profile) has a preferred longitudinal axis, but also (infinitely) many lateral axes of local reflection symmetry. A similar situation is on any longer straight edge with homogeneous surroundings.

The exponential parametrization involves trigonometric expansion to separate natural parameter vector η_j^o in

$$\begin{aligned}
\log p(\phi_i \mid \varphi_j) &= \eta_j^o \cdot T_i^o - 2\log(2\pi I_0(\beta_o)), && (4.31)\\
\eta_j^o &= \kappa \left[2\sin(2\varphi_j),\, 2\sin^2(\varphi_j),\, 1 \right], && (4.32)\\
T_i^o &= \left[\sin(\phi_{i_1} + \phi_{i_2}),\, \cos(\phi_{i_1} + \phi_{i_2}),\, \sin(\phi_{i_1})\sin(\phi_{i_2}) \right], && (4.33)\\
\log p(\phi_i \mid \varphi_j) &= \kappa(\mathbf{u}_{i_2}^\top \mathbf{R}_j \mathbf{u}_{i_1} - \mathbf{u}_{i_1}^\top \mathbf{u}_{i_2}) = \kappa\, \mathbf{R}_j : (\mathbf{u}_{i_1}\mathbf{u}_{i_2}^\top) - \kappa \mathbf{u}_{i_1}^\top \mathbf{u}_{i_2}, && (4.34)\\
\mathbf{R}_j &= \mathbf{I} - 2\mathbf{u}_j\mathbf{u}_j^\top, && (4.35)
\end{aligned}$$

where $\mathbf{u}_{i_1} = (\sin\phi_{i_1},\, \cos\phi_{i_1})$.

4.4.2 Appearance Symmetry

In addition to the geometric attributes of a correspondence x_i we also compare the appearance attributes of the two keypoints i_1 and i_2, namely scales s_i and descriptors d_i, m_i. This helps to differentiate correct correspondences from background. We use *primitive feature* functions for the comparison derived from Loy and Eklundh (2006), where the features are combined using arbitrary weight into a single scalar measure. We instead specify pdfs for each feature and combine them with

$$p(s_i, d_i, m_i \mid \bar{\theta}_j, \hat{\theta}) = p(s_i)\, p(d_i)\, p(m_i). \tag{4.36}$$

4.4.2.1 Scale Symmetry

Keypoints are detected at different scales, which also influences the size of an surrounding image patch encoded in descriptor D. Comparison of descriptors from largely different scales does not reflect similarity of the image regions. This brings us to prefer correspondences with similar scale. We compare keypoint scales with Beta-like distribution

$$p(s_i;\, \beta_s) = \frac{1}{Z_s(\beta_s)} \left(\frac{4 s_{i_1} s_{i_2}}{(s_{i_1} + s_{i_2})^2} \right)^{\beta_s}, \tag{4.37}$$

where $Z_s(\beta_s) = \frac{1}{\beta_s - 1}(\sqrt{\pi}\frac{\Gamma(\beta_s+1)}{\Gamma(\beta_s+\frac{1}{2})} - 2)$, the Γ is the gamma function and $\beta_s > 1$ is a concentration parameter. The exponential parametrization is straightforward:

$$\log p(s_i;\, \beta_s) = \underbrace{(\beta_s - 1)}_{\eta^s} \underbrace{(\log(4 s_{i_1} s_{i_2}) - 2\log(s_{i_1} + s_{i_2}))}_{T_i^s} - \log Z_s(\beta_s). \tag{4.38}$$

4.4.2.2 Descriptor Symmetry

We define pdfs for descriptor similarity features in d_i, m_i to prefer small differences between the descriptors. Descriptor similarity measure d_i from (4.7) and self similarity measure m_i from (4.9) introduced in Sec. 4.2.3 are independent and combined in Beta distributions

$$p(d_i;\, \beta_d) = \mathrm{Be}(d_i;\, 1, \beta_d) = \beta_d \,(1 - d_i)^{\,\beta_d - 1}, \tag{4.39}$$

$$p(m_i;\, \beta_m) = \mathrm{Be}(m_i;\, 1, \beta_m) = \beta_m \,(1 - m_i)^{\,\beta_m - 1}, \tag{4.40}$$

where $\beta_d > 1$ and $\beta_m > 1$ are concentration parameters. The exponential parametrization is

$$\log p(d_i, m_i;\, \beta_d, \beta_m) = \eta^s : T_i^s, \tag{4.41}$$

$$\eta^s = [\beta_d - 1,\, \beta_m - 1,\, \log \beta_d \beta_m], \tag{4.42}$$

$$T_i^s = \log\left[1 - d_i,\, 1 - m_i,\, 1\right]. \tag{4.43}$$

Data Clustering Model Breakdown

$$p(X, Z \mid k, \theta) = \prod_{i=1}^{n} \sum_{j=0}^{k} z_{ji}\, p_j \overbrace{p(\underbrace{\mathbf{y}_{i_1}, \mathbf{y}_{i_1}, \phi_{i_1}, \phi_{i_2}, s_{i_1}, s_{i_2}, d_i, m_i}_{x_i} \mid \underbrace{\mu_j, \varphi_j}_{\bar\theta_j}, \underbrace{\sigma_u, \sigma_v, \sigma_y}_{\hat\theta})}^{\text{likelihood}}$$

$$p(x_i \mid \bar\theta_j, \hat\theta) = \underbrace{\underbrace{p(\underbrace{\mathbf{y}_{i_1}, \mathbf{y}_{i_1}}_{y_i} \mid \underbrace{\mu_j, \varphi_j}_{\bar\theta_j}, \underbrace{\sigma_u, \sigma_v, \sigma_y}_{\hat\theta})}_{\text{location}}\, \underbrace{p(\underbrace{\phi_{i_1}, \phi_{i_2}}_{\phi_i} \mid \varphi_j)}_{\text{orientation}}\, \underbrace{p(\underbrace{s_{i_1}, s_{i_2}}_{s_i})}_{\text{scale}}\, \underbrace{p(d_i)\, p(m_i)}_{\text{descriptors}}}$$

4.4.3 Universal Model

The basis of the two-level inference is the model selection. Since empty configuration ($k = 0$) is also an admissible result in case when data cannot be explained as a set of symmetric objects, we need a data model for this case as well, which we call the *universal model.* It must not be specific to the problem at hand since its role is to explain arbitrary data.

Let us assume the outliers come from a universal model that is described by a probability distribution $p_0(X)$, which has few (fixed) parameters. The universal model must be able to explain all primitives in X. The possible presence of an instance of the model of interest will always be judged against the per-primitive universal model indexed as a virtual component with $j = 0$. This extension of the function (4.15) is denoted as the function

$$p(x_i \mid \bar\theta_0;\, \sigma_0)\,, \qquad i = 1, \ldots, n\,. \tag{4.44}$$

The data term for a non-matching correspondence x_i (outlier) belonging to the universal

background model is calculated as

$$p(x_i \mid \bar{\theta}_0;\, \sigma_0) = p(y_i \mid \bar{\theta}_0;\, \sigma_0)\, p(\phi_i \mid \bar{\theta}_0)\, p(s_i \mid \bar{\theta}_0)\, p(d_i \mid \bar{\theta}_0) \tag{4.45}$$

$$p(y_i \mid \bar{\theta}_0;\, \sigma_0) = \mathcal{N}(\mathbf{y}_{i_1};\, \mathbf{0}, \sigma_0)\, \mathcal{N}(\mathbf{y}_{i_2};\, \mathbf{0}, \sigma_0), \tag{4.46}$$

$$p(\phi_i \mid \bar{\theta}_0) = \frac{1}{4\pi^2}, \tag{4.47}$$

$$p(s_i \mid \bar{\theta}_0) = p(d_i \mid \bar{\theta}_0) = p(m_i \mid \bar{\theta}_0) = 1, \tag{4.48}$$

where the universal model is uniform for all appearance features on the unit interval.

4.5 Shape Prior

The shape parameter set becomes

$$\hat{\theta} = \{\sigma_y, \sigma_u, \sigma_v\}\,. \tag{4.49}$$

The priors for σ_u^2, σ_v^2, σ_y^2 are inverse gamma distributions

$$p(\hat{\theta}) = p(\sigma_u)\, p(\sigma_v)\, p(\sigma_y), \tag{4.50}$$

where

$$p(\sigma_u) = \mathcal{IG}(\sigma_u;\, \alpha_u, \beta_u) = \frac{\beta_u^{\alpha_u}}{\Gamma(\alpha_u)}\, \sigma_u^{2(\alpha_u - 1)}\, \exp\left(-\beta_u \sigma_u^2\right),$$

and the priors $p(\sigma_v)$ and $p(\sigma_y)$ are defined analogically. We denote the set of the associated hyperparameters as $\hat{\alpha} = \{\alpha_u, \alpha_v, \alpha_y\}$ and $\hat{\beta} = \{\beta_u, \beta_v, \beta_y\}$.

4.6 Component Model

Multiple reflection symmetric components are in fact usually not independent. From a top-level point of view we can also model regularity of the entire component set. Unlike previous terms which address individual components independently, this prior describes how multiple components should interact with each other.

Let us define a *grouping field* $\breve{Z}$ for k components into $\breve{k} \leq k$ *component groups*[5]

$$\breve{Z} = \{\breve{z}_{gj}\} \in \{0, 1\}^{\breve{k} \times k}, \tag{4.51}$$

where $\breve{z}_{gj} = 1$ when the component j belongs to the group g, which is an analogy to primitive allocation field Z (Sec. 4.4). Each component belongs to exactly one group ($\sum_{g=1}^{\breve{k}} \breve{z}_{gj} = 1$), the possible groupings range from all components in one group ($\breve{k} = 1$) to each component

[5] In this section we will use the term *group* in a more general sense, i.e. it can be any set of components that do not necessarily form a (mathematical) *symmetry group* as defined in Introduction.

in its own group ($\breve{k} = k$). Let $G(g) = \{j;\, \breve{z}_{gj} = 1\}$ be a set of indices of components in the group g. We assume the component model can be written as component-wise product

$$p(\bar{\theta}, \breve{Z} \mid \breve{\theta}, \breve{k}, \hat{\theta}) = C(\breve{Z}, \breve{k}, k) \prod_{j \in G(g)} p(\Psi_j \mid \bar{\theta}_j, \hat{\theta})\, p(\bar{\theta}_j \mid \breve{\theta}_g, \breve{Z}, \hat{\theta}) = \tag{4.52}$$

$$= C(\breve{Z}, \breve{k}, k) \prod_{j=1}^{k} p(\Psi_j \mid \bar{\theta}_j, \hat{\theta}) \sum_{g=1}^{\breve{k}} \breve{z}_{gj}\, p(\bar{\theta}_j \mid \breve{\theta}_g, \hat{\theta}) \tag{4.53}$$

where $\breve{\theta}$ are the *group parameters* defined as

$$\breve{\theta} = (\breve{\theta}_1, \ldots, \breve{\theta}_g, \ldots \breve{\theta}_{\breve{k}}), \tag{4.54}$$

and Ψ_j are *component features* specific to component j. The combinatorial term $C(\breve{Z}, \breve{k}, k)$ accounts for component index identity in

$$C(\breve{Z}, \breve{k}, k) = \frac{k!}{\prod_{g=1}^{\breve{k}} k_g!}\, \breve{k}!, \tag{4.55}$$

where $k_g = \sum_{j=1}^{k} \breve{z}_{gj}$ is the number of components in a group g and $\sum_{g=1}^{\breve{k}} k_g = k$. We must sum over all permutations of indices within a group that give the same observation, because such permutations of component indices are not observable; that gives the multinomial coefficient. In addition to that the term $\breve{k}!$ accounts for identity of groups, now we sum over all permutations of group indices that give the same observation.

4.6.1 Component Features

We can evaluate data properties of the whole component (integrality) in order to separate a single symmetric object from two aligned ones or from the background. In terms of matching correspondence locations we assume additional geometric and appearance properties. We factorize the term from (4.13) as

$$p(\Psi_j \mid \bar{\theta}_j, \hat{\theta}) = p(\psi_j^c \mid \bar{\theta}_j, \hat{\theta})\, p(\psi_j^o \mid \bar{\theta}_j, \hat{\theta}), \tag{4.56}$$

where ψ_j^c and ψ_j^o are component features for compactness and objectness respectively. We assume independence of the features to facilitate efficient stratified inference (Sec. 4.10).

4.6.1.1 Compactness

Although the shape model, where the correspondence distance δ_d is modeled with a Gaussian distribution centered at the axis midpoint μ, assumes an elliptic shape of the symmetric object with correspondences concentrated around the midpoint, this statistic still allows uneven coverage of the object. This can result in accepting false detections formed by several

groups of random or local symmetries joined together or in biasing the true detection by an outlying local symmetry unrelated to the object resulting in large geometric error w.r.t. true axis location.

To avoid such components, let us assume the correspondences should uniformly sample the object contour and interior, then no separating gap along the axis should appear.

We can interpret this assumption so that the corresponding points on one side of the axis should be neighbors to each other. To test the neighborhood condition, we can construct adjacency matrix N for keypoints y using Delaunay triangulation of the original keypoints, which is fixed for a given problem instance.

Then for a given axis we test if there is a gap between them by querying a subset of the adjacency matrix. For each point i_1 we obtain the number of neighbors i_2 such that the geometric error of the pair satisfies $\delta_y \leq \sigma_y$ and location satisfies $\delta_m^u \leq \sigma_u$, $\delta_m^v \leq \sigma_v$. If a point has less than 3 such neighbors, there is a gap around it. The number of gaps $\psi_j^c = \psi^c(\bar{\theta}_j)$ can be modeled with Beta-binomial distribution

$$p(\psi_j^c \mid \bar{\theta}_j, \hat{\theta};\, \alpha_c, \beta_c) = \binom{n_j}{\psi_j^c} \frac{\mathrm{B}(\psi_j^c + \alpha_c,\, n_j - \psi_j^c + \beta_c^c)}{\mathrm{B}(\alpha_c, \beta_c)}, \tag{4.57}$$

where $\alpha_c < 1$, $\beta_c > 1$ are Beta type parameters.

4.6.1.2 Objectness

As discussed in Introduction, scenes (predominantly man-made) often include a number of local implicitly symmetric objects or parts (stripes, rods, corners), which are usually not considered as symmetries of interest (according to human annotations in datasets), because they do not represent an object.

It is difficult to differentiate between the two cases because the measure is subjective, but we can learn a classifier to help us with this decision. This has been studied in the context of general object detection as a class-independent measure of objectness or region saliency, where it is typically used to propose (sample) image regions for further classification.

The method proposed by Alexe et al. (2010) allows to train an objectness classifier using images with annotated object regions. In our case this requires to transform every axis of symmetry into a bounding box. The cues used include multi-scale saliency, color contrast, edge density and superpixel straddling.

The classifier integrates all the cues with Naive Bayes approach so the resulting score is actually objectness posterior

$$p(\psi_j^o = 1 \mid \bar{\theta}_j, \hat{\theta}) = \frac{p(\psi_j^o = 1) \prod_{c \in C} p(c \mid \psi_j^o = 1)}{p(\psi_j^o = 1) \prod_{c \in C} p(c \mid \psi_j^o = 1) + p(\psi_j^o = 0) \prod_{c \in C} p(c \mid \psi_j^o = 0)}, \tag{4.58}$$

where $\psi_j^o \in (0, 1)$ indicates the classification of the component either to *object* ($\psi_j^o = 1$) or *background* ($\psi_j^o = 0$) class and C is the set of above mentioned cues (features) from Alexe et al. (2010).

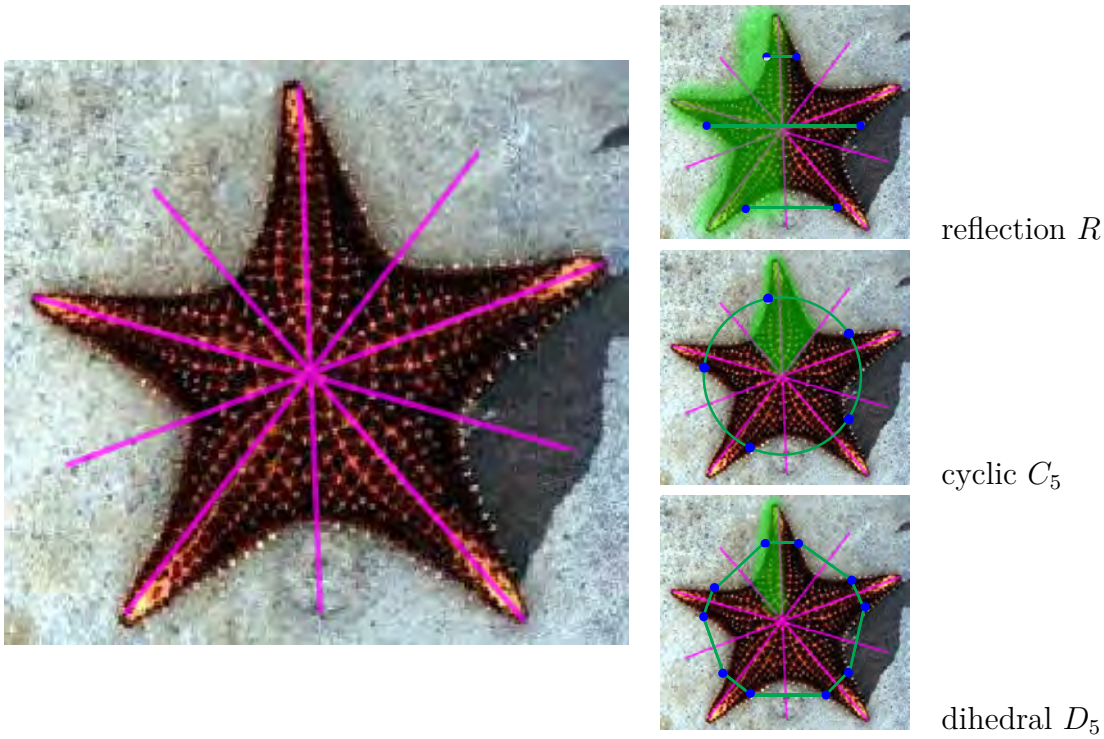


Figure 4.9: An example of dihedral symmetry group (left) with 5 reflection axes (cyan). The shape element repeating in different symmetries (right) is highlighted in green. Keypoints (blue) are connected by elementary symmetries (green lines).

4.6.2 Symmetry Grouping

Symmetry theory presented in Introduction (Sec. 1.2.1) explains which compositions of symmetries (called symmetry groups) can be encountered in 2D. We will implement some of the grouping principles to regularize our model. For example a star-shaped object (like in Fig. 4.9 but considered exact) has multiple symmetries, 5 reflections (R) and 5 rotations (cyclic group C_5), which form a dihedral symmetry group D_5 (Sec. 1.2.1.2). We will discuss options to handle such symmetric objects with our model while taking into account imperfections of the real-world objects resulting in deviations from the exact multiple symmetry, which are observed in the standard symmetry datasets (Liu et al., 2013).

A general symmetry detector should perform model selection w.r.t. the imperfect input and consider all of R, C_n a D_n models and their discrete orders n. In our method we choose the model explicitly based on the facts given below.

The star shape could be explained with a single reflection symmetry R component (Sec. 4.2.2), which should be generally preferred for its simplicity to composed cyclic and dihedral groups (due to 'Occam's razor' (MacKay, 2003)). In the presence of noise and shape imperfections there will be a single reflection axis best matching the given data. There are however five solutions for R annotated in the standard dataset, which forces us to drop this model due to its ambiguity.

The cyclic group C_5 generally allows also reflection-asymetric spikes of the star in Fig. 4.9, but does not include the reflection constraint. Each of the spikes (period) is expected to

have the same shape and a single keypoint is replicated 5×. In this case the primitive data element w.r.t. group C_n is the correspondence of three keypoints from three[6] consecutive spikes (periods). This requires to simultaneously use different primitives for the cyclic groups and different for the reflection components and in practice equals to extending our model to general rotation symmetries. An implicit C_5 model would compare the five cyclic keypoint locations (blue in Fig. 4.9), but in practice there is a low probability of encountering a constellation of five independently detected keypoints lying on a circle due to appearance changes or occlusions. The constellation complexity is a strong reason to avoid the cyclic group model.

The dihedral group D_5 is more specific as a reflection axis constraints the center of the rotated element (phase). Now just a half-spike is expected to reflect and rotate to give the complete shape and its single keypoint is replicated 10×, which makes the constellation argument against the model even stronger. On the other hand the dihedral primitives w.r.t. D_n are now two reflection correspondences with one common keypoint, which allows to reuse reflection primitives but requires to allow two correspondences to share a single keypoint within a group.

We propose to model the dihedral pattern explicitly by grouping reflection symmetry components with axes crossing each other and assigning higher probability to groups with the same angle between axes (rotation constraint). This provides greater flexibility than implicit models given above while all reflection symmetries are part of the solution. The possibility of correspondences sharing a keypoint however needs to be implemented at least in tentative correspondence selection (Sec. 4.2.4).

For frieze symmetry patters (Sec. 1.2.1.2) combining reflection and translation symmetries we could group axes with a similar orientation (parallel) and prefer their equal spacing and alignment (like in Chapter 2), but we do not find enough examples in the evaluated datasets and leave this grouping for future work.

4.6.2.1 Dihedral Group Model

We identify dihedral group components based on two weak constraints: Their axis intersect in a common point within the image frame and their midpoints should be close to each other.

Let us assume a *dihedral grouping* $\breve{Z}$ and associated *dihedral group parameters* $\breve{\theta}_g = \{\breve{\mu}_g, \breve{\varphi}_g\}$, $g = 1, \ldots, \breve{k}$, are given and that the reflection axes in a group g are arranged in a rotation symmetric pattern with rotation center at location $\breve{\mu}_g$ and starting angle (phase) $\breve{\varphi}_g$. This geometrically translates into condition on angle differences: The angles between each two neighboring axes are equally $2\pi/k_g$ depending solely on the number of components in the group k_g (order of rotation). The angle of j-th axis in the group $j = 1, \ldots, m_g$ ordered by $\varphi_j < \varphi_{j+1}$ is then given by

$$\breve{\varphi}_{jg} = \breve{\varphi}_g + (j-1)\frac{\pi}{k_g}. \tag{4.59}$$

[6]Exactly 2.5 points are sufficient to determine the rotation center, only one coordinate of the third point is needed.

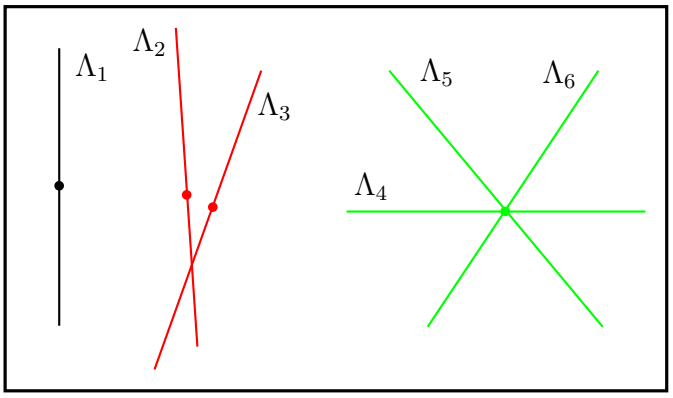


Figure 4.10: Dihedral grouping based on axis intersection with groups color encoded. Rectangle is an image frame, dots indicate axis midpoints μ_j and the axis segment length is $2\sigma_u$. Black component group $\{\Lambda_1\}$ is an isolated component. Consider axis Λ_2, Λ_3 indicate some local reflection symmetries. Red component group $\{\Lambda_2, \Lambda_3\}$ is not rotation symmetric and will receive low probability from component group model, unlike green group $\{\Lambda_3, \Lambda_4, \Lambda_5\}$ which is close to dihedral symmetry group D_3.

In the case there are just two components in the group the axis are preferred to be perpendicular. An isolated component j results in identity $\breve{\varphi}_{jg} = \breve{\varphi}_g$.

The component-wise feature pdf is modeling the deviations of the predicted and actual center locations in

$$p(\bar{\theta}_j \mid \breve{\theta}_g, \hat{\theta}) = p(\varphi_j \mid \breve{\varphi}_{jg})\, p(\mu_j \mid \breve{\mu}_g), \tag{4.60}$$

$$p(\varphi_j \mid \breve{\varphi}_{jg};\, \kappa_\varphi) = \mathcal{N}_c(\varphi_j;\, \breve{\varphi}_{jg}, \kappa_\varphi) = \frac{1}{2\pi I_0(\kappa_\varphi)} e^{\kappa_\varphi \cos(\varphi_j - \breve{\varphi}_{jg})}, \tag{4.61}$$

$$p(\mu_j \mid \breve{\mu}_g;\, \sigma_\mu) = \mathcal{N}(\mu_j;\, \breve{\mu}_g, \sigma_\mu), \tag{4.62}$$

where $\kappa_\varphi, \sigma_\mu$ are the concentration parameters and $\mathcal{N}_c$ is the circular normal (von Mises) pdf with I_0 as modified Bessel function of order 0.

4.6.2.2 Natural Parameters

The exponential form of the component model is

$$\log p(\psi_j^c \mid \bar{\theta}_j, \hat{\theta};\, \alpha_c, \beta_c) = \log \frac{\Gamma(\psi+\alpha)\Gamma(n-\psi+\beta)}{\Gamma(\psi+1)\Gamma(n-\psi+1)} + \log \frac{\Gamma(n+1)}{\Gamma(n+\alpha+\beta)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \tag{4.63}$$

$$\log p(\psi_j^o \mid \bar{\theta}_j, \hat{\theta}) = \log p(\psi_j^o = 1 \mid \bar{\theta}_j, \hat{\theta}), \tag{4.64}$$

$$\log p(\mu_j \mid \breve{\mu}_g;\, \sigma_\mu) = -\frac{1}{2\sigma_\mu^2} \begin{bmatrix} \mathrm{diag}(1,1) & \mu_j \\ \mu_j^\top & \mu_j^\top \mu_j \end{bmatrix} : (\breve{\mu}_g \breve{\mu}_g^\top) - \log(2\pi\sigma_\mu^2), \tag{4.65}$$

$$\log p(\varphi_j \mid \breve{\varphi}_{jg};\, \kappa_\varphi) = \kappa_\varphi\, \mathbf{u}_j^\top \breve{\mathbf{u}}_{jg} - \log(2\pi I_0(\kappa_\varphi)), \tag{4.66}$$

where $\breve{\mathbf{u}}_{jg} = (\cos\breve{\varphi}_{jg},\, \sin\breve{\varphi}_{jg})$.

Component Model Breakdown

$$p(\bar{\theta}, \breve{Z} \mid \breve{\theta}, \breve{k}, \hat{\theta}) = \prod_{j=1}^{k} \overbrace{p(\underbrace{\psi_j^o, \psi_j^c}_{\Psi_j} \mid \underbrace{\mu_j, \varphi_j}_{\bar{\theta}_j}, \underbrace{\sigma_u, \sigma_v, \sigma_y}_{\hat{\theta}})}^{\text{integrality}} \sum_{g=1}^{\breve{k}} \breve{z}_{gj} \overbrace{p(\underbrace{\mu_j, \varphi_j}_{\bar{\theta}_j} \mid \underbrace{\breve{\mu}_g, \breve{\varphi}_g}_{\breve{\theta}_g}, \underbrace{\sigma_u, \sigma_v, \sigma_y}_{\hat{\theta}})}^{\text{likelihood}}$$

$$p(\bar{\theta}_j \mid \breve{\theta}_g, \hat{\theta}) = \underbrace{p_I(\bar{\theta}_j \mid \breve{\theta}_g, \hat{\theta})}_{\text{intersection}} \underbrace{p(\varphi_j \mid \breve{\varphi}_{jg})}_{\text{angle}} \underbrace{p(\mu_j \mid \breve{\mu}_g)}_{\text{center}}$$

4.7 Component Group Prior

While components should be concentrated around the group center, the opposite should hold for the group centers $\breve{\mu}$ that should be spread out. Without this assumption the grouping would tend to degenerate configurations with each component in its own group. We have used the bounded domain $\breve{\mu} \in (0, 1)^2$ (rather then unbounded $\mathbb{R}$) for the group prior (4.68) to be a proper pdf and to implicitly restrict it to the image frame. We compute the mean distance of a group g from other groups $h \neq g$ with

$$d_g = \frac{1}{2(\breve{k} - 1)} \sum_h \|\breve{\mu}_g - \breve{\mu}_h\|^2, \tag{4.67}$$

and model this feature with Beta distribution preferring $d_g \to 1$ in

$$p(\breve{\theta}) = \prod_{g=1}^{\breve{k}} p(d_g \mid \breve{\mu}_g)\, p(\breve{\mu}_g)\, p(\breve{\varphi}_g), \tag{4.68}$$

$$p(d_g \mid \breve{\mu}_g;\, \beta_g) = \mathrm{Be}(d_g;\, \beta_g, 1) = \beta_g (d_g)^{\beta_g - 1}, \tag{4.69}$$

where β_g controls the concentration. In the case $\breve{k} = 1$ we assume $d_g = 1$. The prior $p(\breve{\mu}_g) = 1$ is uniform on the unit image frame and $p(\breve{\varphi}_g) = \frac{1}{2\pi}$ is uniform on the circle.

We argue that the search for the dihedral group D_n order k_g does not require model selection scheme (Sec. 1.3.3.4) because the number of group parameters is fixed and does not depend on the number of components k_g in the group (unlike when the component parameter dimension changes with k).

4.8 Configuration Prior

The role of a default configuration prior in Šára (2014) is to regularize parameter estimation. The regularizing assumption is that the number of inliers per component is equal in all components. A Dirichlet distribution with equal parameters corresponding to non-background

components softens this constraint

$$p(\dot{\theta} \mid k;\, \alpha_0, \alpha_I) = \mathrm{Dir}(p_0, p_1, \ldots, p_k;\, \alpha_0, \alpha_I) = \frac{\Gamma(\alpha_O + k\,\alpha_I)}{\Gamma(\alpha_O)\,\Gamma(\alpha_I)^k}\, p_0^{\alpha_O - 1} \prod_{j=1}^{k} p_j^{\alpha_I - 1}\,, \tag{4.70}$$

where $p_j \in \dot{\theta}$ as in (4.14). The mode of this prior is at

$$p_j^* = \begin{cases} \dfrac{\alpha_O - 1}{\alpha_O + k\,\alpha_I - (k+1)}, & j = 0, \\ \dfrac{\alpha_I - 1}{\alpha_O + k\,\alpha_I - (k+1)}, & j > 0. \end{cases}$$

4.9 Complexity Priors

The role of a default prior on complexity k in Šára (2014) is to help select empty configuration when data is not containing any object instance. Note there are at most n components in a configuration, therefore $k \leq n$. We use binomial distribution

$$p(k;\, p_c) = \binom{n}{k} p_c^k\, (1 - p_c)^{n-k}, \tag{4.71}$$

where p_c is the component ratio. The mode of this prior is (approximately) at

$$k^* \approx (n+1)\, p_c\,. \tag{4.72}$$

On the next level the group complexity $\breve{k} \leq k$ is similarly modeled with

$$p(\breve{k};\, p_g) = \binom{k}{\breve{k}} p_g^{\breve{k}}\, (1 - p_g)^{k-\breve{k}}, \tag{4.73}$$

where p_g is the group ratio.

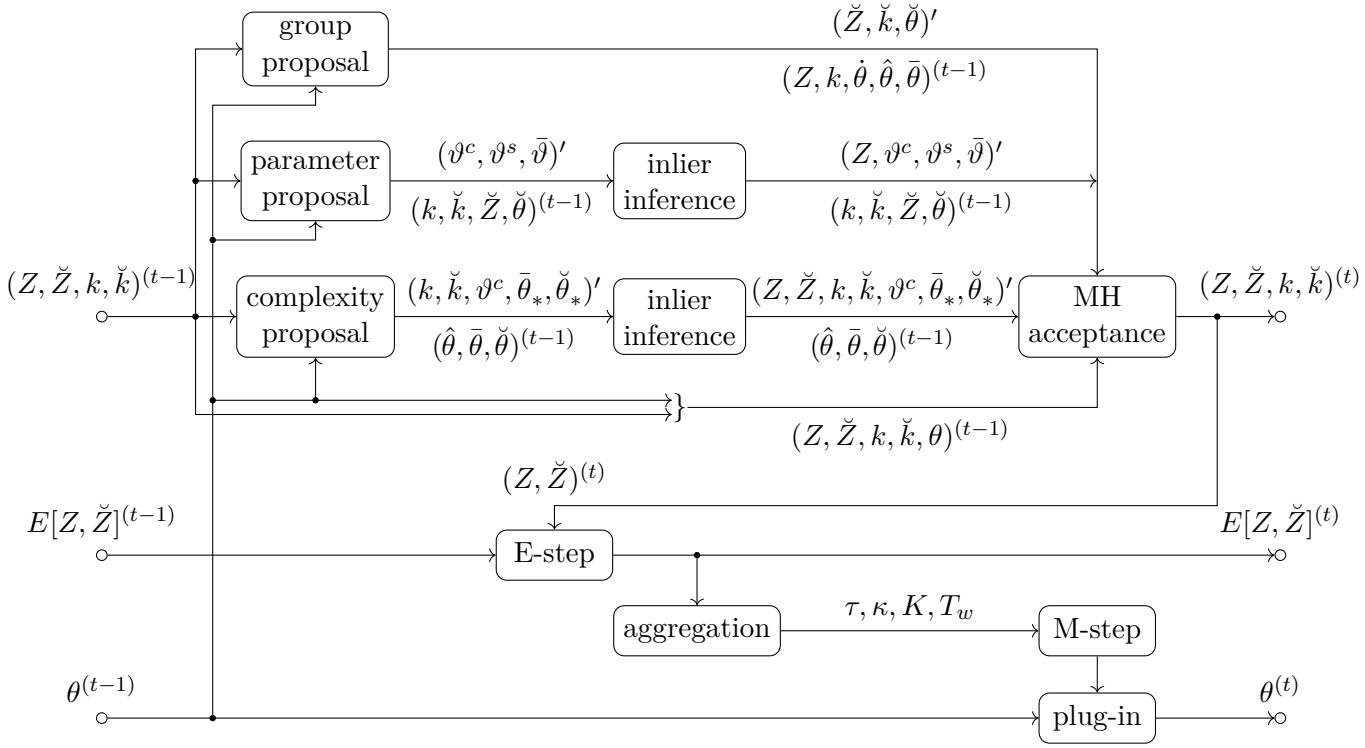


Figure 4.11: LiSAEM inference diagram adapted from Šára (2014). This block represents one major iteration of the algorithm, i.e. the time step $(t-1) \mapsto (t)$. The ϑ' is an update of a local copy of $\theta^{(t-1)}$. The τ, κ, K, T_w are statistics output by the E-step and needed by the M-step. Variables above each arrow are updated, output variables below each arrow are just copies of the input sent to the input of a block.

4.10 Inference

As mentioned above, we have chosen to follow two-level inference scheme to perform model selection (Sec. 1.3.3.4), where the 'models' are the different complexities k.

In particular we use the LiSAEM algorithm (Šára, 2014) introduced in Sec.1.3.4.6. In this section we will overview its design and add parts specific to the reflection symmetry model described in the previous sections.

4.10.1 Algorithm Overview

LiSAEM is a generic two-level inference engine for the problems in the form of Sec. 1.3.4.6, where the model selection is performed over k. The principal components are a sampler from the posterior distribution (1.23) and a stochastic approximation EM algorithm for estimating the maximum posterior parameters θ whose Q-function is

$$Q(\theta \mid \theta^{(t-1)}, Z^{(t-1)}, \breve{Z}^{(t-1)}, k^{(t-1)}, \breve{k}^{(t-1)}) = \mathcal{E}\Big[\, \mathcal{L}(\theta; X, Z, \breve{Z}, k, \breve{k}) \,\Big], \tag{4.74}$$

which is the lower bound on the target likelihood $\mathcal{L}(\theta; X, Z, \breve{Z}, k, \breve{k})$ from (4.12) and where t is the time step (iteration), θ is the free variable and $\mathcal{E}[f(\theta; x)]$ is the expectation of $f(\theta; x)$ over the posterior distribution of x. The expectation (E-step) is implemented by means of a Metropolis-Hastings (MH) sampler made efficient with a stochastic averaging filter and the maximization (M-step) is deterministic. Simultaneously, the sampler is used to estimate the posterior distribution $p(k \mid X, \dot{\theta}, \hat{\theta})$ by histogramming the generated samples. The mode of the histogram is the most probable posterior complexity.

To implement a new probabilistic model of the type of (4.12) with LiSEAM an user needs to write a few callbacks that define the target distribution (4.12). For the sake of completeness of this thesis we briefly describe these parts of LiSAEM that employ the callbacks. The core blocks of LiSAEM are summarized in Fig. 4.11 and we will now explain them in some detail. The full specification and derivation can be found in Šára (2014).

4.10.2 Inlier Inference

This is a core deterministic mechanism. The role of this procedure is to map random parameters $\bar{\theta}, \hat{\theta}, \breve{\theta}$ to configurations Z. Given all parameters $\theta = (\bar{\theta}, \hat{\theta}, \breve{\theta}, \dot{\theta})$ the inlier inference maps each primitive i to a definite component j.

Primitive inlier inference maximizes the likelihood ratio (4.16) by

$$Z' = \arg\max_{Z} p(X, Z \mid k, \bar{\theta}, \hat{\theta}), \tag{4.75}$$

which breaks down to independent primitive allocations $z_i \in \{0, \dots, k\}$ to components

$$z'_i = \arg\max_{z_i} p(x_i, z_i \mid \bar{\theta}, \hat{\theta}),\ i = 1, \dots, n. \tag{4.76}$$

This in effect allocates each primitive to the most likely component $\{0, \dots, k\}$ (including background).

4.10.3 Complexity Proposals

In a Metropolis-Hastings (MH) sampler with reversible jumps across different configuration complexities k (Sec. 1.3.4) a generalization of *Sequential Multipoint Metropolis* (SMM) method (Qin and Liu, 2001) is used. Its complexity proposal scheme consists of a sequence of elementary proposals: A *disassembly* proposal decreases complexity k by unity, by deleting a random component from the configuration, and an *assembly* proposal initializes a new component location parameters $\bar{\theta}_j$ (RANSAC-like empirical distribution sample) and copies the other components' parameters.

Propose component parameters (callback). Parameters $\bar{\theta}_j$ of a component represented by a symmetry axis are proposed by a single correspondence (*seed*) x_i determining its parameters μ_j and φ_j using (4.6).

Then configuration Z is inferred by means of deterministic inlier inference (Sec. 4.10.2). One of the sequence of proposed configurations Z is then randomly selected and subjected to the standard MH acceptance rule in which the forward and reverse proposal probabilities are computed in a specific way.

The sequential multipoint proposal scheme has several major advantages over the basic RJ-MCMC: improved mixing and the possibility to perform only approximate incremental inference to improve computational efficiency.

4.10.4 Group Proposals

The existence of groups is linked to the underlying components. We define the implicit grouping changes to $\breve{Z}$ when a complexity change is proposed (see section above).

Following *assembly*, a new component becomes a single member of its own new group with $\breve{\mu}_* = \mu_*$.

Following *disassembly*, the deleted component is removed from its group. If it was the group's last member, then the group is deleted too.

The actual group proposals follow the merge and split procedure. Let us first define a weighted complete graph C where the current configuration components correspond to graph nodes and graph edge weights $w_{ij}^{\varphi}, w_{ij}^{\mu}$ evaluate the conditions on axis intersection and midpoint distance

$$w_{ij}^{\varphi} = \begin{cases} 1, & (\Lambda_i \cap \Lambda_j) \in \operatorname{dom} I, \\ 0, & \text{otherwise}, \end{cases} \tag{4.77}$$

$$w_{ij}^{\mu} = \|\mu_i - \mu_j\| / \sigma_u, \tag{4.78}$$

where $i, j \in J$ are configuration components and $(\Lambda_i \cap \Lambda_j)$ is the intersection point of the two axes. A necessary condition for two axes i, j to belong to the same group is $w_{ij}^{\varphi} = 1$. Then we construct an edge-induced subgraph $C' \subseteq C$ by randomly thresholding edge weights w_{ij}^{μ} and including only selected edges with the condition satisfied.

Merge joins two groups together. Only group pairs (g, h) within the same connected component of graph C' are considered and one such pair is uniformly sampled, $g \neq h$. The merged group parameter becomes

$$\breve{\mu}_* = \frac{1}{2} \left(\breve{\mu}_g + \breve{\mu}_h \right). \tag{4.79}$$

Split removes a randomly selected component g from a group. Only groups with two or more components are considered and uniformly sampled in the first step. In the next step the component to be removed is sampled uniformly within the chosen group and becomes a single member of its own new group with $\breve{\mu}_* = \mu_g$.

4.10.5 Parameter Proposals

The purpose of a parameter proposal is to provide exploration in configuration space with parameter θ. In practice LiSAEM without this proposal tends to get stuck in a configuration when the core parameters $\dot{\theta}$ are not estimated correctly. The exploration ability of the parameter proposal helps jump out of such configuration by following the gradient of target distribution stochastically.

The parameter proposal is based on a modification of *Metropolis-Adjusted Langevin* (MALA) algorithm by (Roberts and Tweedie, 1996). Given the current parameter value $\theta^{(t-1)}$, MALA proposes samples ϑ' as

$$\vartheta' = \theta^{(t-1)} + \frac{\sigma_\theta^2}{2} \nabla \log \pi(\theta^{t-1}) + \zeta, \tag{4.80}$$

in which ζ is a normally distributed random variable $\zeta \sim N(0; \sigma_\theta^2)$ and π is the target distribution (4.12). The ϑ' is then accepted using the standard MH acceptance rule.

LiSAEM uses a modification of MALA called *Scaled Stochastic Newton* (SSN) algorithm (Bui-Thanh and Ghattas, 2012), in which the gradient is scaled by the inverse of negative Hessian in a Newton-like step. Instead of the logarithm of the target distribution $\log \pi(\theta^{(t-1)})$ the Q function (4.74) is used, provided by the SAEM algorithm. The stochastic approximation filter of SAEM helps provide temporal stability of such gradient and Hessian estimates and improves mixing.

On-line adaptation of the scaling constant σ_θ from (4.80) is performed according to Atchadé (2006). The goal of adaptation is to provide acceptance rate close to the optimal value (approximately 0.574 derived for standard normal distribution (Roberts and Tweedie, 1996)) and improve stability of SSN. .

Since an E-step and an M-step of SAEM follow the process illustrated in Fig. 4.11 the proposed parameter ϑ' is essentially forgotten. In fact, since ϑ' is used in inlier inference, it influences the proposed configuration Z, so it does contribute to the next estimate $\theta^{(t)}$.

4.10.6 E-step

The **E-step** performs expectation of Q-function (4.74) which in our representation can be written as summation over all possible configurations $Z, \breve{Z}$ in

$$\begin{aligned} &Q(\theta \mid \theta^{(t-1)}, Z^{(t-1)}, \breve{Z}^{(t-1)}, k^{(t-1)}, \breve{k}^{(t-1)}) = \\ &= \sum_{k=0}^{n} \sum_{\breve{k}=1}^{k} \sum_{Z_k} \sum_{\breve{Z}_{\breve{k}}} p(Z_k, k, \breve{Z}_{\breve{k}}, \breve{k} \mid X, \theta^{(t-1)})\, \mathcal{L}(\theta; X, Z, \breve{Z}, k, \breve{k}), \end{aligned} \tag{4.81}$$

where t denotes the current time step (iteration), Z_k represents the set of fields for fixed complexity k and the maximum complexity is the number of primitives, $k \leq n$. Analogically

$\breve{Z}_{\breve{k}}$ represents the set of fields for fixed number of groups $\breve{k}$ limited by $\breve{k} \leq k$.

It assumes it is possible to track the identity of each component j from a set of all components J in the sequence of configurations produced by the sampler, even if the component disappears from the configuration and reappears later. The $p(Z_k, k, \breve{Z}_{\breve{k}}, \breve{k} \mid X, \theta^{(t-1)})$ is estimated by the sampler using Robbins-Monro (RM) sequential update scheme. It can be shown it leads to

$$P^{(t-1)}(\theta) = (1 - \gamma_t)\, P^{(t-1)}(\theta) + \gamma_t\, Q^{(t-1)}(\theta \mid \theta^{(t-1)}), \tag{4.82}$$

in which t is the simulation step index, $Q^{(t-1)}$ is the Q-function (4.81) in which the expectation is replaced by a single random sample from the (target) posterior $p(z_{ji}, k, \breve{z}_{gj}, \breve{k} \mid X, \theta^{(t-1)})$; this is a limiting case of empirical averaging. The γ_t is a sequence of diminishing multipliers such that RM guarantees convergence (Delyon et al., 1999): $\gamma_t > 0$, $\sum_t p_t = \infty$, $\sum_t p_t^2 < \infty$. In our case the P-function of SAEM has the form of

$$P^{(t)}(\theta) = P_0^{(t)}(\theta) + \sum_{j \in J} P_j^{(t)}(\theta), \tag{4.83}$$

where the default term comes from (4.70) as

$$P_0^{(t)}(\theta) \propto \log p(\hat{\theta}) + \log p(\breve{\theta}) + \left(\tau_0^{(t)} + \alpha_O - 1\right) \log p_0\,, \tag{4.84}$$

and the component-wise term for component j in the group g has the form of

$$P_j^{(t)}(\theta) \propto \nu_j(\bar{\theta}_j, \hat{\theta})\, \mathcal{K}_j^{(t)} + \left(\tau_j^{(t)} + (\alpha_I - 1)\kappa_j^{(t)}\right) \log p_j + \sum_{w=1}^{W} \eta_{jw}(\bar{\theta}_j, \hat{\theta})\, \mathcal{T}_{jw}^{(t)}, \tag{4.85}$$

where $\propto$ means here 'up to an additive constant' and terms are the following:

$\nu_j(\bar{\theta}_j, \hat{\theta})$ is the universal model statistic (4.44),

$\tau_{ji}^{(t)} = (1 - \gamma_t)\tau_{ji}^{(t-1)} + \gamma_t z_{ji}$ is the current estimate of primitive allocation z_{ji},

$\tau_j^{(t)} = \sum_{i=1}^{n} \tau_{ji}^{(t)}$ is the current estimate of number of inliers in the component j, (τ_0 for outliers),

$\kappa_j^{(t)} = (1 - \gamma_t)\kappa_j^{(t-1)} + \gamma_t$ is the posterior expectation of component's presence in the current configuration (inclusion),

$\kappa^{(t)} = \sum_{j \in J} \kappa_j^{(t)}$ is the current estimate of the posterior expectation of complexity k,

$\mathcal{K}_j^{(t)}$ is the current number of inliers in the component j,

$\mathcal{T}_{jw}^{(t)} = \sum_{i=1}^{n} \tau_{ji}^{(t)} T_w(x_i)$ is the statistic aggregated from all primitives in (4.16),

$\eta_{jw}(\bar{\theta}_j, \hat{\theta})$ are natural parameters (callback) given in (4.25) and (4.63).

The E-step in LiSAEM thus just estimates the quantities τ_{ji} and κ_j.The appeal of SAEM over the standard versions of Monte-Carlo EM algorithms is that it makes use of all simulated samples for the hidden variables $Z^{(t-1)}$ and also leads to computationally efficient parameter update scheme.

4.10.7 M-step

The M-step implements parameter estimation to maximize the target function

$$\theta^{(t)} = \arg\max_{\theta} P^{(t-1)}(\theta), \tag{4.86}$$

where θ are the estimated parameters while the statistics $\mathcal{T}$ are fixed and the implementation is specific to the given problem.

In the case of symmetry detection model we have statistics $\mathcal{T}_j^d = \sum_i \underline{\mathbf{y}_d^i}\,\underline{\mathbf{y}_d^i}^\top$,$\mathcal{T}_j^m = \sum_i \underline{\mathbf{y}_m^i}\,\underline{\mathbf{y}_m^i}^\top$ and the active part of P-function is

$$\begin{aligned} P^{(t)}(\sigma_u, \sigma_v, \sigma_y, \mu, \mathbf{u}, \mathbf{v}) &= -(\alpha_u + 1)\log\sigma_u^2 - (\alpha_v + 1)\log\sigma_v^2 - (\alpha_y + 1)\log\sigma_y^2 \\ &+ \sum_{j=1}^{k} P_j^{(t)}(\sigma_u, \sigma_v, \sigma_y, \mu_j, \mathbf{u}_j, \mathbf{v}_j), \end{aligned} \tag{4.87}$$

with component-wise terms

$$\begin{aligned} P_j^{(t)}(\sigma_u, \sigma_v, \sigma_y, \mu_j, \mathbf{u}_j, \mathbf{v}_j) &\propto \eta_j^d : \mathcal{T}_j^d + \eta_j^m : \mathcal{T}_j^m + \eta_j^g : \mathcal{T}_j^g + \nu_j\mathcal{K}_j = \\ &= -\mathbf{E}_j^d : \mathbf{S}_j^d - \log(2\pi\sigma_u\sigma_y) - \mathbf{E}_j^m : \mathbf{S}_j^m - \log(2\pi\sigma_y\sigma_v) - \\ &\quad -\mathbf{E}_j^g : \mathbf{S}_j^g - \log(2\pi\sigma_\mu^2) + \underbrace{\kappa_\varphi}_{\eta_j^g}\,\underbrace{\mathbf{u}_j^\top \breve{\mathbf{u}}_{jg}}_{\mathcal{T}_j^g} - \log(2\pi I_0(\kappa_\varphi)), \end{aligned} \tag{4.88}$$

where the newly introduced symbols following (4.25) are

$$\mathbf{E}_j^d = \mathbf{y}_d\,\mathbf{y}_d^\top, \tag{4.89}$$

$$\mathbf{E}_j^m = \mathbf{y}_m\,\mathbf{y}_m^\top, -(2\mathbf{y}_m - \mu_j)\,\mu_j^\top, \tag{4.90}$$

$$\mathbf{E}_j^g = \breve{\mu}_g\,\breve{\mu}_g^\top - (2\breve{\mu}_g - \mu_j)\,\mu_j^\top, \tag{4.91}$$

$$\mathbf{S}_j^g = \frac{1}{2}\mathrm{diag}(\sigma_\mu^{-2}, \sigma_\mu^{-2}), \tag{4.92}$$

and $\mathcal{K}_j$ represents the number of inliers, and the background model for mean and angle is

$$\nu_j = -\log(2\pi^2\sigma_0^2) - \frac{\|\mu_j\|^2}{2\sigma_0^2}. \tag{4.93}$$

M-step (callback). The M-step is in our case not closed-form. We use an initial estimate

$$\hat{\mu}_j \approx \frac{\mathbf{y}^m}{\mathcal{K}_j}. \tag{4.94}$$

The initial estimate of φ or $(\mathbf{u}, \mathbf{v})$ is obtained by solving

$$\partial P_j(\cdot)/\partial\varphi = \left(\left(\frac{1}{2\sigma_y^2} - \frac{1}{2\sigma_u^2}\right)\mathbf{E}^d + \left(\frac{1}{2\sigma_v^2} - \frac{1}{2\sigma_y^2}\right)\mathbf{E}^m\right) : \left(\mathbf{u}\mathbf{v}^\top + \mathbf{v}\mathbf{u}^\top\right) = 0, \tag{4.95}$$

where $\mathbf{E}^m = \mathbf{A} - \mu\mu^\top$ is a symmetric matrix. The solution of (4.95) are its eigenvectors $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ giving $\hat{\varphi}_j$. The estimates of shape parameters for the given component are the eigenvalues $\hat{\sigma}_{ju}^2$ of $\mathbf{E}_j^d$ associated with eigenvector $\hat{\mathbf{u}}$ and analogically for $\hat{\sigma}_{jv}^2$ of $\mathbf{E}_j^m$ and $\hat{\mathbf{v}}$. These estimates are then combined together with shape prior resulting in

$$\hat{\sigma}_u^2 = \frac{b_u + \sum_{j=1}^k \hat{\sigma}_{ju}^2}{1 + a_u + k}, \qquad \hat{\sigma}_v^2 = \frac{b_v + \sum_{j=1}^k \hat{\sigma}_{jv}^2}{1 + a_v + k}, \tag{4.96}$$

which solves $\partial P_j(\cdot)/\partial\sigma_u = 0$ or $\partial P_j(\cdot)/\partial\sigma_v = 0$ respectively. The initial estimate of σ_y^2 is similarly computed from the other eigenvalues $\hat{\sigma}_{jy^d}^2$, $\hat{\sigma}_{jy^m}^2$ of both $\mathbf{E}^d$ and $\mathbf{E}^m$ resulting in

$$\hat{\sigma}_y^2 = \frac{b_y + \sum_{j=1}^k (\hat{\sigma}_{jy^d}^2 + \hat{\sigma}_{jy^m}^2)}{1 + a_y + 2k}. \tag{4.97}$$

In order to obtain an initial estimate of the group parameters $\breve{\theta}$ w.r.t. (4.62) the rotation center $\breve{\mu}_g$ becomes the mean of all axis midpoints in the group g

$$\breve{\mu}_g = \frac{1}{|J_g|} \sum_{j \in J_g} \mu_j. \tag{4.98}$$

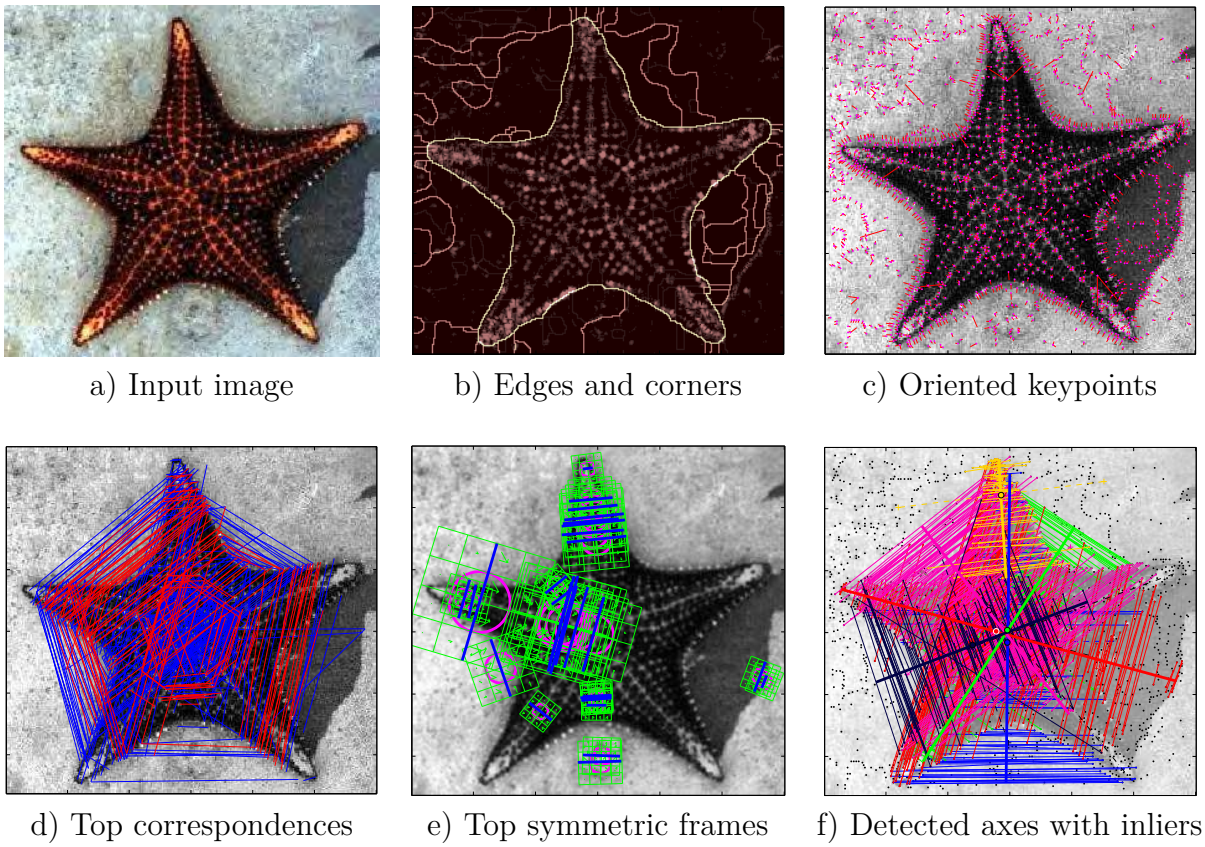
An isolated component j has identically $\breve{\mu}_g = \mu_j$. The starting angle $\breve{\varphi}_g$ is chosen to be equal to the orientation of the component φ_j in the group g such that maximizes the likelihood in (4.61) by

$$\breve{\varphi}_g \quad = \quad \arg\max_j \sum_{i \in G(g)} \log p(\varphi_i \mid \breve{\varphi}_g = \varphi_j).$$

The initial estimates are refined by regularized Newton gradient method to find the (local) optimum, which requires calculation of gradient and Hessian of the P-function w.r.t. to the parameters (callback).

4.10.8 Post-processing

After the inference endpoints of axis segment μ_j^1, μ_j^2 are chosen as intersections of the most extending correspondences $\mu^1 = x_p$ and $\mu^2 = x_q$ with the given axis $\bar{\theta}_j$.



a) Input image b) Edges and corners c) Oriented keypoints

d) Top correspondences e) Top symmetric frames f) Detected axes with inliers

Figure 4.12: Detection pipeline intermediate results, image from 2013 dataset (Liu et al., 2013)

4.11 Experimental Evaluation

4.11.1 Implementation Overview

The detection process is illustrated in Fig. 4.12:

a) The input image is acquired.

b) Edge and corner map is constructed for keypoint sampling. Contour magnitude from (Maire et al., 2008) is added to the Harris operator response (Harris and Stephens, 1988b) to sample well localized points on the image edges.

c) Up to 5000 total points are found as local maxima in the edge and corner map with non-maximum suppression search stratefy. Orientations are given by multi-scale image gradient (Sec. 4.2).

d) All keypoint pairs are sorted according to feature symmetry (Sec. 4.4.2, except self-similarity M, which is slow to evaluate on the full set) and up to $n = 5000$ tentative correspondences are selected (Sec. 4.2.4), the 100 top-ranking are shown in red, next 500 in blue.

e) Self-similarity M is evaluated for all tentative correspondences. Of those, only 100 top-ranking feature frames (patches) are shown.

f) Symmetry axes are detected with LiSAEM, shown with color-coded inliers. Axes midpoints (Sec. 4.10.8) are shown in white circle when it matches some ground truth axis (true positives). Otherwise the circle is black, in this example the axis extent (endpoints) goes beyond the required tolerance of a ground truth axis extent.

4.11.2 Hyperparameter Estimation

Although the ground truth annotations specify symmetry axes, supervised estimation of the prior parameters is not possible without assignment of keypoints and correspondences to the axes (inliers). For this purpose we have selected the inliers of a given axis based on two properties:

1. The training set annotations were enhanced with object segmentation and only keypoints within a segment were be assigned to the associated axis.

2. Only correspondences of such points with reflection error δ^y under a given threshold w.r.t. the ground truth axis are considered inliers.

The priors for parameters $\hat{\theta}, \breve{\theta}$ were then estimated by maximum likelihood fitting of the respective distributions to the inliers. Note the inference includes the parameter estimation and does not rely on exact priors.

4.11.3 Experimental Results

Evaluation of the model is based on two publicly available benchmarks.

Benchmark 2011 (Rauschert et al., 2011) contains 15 real and 15 synthetic images. Precision and recall of the results for 2011 dataset (Rauschert et al., 2011) is shown in Tab. 4.2 and Fig. 4.13. The precision/recall curve was obtained by varying the universal model $\sigma_0 \in (0.1, 1.5)$, which influences the likelihood ratio and inlier acceptance. The resulting points were connected by lines for better visibility.

We have compared the performance of our method using both static SIFT descriptors and steerable Daisy descriptors. The latter have shown to allow more accurate calculation the appearance similarity, resulting in a performance boost for some images (particularly real-world). As with other methods, real world images taken by camera are the more difficult category, and among such images the natural objects like plants are the most challenging because there is usually no exact appearance symmetry, see Fig. 4.15. In some difficult cases our method found no symmetry in the image.

The final results of our method with the objectness prior indicate the performance compared to the state of the art methods is similar for single instances, while there is a

notable increase in recall for multiple instances. Overall the other methods are outperformed approximately by 10% in precision and 20% in recall.

Dataset	Benchmark 2011 Precision / Recall				
Method	*single*	*multiple*	*synthetic*	*real*	*all*
SIFT + voting (Loy and Eklundh, 2006)	0.57 / 0.80	0.75 / 0.47	0.82 / 0.72	0.48 / 0.52	0.65 / 0.63
Contours + voting (Wang et al., 2014)	0.75 / 0.80	0.58 / 0.55	0.85 / 0.78	0.46 / 0.53	0.65 / 0.62
Daisy + Lisaem + objectness (our)	0.73 / 0.85	**0.77 / 0.81**	0.80 / 1.00	0.64 / 0.60	**0.74 / 0.82**

Table 4.2: Results on the reflection symmetry dataset (2011) as reported in the benchmark. Results of our method correspond to the optimal point on the overall curve in Fig. 4.13.

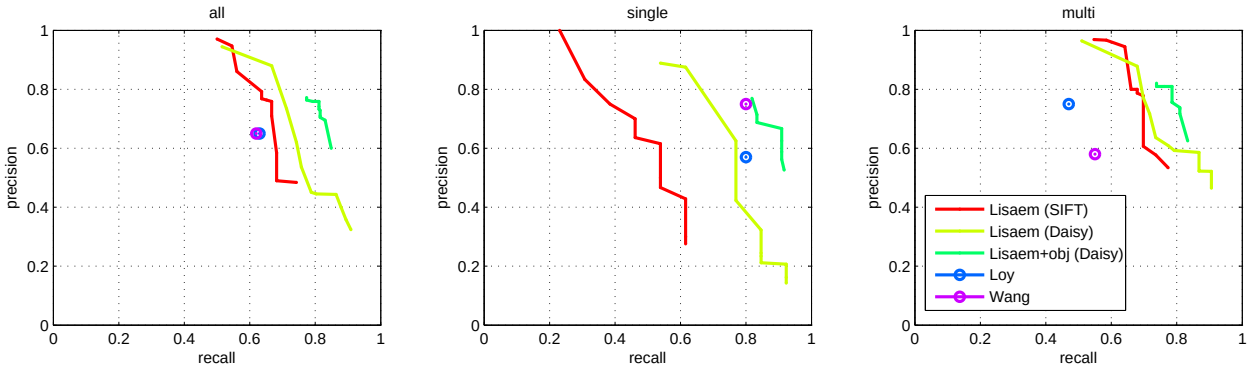


Figure 4.13: Results on the reflection symmetry dataset (Rauschert et al., 2011) as precision/recall curves (connected points) for all images, their single instance subset and multiple instance subset. Curves are not available for other methods, only single point results.

Benchmark 2013 (Liu et al., 2013) contains 70 real images. It is more challenging because there are more instances of symmetric objects with a lack of symmetry in their appearance, mostly due to shadows and occlusions, see Fig. 4.16. Many natural objects such as humans, animals or plants are symmetric only in the large scale, their texture is locally random. As a result no method competing in this benchmark was able to outperform the baseline method (Loy and Eklundh, 2006), except of Patraucean et al. (2013) for a few points on the precision/recall curve.

The precision/recall curves shown in Fig. 4.14 shows our method without the objectness prior obtains results similar to the state of the art. Our method can't benefit much from the improved inference because there are only few images with more than two axes of symmetry in this dataset.

By including a weak semantic information in the form of the objectness prior it was possible for our method to consistently achieve results above the state of the art. However,

in the case local axes of object's parts are also annotated, th objectness prior can suppress them, resulting in lower performance score. In general, by suppressing false positives the objectness prior does not allow the precision to fall under a certain limit, which results in shorter curves in Fig. 4.14.

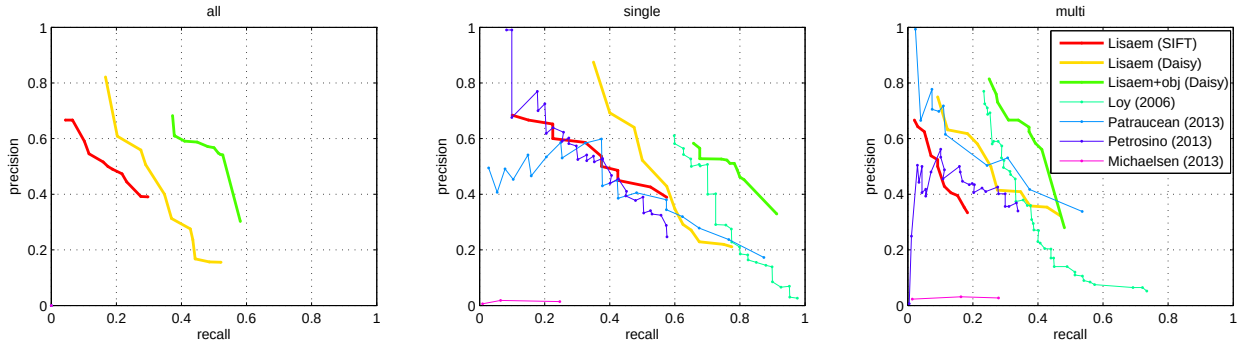


Figure 4.14: Results on the reflection symmetry dataset (Liu et al., 2013) as precision/recall curves (connected points) for all images, their single instance subset and multiple instance subset. Overall curves for other methods are not available.

4.12 Conclusion

We have shown the chosen image features and Bayesian inference method achieve better performance in detecting multiple instances of symmetry in an image. The integral cues of compactness and objectness help us to identify true reflection symmetric objects and discard local symmetries, which results in lower false positive rate when compared to other methods.

A possible extension is to implement the model also for other types of symmetries (radial, translational) and some more possible extensions in both model and inference are suggested below.

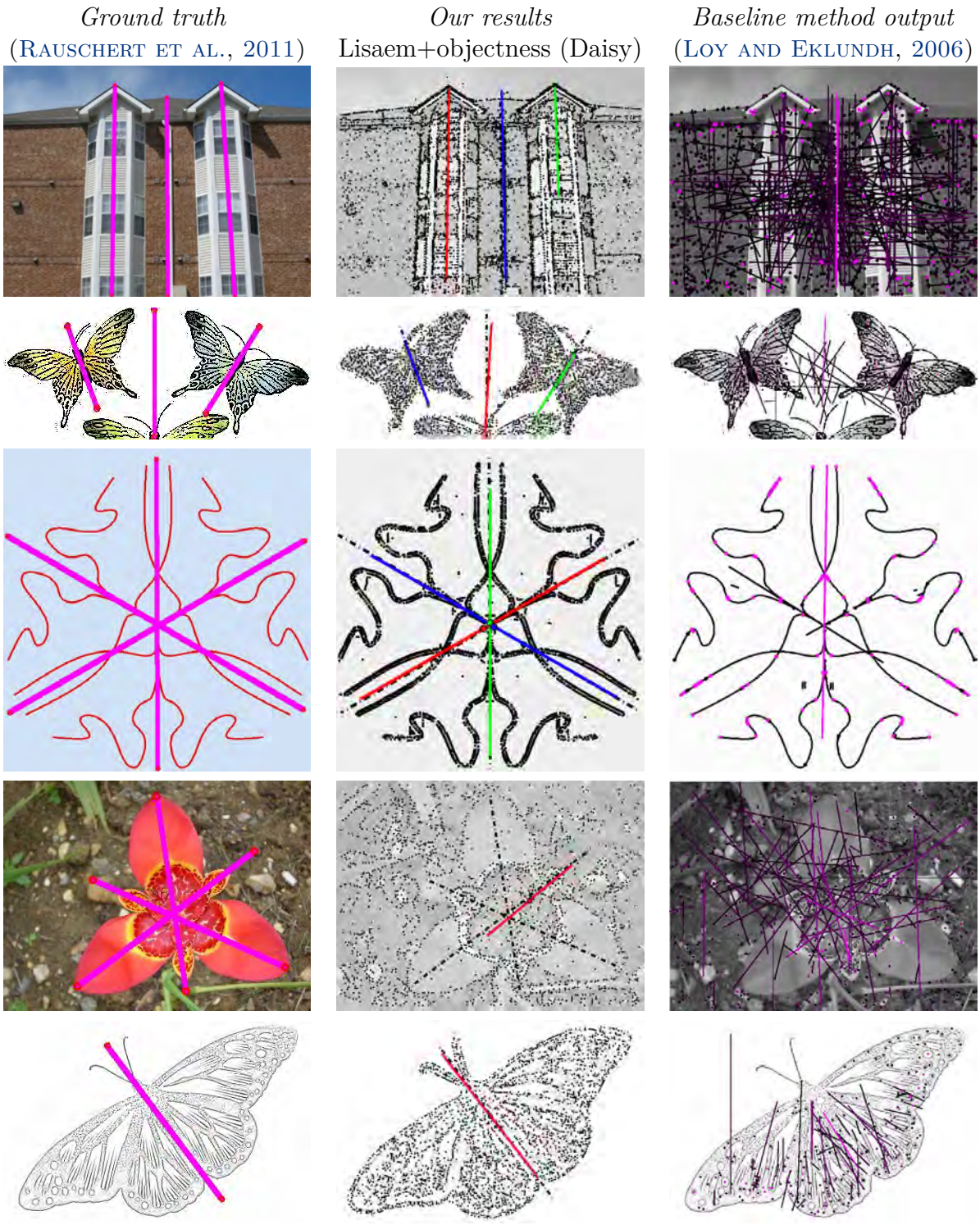


Figure 4.15: Selected images with symmetry axes and inliers from 2011 benchmark.

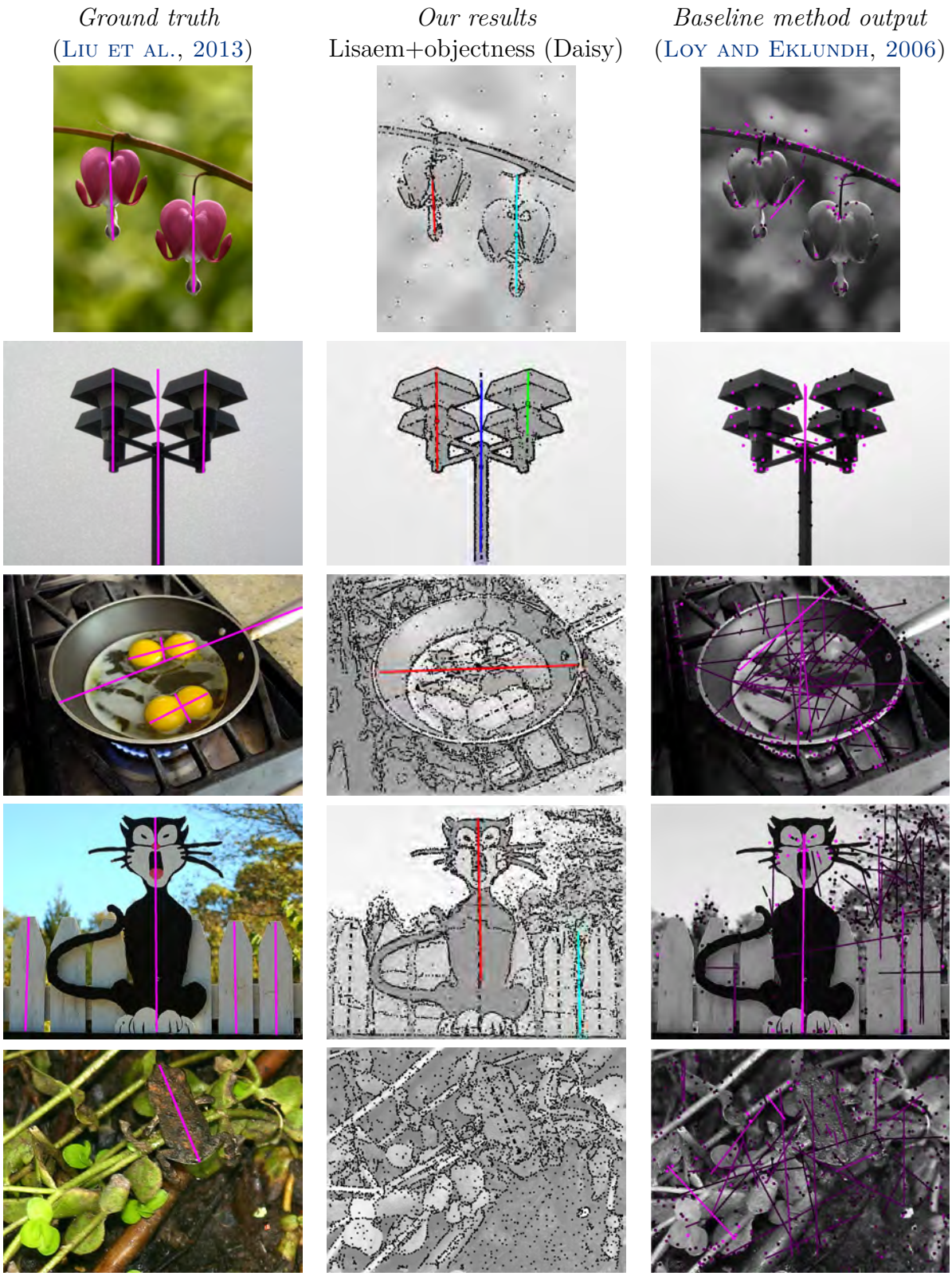


Figure 4.16: Selected images with symmetry axes and inliers from 2013 benchmark.

Chapter 5

Conclusion

"We find, therefore, under this orderly arrangement, a wonderful symmetry in the universe, and a definite relation of harmony in the motion and magnitude of the orbs, of a kind that is not possible to obtain in any other way."

Johannes Kepler (1571-1630)

In this thesis three applications of symmetry principles to computer vision problems of object detection in images were presented, focusing on the ways how our prior knowledge on translation, reflection and rotation symmetries can be encoded in probabilistic models. We followed a weak object-centered approach, which lies between general symmetry detection and strongly informed procedural modeling.

The following table summarizes proposed models and their properties:

Method	**WSM** Chapter 2	**SPT** Chapter 3	**BMRS** Chapter 4
Symmetries	Translation	Translation	Reflection, Rotation
Groups	Wallpaper	Wallpaper	Dihedral
Model type	Flat	CRF	Hierarchical
Data model	Generative	Discriminative	Hybrid
Structural model	Local	Local	Global
Inference method	MAP	Message Passing	Bayesian Model Selection
Inference algorithm	RJ-MCMC	TRW-BP	LiSAEM
Learning	Empirical	MPL	Empirical
Data point	Pixel	Pixel	Keypoint
Primitive element	Pixel	Segment	Correspondence
Component	Object	Class	Axis
Structural element	Neighborhood	N-Tuple	Group

The first two methods successfully dealt with translation symmetry in the task of facade image parsing. The *Weak Structure Model (WSM)* was our first attempt to tackle problems with variable complexity and opened the question of learning and the question of model selection.

The answer to the call for learning were *Spatial Pattern Templates (SPT)* , which facilitated learning and inference for models with a dense relation structure. Our experience suggests a tailored customization of employed general machine learning algorithms is required for further progress in this direction.

The third method aimed at *Bayesian Multiple Reflection Symmetry (BMRS)* detection. It validated the Bayesian model selection as a powerful inference mechanism for complexity estimation by producing more accurate detections when compared to the current state-of-the-art in symmetry detection.

Each method approaches the discovery of structure in the image data differently. In WSM the structure is inferred locally in terms of pairwise neighborhood, and top-level groups can be obtained as connected components, unlike in BMRS where the grouping is explicit and global. The SPT deals with structure in the learning phase, at the cost of restricting the object locations according to unsupervised image pre-segmentation.

We would like to emphasize the following contributions of this thesis:

Minimal modeling principle. We showed that probabilistic methods can be successful in reliable symmetric object detection without hard-coded domain-specific heuristics or complex features and classifiers.

Parsimony (BMRS). We found that Bayesian two-level inference does implement the Occam's razor in a mechanism that balances model complexity and error and even makes the balance data-adaptive. The result of this behavior is that the method does not oversegment in a wide variety of images (Sec. 4.10).

Model selection for complexity (BMRS). We confirmed that a rigorous estimation of the number of objects in an image (components) can be implemented with Bayesian model selection (Sec. 4.10).

Grouping priors. (WSM, BMRS). We demonstrated that principles of grouping for structural relations can be efficiently implemented by Bayesian priors. We surmise an efficient stochastic inference mechanism is needed for such models (Sec. 2.6).

Learning important relations (SPT). We showed it is possible to learn which structural relations are important and make inference more efficient and accurate (Sec. 3.3.3).

Objectness priors (BMRS). We managed to incorporate a discriminative prior for objectness in our probabilistic model. It is shown that objectness contributes to a performance increase in the multiple symmetry detection problem (Sec. 4.6).

Facade database for learning (SPT). We created a public annotated dataset sufficiently large for learning, diverse in architectural styles and of greater complexity than other datasets (Sec. A).

5.1 Possible Extensions

We see there is a potential in extending our methods in the following research directions:

Active inference strategy. Which primitives (correspondences) should be considered to detect a symmetry? Clearly, even among the top 100 tentative correspondences ranked by descriptor similarity there are just too few supporting the symmetry, for example only 10 of them match the axis of the deer's head in Fig. 5.1. A larger set could bring more support, but the number of all keypoint pairs grows quadratically with the number of keypoints and the size of their representation soon becomes prohibitive. Rather than enumerating them all not to miss some correspondence supporting the symmetry we can start with a small working subset (like in Fig. 5.1) and incrementally discover additional or more efficient correspondences during sampling with an active inference strategy. An efficient strategy must guarantee consistency of the probabilistic model and its implementation will need to revise the approach to primitive element set representation in the inference algorithm (Sec. 4.10).

Hierarchy of symmetries. Which symmetries describe the image in Fig. 5.2? The difficulty is in answering the question of which interpretation is better, symmetries of individual objects, even if the symmetry acts across the image or symmetry of symmetric objects that fills the entire image? The question is what symmetric composition of elementary symmetries describes the image best. The answer can be possibly extended from the grouping prior (Sec. 4.6) in a hierarchical model and the associated group inference mechanism.

Symmetry for saliency. Symmetry detections and measures can be used as indicators of saliency in images for a complex object recognition problems.

Integration with 3D data. Depth information associated with a given view could be included in the symmetry models as a strong additional cue. In return, a detected 2D symmetry typically indicates a symmetry in 3D as well.

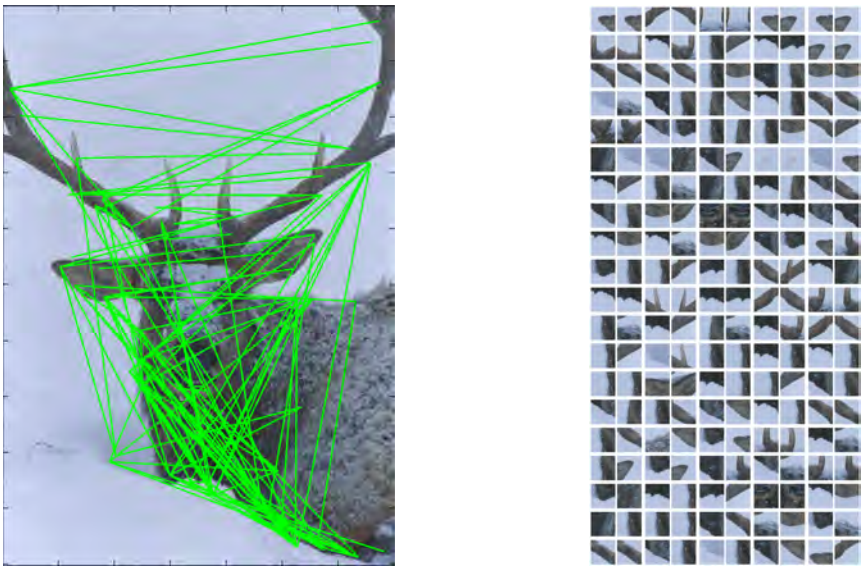


Figure 5.1: Selected tentative correspondences (left) and pairs of image patches around their keypoints (right).

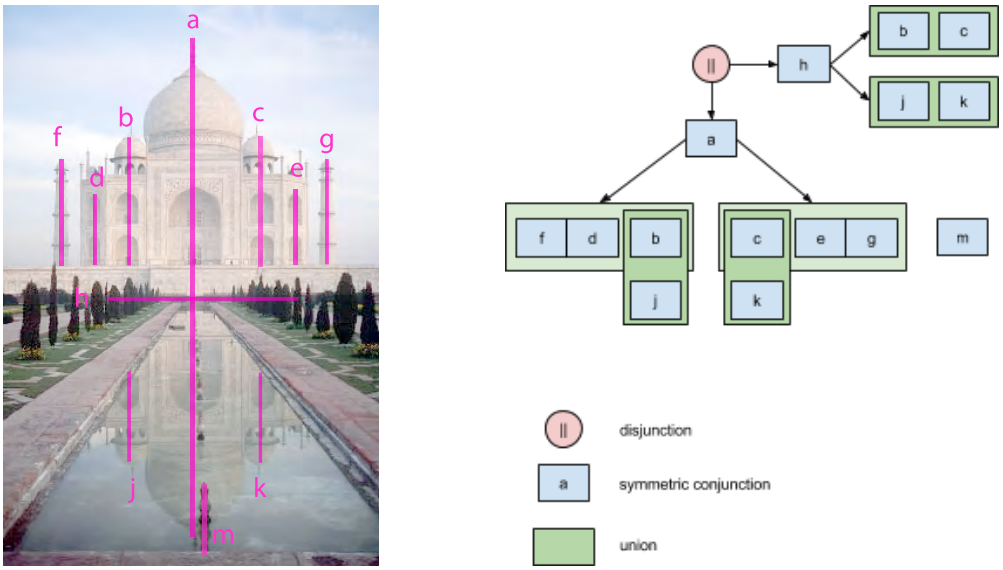


Figure 5.2: Hierarchy of symmetries in an image (left) and the associated structure (right).

Appendix A

New Facade Dataset

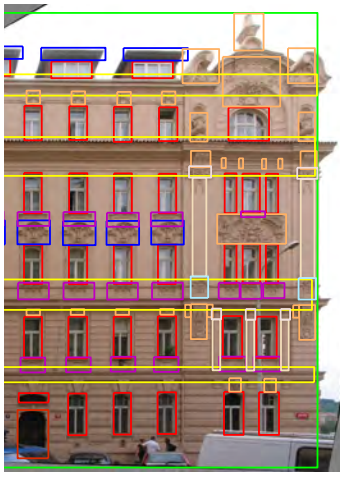
The size of annotated training set containing regular structures is the limiting factor for learning of complex relations between objects of many classes. Because of only limited data sets are publicly available, we have set up a new data set *CMP Facade Database* (Tyleček, 2012), which is sufficiently large for learning, diverse in architectural styles and allows to describe more general relative locations of objects (overlapping, nesting).

A.1 Image Data

Our dataset originates from different sources, details are provided in the following sections. Images were rectified with a method based on estimation of vanishing points from lines detected in the image, and suitably cropped afterward (does not apply to already rectified adopted images).

A.1.1 CMP-Prague

Newly presented images acquired by CMP in Prague.



Location Prague, Czech Republic

Date 2007

Camera Canon G2

Resolution ~6 MPx

Size 213 images

Source J.Šochman, R.Šára (CMP)

A.1.2 CMP-World

Newly presented images acquired by CMP worldwide.

Location Bratislava, Buenos Aires, Frankfurt, Graz, London, Ostrava, Rome, Znojmo

Date 2007-2009

Camera Various

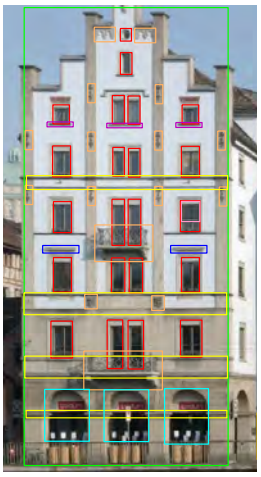
Resolution ~6 MPx

Size 99 images

Source J.Šochman, R.Šára (CMP)

A.1.3 ZuBuD

Images were adopted as a subset of unannotated Zurich Building Database.



Location Zurich, Switzerland

Date 2003

Camera ?

Resolution ~0.3 MPx

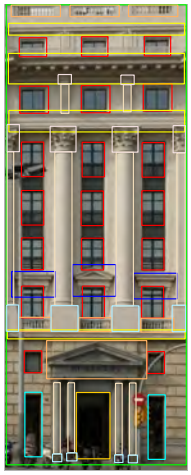
Size 177 images

Source H. Shao, T. Svoboda and L. Gool (ETH) Hao Shao and Gool (2003b)

http://www.vision.ee.ethz.ch/showroom/zubud/

A.1.4 ECP-World

Images were adopted as a subset of unannotated part of the Ecole Central Paris datasets.



Location Barcelona, Greece, Budapest, USA

Date 2010

Camera ?

Resolution ~0.6 MPx

Size 177 images

Source O. Teboul (ECP) **?**

http://vision.mas.ecp.fr/Personnel/teboul/data.php

A.2 Annotations

In this dataset *image annotation* is a set of rectangles scope with assigned class labels. Such rectangles are limited by the image scope in size and position, but otherwise they are allowed to overlap. The annotation do not necessarily explain the entire images, only objects of classes of interest are labeled. The unexplained part of the image is considered a **Background**.

A.2.1 Object classes

Dataset contains definitions for basic classes and sub-classes specified below.

Facade bounding box for a single plane wall, from pavement to roof, only complete facades are labeled, as if there is no occlusion by cars or others

Window entire glass area including borders, subtypes according to subdivision of window panes; all visible windows are annotated even if not within *Facade.*

Blind any functional obstacle to light on the window, both open or closed

Cornice decorative (raised) panel above the window

Sill decorative (raised) panel or stripe under the window

Door entrance

Balcony including railing, overlap with window when glass is visible behind

Deco any bigger piece of original art, paintings, reliefs, statues, when no other class is applicable

Molding horizontal decorative stripe across the facade, possibly with a repetitive texture pattern

Pillar vertical decorative stripe across the facade, possibly with a repetitive texture pattern, terminators (cap, base) are labeled separately

Shop shop windows, commercials, signs

A.2.1.1 Z-Order

While overlapping of object is allowed, we also sort the classes according to depth levels (Z-Order) in which the appear in the image. Rendering pixel-wise label map is then possible by sequentially painting elements according to their class labels and this order.

A.2.2 Principles

- All object annotations have *rectangular* shape.
- *Overlaps* are allowed.
- *Nesting* principle should be kept where applicable, i.e. windows inside a facade.
- Stretching of *stripes* should be to the maximal meaningful extend, i.e. side to side.
- Objects are annotated if they are not *occluded* by more than 33% of their area. Occlusion means that appearance of object borders or contents is substantially different from the expected appearance.
- The rectangle does not respect occlusions, it should reflect the occluded *reality* as much as possible. At least two opposite corners must be visible, or less if their position can be assumed from symmetry.
- Objects are annotated only on the major (rectified) plane.
- The major facade should be always labeled, additional *facades* only if their substantial part is visible.

A.2.3 Formats and Software

Annotation was performed in a custom tool for Matlab, which uses a single database file to store all annotations. Annotations are exported to the XML and PNG formats below, see Fig. A.1.

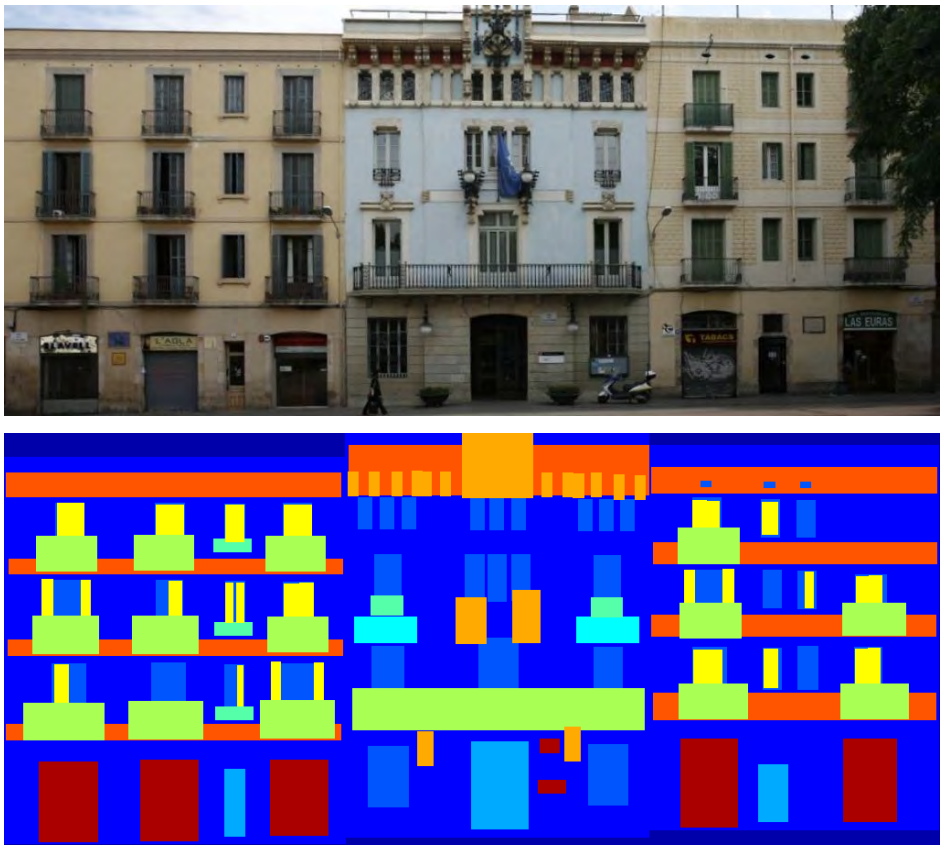


Figure A.1: Input image (top) and its annotation rendered in PNG format (bottom).

A.3 Dataset Summary

Currently we provide two datasets based on the degree of regularity present in the facades:

CMP-base planar facades with dense/strong regularity

CMP-extended irregular, non-planar and sparse facades or images with substantial occlusion from vegetation etc.

Total numbers for individual datasets are presented in Tab. A.1, which also provides comparison with previsous datasets.

Dataset	*Images*	*Objects*	*Classes*	*Avg. obj/im*	*Source*
CMP-base	378	32861	12	88	CTU
CMP-extended	228	18870	12	82	CTU
Tyleček (2012)					
Totals	**606**	**51731**			
ECP-Monge	109	?	8	?	ECP
Teboul et al. (2011)					
eTrims-8	60	1702	8	28	UBonn
Korč and Förstner (2009)					
Totals	**169**				

Table A.1: Statistics for current annotated datasets

Bibliography

Author's own bibliography is given in the following Publication List *on page 139.*

F. Alegre and F. Dellaert. A probabilistic approach to the semantic interpretation of building facades. In *International Workshop on Vision Techniques Applied to the Rehabilitation of City Centres*, 2004.

B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, June 2010.

C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4): 343–373, 2008.

Y. F. Atchadé. An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift. *Methodology and Computing in Applied Probability*, 8(2):235–254, 2006.

A. Barbu and S.-C. Zhu. Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1239 –1253, August 2005. ISSN 0162-8828.

R. Bellman and R. Bellman. *Adaptive Control Processes: A Guided Tour*. 'Rand Corporation. Research studies. Princeton University Press, 1961.

G. D. Birkhoff. *Aesthetic Measure*. Harvard University Press, Cambridge, 1932.

T. Bui-Thanh and O. Ghattas. A scaled stochastic Newton algorithm for Markov Chain Monte Carlo simulations. *SIAM Journal on Uncertainty Quantification*, 2012.

B. Calderhead and M. Girolami. Estimating bayes factors via thermodynamic integration and population MCMC. *Computational Statistics & Data Analysis*, 53(12):4028 – 4045, 2009. ISSN 0167-9473. URL http://www.sciencedirect.com/science/article/pii/S0167947309002722.

J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. ISSN 0162-8828.

J. Čech and R. Šára. Languages for constrained binary segmentation based on maximum a posteriori probability labeling. *IJIST*, 19(2):69–79, 2009.

L. Chun and A. Gagalowicz. 3D modeling of Haussmannian facades. In *Proc. of the 5th international conference on CV/CG collaboration techniques*, MIRAGE'11. Springer-Verlag, 2011.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

L. S. Davis. Understanding shape, ii: Symmetry. *IEEE Transactions on Systems, Man, and Cybernetics*, (7):204–212, 1977.

B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):94–128, February 1999.

Dempster, A.P., Laird, N.M., and Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc.*, (39):1–38, 1977.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, September 1987.

P. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal on Computer Vision*, 61(1):55–79, 2005. ISSN 0920-5691.

P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

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

W. Förstner. Optimal vanishing point detection and rotation estimation of single images from a legoland scene. In *In Proc. ISPRS Commission III Symp. Photogramm. Comput. Vis. Image Anal*, pages 157–162, 2010.

B. Fröhlich, E. Rodner, and J. Denzler. A fast approach for pixelwise labeling of facade images. In *Proceedings of the International Conference on Pattern Recognition*, pages 3029–3032, Aug 2010. doi: 10.1109/ICPR.2010.742.

C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis, Second Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2003. ISBN 9781420057294. URL https://books.google.cz/books?id=TNYhnkXQSjAC.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6 (6):721–741, Nov 1984. ISSN 0162-8828.

W. R. Gilks and G. O. Roberts. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, 1996.

J. Gips. *Shape grammars and their uses*. Birkhäuser, 1975.

R. B. Girshick, P. F. Felzenszwalb, and D. Mcallester. Object detection with grammar models. In *Proceedings of Conference on Neural Information Processing Systems*, 2011.

E. B. Goldstein. Perceiving objects and scenes: The gestalt approach to object perception. In *Sensation and perception*. Cengage Learning, 2009.

L. Gool. Projective subgroups for grouping. *Philosophical Transactions of the Royal Society of London*, A(356):1252–1266, 1998.

S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *International Journal on Computer Vision*, 80(3):300–316, 2008.

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

J. M. Hammersley and P. Clifford. *Markov fields on finite graphs and lattices*. 1971.

T. S. Hao Shao and L. V. Gool. Zubud - zurich buildings database for image based recognition. Technical Report 260, April 2003b.

C. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988b.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

J. B. Hayfron-Acquah, M. S. Nixon, and J. N. Carter. Automatic gait recognition by symmetry analysis. *Pattern Recognition Letters*, 24(13):2175 – 2183, 2003. ISSN 0167-8655. URL http://www.sciencedirect.com/science/article/pii/S0167865503000862.

J. Hays, M. Leordeanu, A. Efros, and L. Yanxi. Discovering texture regularity as a higher-order correspondence problem. *LNCS*, pages 522–535, 2006b.

B. Hohmann, U. Krispel, S. Havemann, and D. Fellner. CITYFIT: High-quality urban reconstructions by fitting shape grammars to images and derived textured point cloud. In *Proc. of the International Workshop 3D-ARCH*, 2009.

H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *International Journal on Computer Vision*, 97(2):123–147, April 2012.

S. Jain and R. Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182, 2000.

P. Kohli, L. Ladicky, and P. Torr. Robust higher order potentials for enforcing label consistency. *International Journal on Computer Vision*, 82(3):302–324, 2009.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN 0262013193, 9780262013192.

V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

F. Korč and W. Förstner. etrims image database for interpreting images of man-made scenes. Technical Report TR-IGG-P-2009-01, March 2009. URL http://www.ipb.uni-bonn.de/projects/etrims_db/.

L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative hierarchical CRFs for object class image segmentation. In *Proceedings of the International Conference on Computer Vision*, pages 739–746, 2009.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.

K. B. Laskey and J. Myers. Population markov chain monte carlo. In *Machine Learning*, pages 175–196. University Press, 2003.

S. Lee. Symmetry-driven shape description for image retrieval. *Image and Vision Computing*, 31(4):357 – 363, 2013. ISSN 0262-8856. URL http://www.sciencedirect.com/science/article/pii/S0262885613000358.

S. Lee, R. Collins, and Y. Liu. Rotation symmetry group detection via frequency analysis of frieze-expansions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587831.

J. Liu, G. Slota, G. Zheng, Z. Wu, M. Park, S. Lee, I. Rauschert, and Y. Liu. Symmetry detection from realworld images competition 2013: Summary and results. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 200–205, June 2013.

Y. Liu, W.-C. Lin, and J. Hays. Near-regular texture analysis and manipulation. *ACM Trans. Graph.*, 23(3):368–376, Aug. 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015731.

Y. Liu, H. Hel-Or, C. S. Kaplan, and L. V. Gool. Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Computer Graphics and Vision*, 5(1-2):199, 2010.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60:91–110, 2004.

G. Loy and J. Eklundh. Detecting symmetry and symmetric constellations of features. In *Proceedings of European Conference on Computer Vision*, pages 508–521, 2006.

D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. URL http://www.cambridge.org/0521642981.

M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

G. Marola. On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1): 104–108, Jan 1989. ISSN 0162-8828. doi: 10.1109/34.23119.

T. Marshall and G. Roberts. An adaptive approach to langevin mcmc. *Statistics and Computing*, 22(5):1041–1057, 2012. ISSN 0960-3174.

A. Martinović and L. Van Gool. Bayesian grammar learning for inverse procedural modeling. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

A. Martinovic, M. Mathias, J. Weissenberg, and L. J. V. Gool. A three-layered approach to facade parsing. In *Proceedings of European Conference on Computer Vision*, pages 416–429, 2012.

A. Martinovic, J. Knopp, H. Riemenschneider, and L. V. Gool. 3d all the way: Semantic segmentation of urban scenes from start to end in 3d. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June 2015a.

H. Mayer and S. Reznik. Building facade interpretation from uncalibrated wide-baseline image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(6):371–380, 2007.

E. Michaelsen, D. Muench, and M. Arens. Recognition of symmetry structure by use of gestalt algebra. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 206–210, June 2013.

B. Micusik and J. Kosecka. Piecewise planar city 3D modeling from street view panoramic sequences. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proceedings of the International Conference on Computer Vision*, pages 786–793, Jun 1995.

P. Müller, G. Zeng, P. Wonka, and L. Van Gool. Image-based procedural modeling of facades. *Transactions on Graphics*, 26(3):85, 2007.

R. M. Neal. Mcmc using hamiltonian dynamics. In S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.

W. Neiswanger, C. Wang, and E. Xing. Asymptotically exact, embarrassingly parallel mcmc. In *The Conference on Uncertainty in Artificial Intelligence*, 2014.

S. Nowozin, P. Gehler, and C. Lampert. On parameter learning in CRF-based approaches to object class image segmentation. In *Proceedings of European Conference on Computer Vision*, pages 98–111, 2010.

S. Pandolfi, F. Bartolucci, and N. Friel. A generalized multiple-try version of the reversible jump algorithm. *Computational Statistics and Data Analysis*, 72:298–314, 2014.

M. Park, K. Brocklehurst, R. T. Collins, and Y. Liu. Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1804–1816, Oct. 2009. ISSN 0162-8828.

V. Patraucean, R. von Gioi, and M. Ovsjanikov. Detection of mirror-symmetric image patches. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 211–216, June 2013.

M. Pauly, N. Mitra, J. Wallner, H. Pottmann, and L. Guibas. Discovering structural regularity in 3D geometry. *Transactions on Graphics*, 27(3), 2008.

Z. S. Qin and J. S. Liu. Multipoint Metropolis method with application to Hybrid Monte Carlo. *Journal of Computational Physics*, 172:827–840, 2001.

I. Rauschert, J. Liu, K. Brockelhurst, S. Kashyap, and Y. Liu. Symmetry detection competition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2011.

S. Richardson and P. J. Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792, 1997. ISSN 1467-9868.

H. Riemenschneider, U. Krispel, W. Thaller, M. Donoser, S. Havemann, D. Fellner, and H. Bischof. Irregular lattices for complex shape grammar facade parsing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1640–1647, June 2012. doi: 10.1109/CVPR.2012.6247857.

H. Riemenschneider, A. Badis-Szomoras, J. Weissenberg, and L. Van Gool. Learning where to classify in multi-view semantic segmentation. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Proceedings of European Conference on Computer Vision*, volume 8693 of *Lecture Notes in Computer Science*, pages 516–532. Springer International Publishing, 2014. ISBN 978-3-319-10601-4.

N. Ripperda and C. Brenner. Data driven rule proposal for grammar based facade reconstruction. *Photogrammetric Image Analysis*, 36(3/W49A), 2007.

N. Ripperda and C. Brenner. Application of a formal grammar to façade reconstruction in semiautomatic and automatic environments. In *Proc. of AGILE Conference on GIScience, Hanover, Germany*, 2009.

C. Robert. *The Bayesian Choice*. Springer-Verlag New York, second edition, 2007.

G. O. Roberts and R. L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, December 1996.

J. S. Rosenthal. Optimal proposal distributions and adaptive MCMC. In *Handbook of Markov Chain Monte Carlo*, pages 93–112. CRC Press, 2011.

B. Russell, A. Torralba, K. Murphy, and W. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal on Computer Vision*, 77(1-3):157–173, 2008. ISSN 0920-5691.

R. Šára. A general two-level bayesian inference solver for the detection of an unknown number of objects. Research Report CTU–CMP–2014–29, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, December 2014.

C. Schmid. A structured probabilistic model for recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages –490 Vol. 2, 1999.

M. Schmidt and K. Murphy. Convex structure learning in log-linear models: Beyond pairwise potentials. In *Proc. AISTATS*, 2010.

M. Schmidt, K. Murphy, G. Fung, and R. Rosales. Structure learning in random fields for heart motion abnormality detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

B. Shaby and M. T. Wells. Exploring an adaptive metropolis algorithm. *Currently under review*, 1:1–17, 2010a.

L. Simon, O. Teboul, P. Koutsourakis, and N. Paragios. Random exploration of the procedural space for single-view 3D modeling of buildings. *International Journal on Computer Vision*, 93(2), 2011.

S. N. Sinha, K. Ramnath, and R. Szeliski. Detecting and reconstructing 3D mirror symmetric objects. In *Proceedings of European Conference on Computer Vision*, Lecture Notes in Computer Science, pages 586–600. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33708-6.

O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape prior. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios. Shape grammar parsing via reinforcement learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Spring, USA, 2011.

J. Tighe and S. Lazebnik. Understanding scenes on many levels. In *Proceedings of the International Conference on Computer Vision*, pages 335–342, 2011.

E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5), 2010.

H. Wang, T.-J. Chin, and D. Suter. Simultaneously fitting and segmenting multiple-structure data with outliers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6): 1177–1192, June 2012. ISSN 0162-8828.

Z. Wang, L. Fu, and Y. Li. Unified detection of skewed rotation, reflection and translation symmetries from affine invariant contour features. *Pattern Recognition*, 47(4):1764 – 1776, 2014. ISSN 0031-3203. URL http://www.sciencedirect.com/science/article/pii/S0031320313005025.

Y. Weiss and E. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 321–326, Jun 1996. doi: 10.1109/CVPR.1996.517092.

G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. Number 27 in Stochastic Modelling and Applied Probability. Springer, Berlin, Germany, 2nd edition, 2003. ISBN 3-540-44213-8.

A. Y. Yang, K. Huang, S. Rao, W. Hong, and Y. Ma. Symmetry-based 3-D reconstruction from perspective images. *Computer Vision and Image Understanding*, 99(2):210–240, 2005. ISSN 1077-3142.

M. Yang and W. Förstner. A hierarchical conditional random field model for labeling and classifying images of man-made scenes. In *Proceedings of the International Conference on Computer Vision Workshops*, 2011.

S. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):362, 2006.

M. Zuliani, C. S. Kenney, and B. Manjunath. The multiRANSAC algorithm and its application to detect planar homographies. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 153–156. IEEE, 2005.

Publication List

Related to the thesis topic

Peer Reviewed Journals

R. Tyleček and R. Šára. Stochastic recognition of regular structures in facade images. *IPSJ Transactions on Computer Vision and Applications*, 4:63–70, May 2012. ISSN 1882-6695. URL http://www.jstage.jst.go.jp/article/ipsjtcva/4/0/4_63/_article.

ISI Excerpted Publications

R. Tyleček and R. Šára. A weak structure model for regular pattern recognition applied to facade images. In *Proceedings of the Asian Conference on Computer Vision*, volume 6492 of *Lecture Notes in Computer Science*, pages 450–463, Berlin, Germany, November 2011a. Springer. ISBN 978-3-642-19314-9. URL http://link.springer.com/chapter/10.1007%2F978-3-642-19315-6_35.

R. Tyleček and R. Šára. Modeling symmetries for stochastic structural recognition. In *Proceedings of the International Conference on Computer Vision Workshop on Stochastic Image Grammars*, pages 632–639, Piscataway, USA, November 2011b. ISBN 978-1-4673-0061-2. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6130302.

R. Tyleček and R. Šára. Spatial pattern templates for recognition of objects with regular structure. In *Proceedings of the German Conference on Pattern Recognition*, volume 8142 of *Lecture Notes in Computer Science*, pages 364–374, Heidelberg, Germany, September 2013. Springer. ISBN 978-3-642-40601-0. URL http://link.springer.com/chapter/10.1007%2F978-3-642-40602-7_39.

Other publications

R. Tyleček. The CMP606 facade database. Research Report CTU–CMP–2012–24, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, December 2012. URL http://cmp.felk.cvut.cz/~tylecr1/facade/.

Other (Unrelated) publications

Peer Reviewed Journals

R. Tyleček and R. Šára. Refinement of surface mesh for accurate multi-view reconstruction. *International Journal of Virtual Reality*, 9(1):45–54, March 2010. ISSN 1081-1451. URL http://cmp.felk.cvut.cz/ftp/articles/tylecek/Tylecek-IJVR2010.pdf.

Other publications

V. Hlaváč, R. Šára, J. Matas, T. Pajdla, J. Kostlivá, V. Franc, P. Doubek, M. Havlena, M. Jančošek, A. Torii, and R. Tyleček. Stereoscopic imaging project summary report. Research Report CTU–CMP–2009–15, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, October 2009.

R. Tyleček. Representation of geometric objects for 3D photography. Master's thesis, Department of Cybernetics, Czech Technical University in Prague, Prague, Czech Republic, February 2008.

R. Tyleček and R. Šára. Depth map fusion with camera position refinement. In *Computer Vision Winter Workshop 2009*, pages 59–66, Wien, Austria, February 2009. PRIP TU Wien. ISBN 978-3-200-01390-2. URL http://cmp.felk.cvut.cz/~tylecr1/papers/Tylecek-CVWW2009.pdf.

Citations and Authorship

The 39 known citations of the author's work are listed below in order given by their citation counts. The corresponding H-index is 4. Equal authorship is assumed for all publications with two or more authors.

Tyleček and Šára (2012), 50% authorship, SJR=0.678, 1 citation:

- Miao Jun, Chu Jun, Zhang Guimei. Window Detection Based on Constraints of Image Edges and Glass Attributes. Journal of Graphics Vol.36 No.5, China, 2015.

Tyleček and Šára (2011a), 50% authorship, SJR=0.339, 4 citations:

- Musialski, Przemyslaw, et al. A survey of urban reconstruction. Computer Graphics Forum. Vol. 32. No. 6. 2013.
- Han, Tian, et al. Quasi-regular facade structure extraction. Asian Conference on Computer Vision, 2013.
- Liu, Chun, and André Gagalowicz. 3D modeling of Haussmannian facades. IEEE Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications, France, 2011.
- Wenzel, Susanne, and Wolfgang Förstner. Learning a Compositional Representation for Facade Object Categorization. International Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I 3 (2012): 197-202.

Tyleček and Šára (2011b), 50% authorship, SJR=1.893, 4 citations:

- Teboul, Olivier, et al. Parsing facades with shape grammars and reinforcement learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35.7 (2013): 1744-1756.
- Han, Tian, et al. Quasi-regular facade structure extraction. Springer Berlin Heidelberg, 2013.
- Pedro, Ricardo Wandré Dias, Fátima LS Nunes, and Ariane Machado-Lima. Using grammars for pattern recognition in images: A systematic review. ACM Computing Surveys (CSUR) 46.2 (2013): 26.
- Pedro, Ricardo Wandre Dias. Inferência de gramáticas estocásticas para reconhecimento de padrões de imagens utilizando quadtrees. PhD Thesis. Universidade de São Paulo.

Tyleček and Šára (2013), 50% authorship, SJR=0.339, 5 citations:

- Kushnir, Maria, and Ilan Shimshoni. Epipolar geometry estimation for urban scenes with repetitive structures. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36.12 (2014): 2381-2395.
- Jampani, Varun, Raghudeep Gadde, and Peter V. Gehler. Efficient Facade Segmentation Using Auto-context. IEEE Winter Conference on Applications of Computer Vision, 2015.

- Mathias, Markus, Martinović, Anđelo and Luc Van Gool. ATLAS: A Three-Layered Approach to Facade Parsing. International Journal of Computer Vision (2015): 1-27.
- Martinović, Anđelo et al. 3D All The Way: Semantic Segmentation of Urban Scenes From Start to End in 3D., IEEE Conference on Computer Vision and Pattern Recognition, 2015.
- Lian, Yongjian, and Xukun Shen. Probabilistic reasoning for repeatability detection from urban facade image. International Conference on Optical and Photonic Engineering, 2015.

Tyleček (2012), 100% authorship, 1 citation:

- Gadde, Raghudeep, Renaud Marlet, and Paragios Nikos. Learning grammars for architecture-specific facade parsing. Research Report, INRIA (2014).

Tyleček and Šára (2010), 50% authorship, 9 citations:

- Vu, Hoang-Hiep, et al. High accuracy and visibility-consistent dense multiview stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34.5 (2012): 889-901.
- Li-Min Shi, Fu-Sheng Guo, Zhan-Yi Hu. An Improved PMVS through Scene Geometric Information. Acta Automatica Sinica Vol. 37, No. 5. May, 2011.
- Hu, Xiaoyan, and Philippos Mordohai. Least Commitment, Viewpoint-Based, Multi-view Stereo. IEEE Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012.
- Graber, Gottfried, et al. Efficient Minimal-Surface Regularization of Perspective Depth Maps in Variational Stereo. IEEE Conference on Computer Vision and Pattern Recognition, 2015.
- Zheng, Enliang, et al. Patchmatch based joint view selection and depthmap estimation IEEE Conference on Computer Vision and Pattern Recognition, 2014.
- Kannala, Juho, et al. Multi-View Surface Reconstruction by Quasi-Dense Wide Baseline Matching. Emerging Topics in Computer Vision and Its Applications 1 (2011): 403.
- Vu, Hoang Hiep. Large-scale and high-quality multi-view stereo. PhD Thesis. Paris Est, 2011.
- Miao, Jun, Jun Chu, and Guimei Zhang. Disparity map optimization using sparse gradient measurement under intensity-edge constraints. Signal, Image and Video Processing (2014): 1-9.
- Balzer, Jonathan, and Stefano Soatto. Second-order Shape Optimization for Geometric Inverse Problems in Vision. arXiv:1311.2626 (2013).

Tyleček and Šára (2009), 50% authorship, 15 citations:

- Hiep, Vu Hoang, et al. Towards high-resolution large-scale multi-view stereo. IEEE Conference on Computer Vision and Pattern Recognition, 2009.

- Vu, Hoang-Hiep, et al. High accuracy and visibility-consistent dense multiview stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34.5 (2012): 889-901.
- Tola, Engin, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. Machine Vision and Applications 23.5 (2012): 903-920.
- Jančošek, Michal, Alexander Shekhovtsov, and Tomáš Pajdla. Scalable multi-view stereo. IEEE Conference on Computer Vision Workshops, 2009.
- Salman, Nader, and Mariette Yvinec. Surface Reconstruction from Multi-View Stereo of Large-Scale Outdoor Scenes. International Journal of Virtual Reality 9.1 (2010): 19-26.
- Koskenkorva, Pekka, Juho Kannala, and Sami S. Brandt. Quasi-dense wide baseline matching for three views. IEEE Conference on Pattern Recognition, 2010.
- Hu, Xiaoyan, and Philippos Mordohai. Least Commitment, Viewpoint-Based, Multi-view Stereo. IEEE Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012.
- Labatut, Patrick. Labeling of data-driven complexes for surface reconstruction. Phd Thesis. Université Paris-Diderot-Paris VII, 2009.
- De Cubber, Geert. Variational methods for dense depth reconstruction from monocular and binocular video sequences. Phd Thesis. Image Processing and Machine Vision Group (IRIS), Vrije Universiteit Brussel, 2009.
- De Cubber, Geert, and Hichem Sahli. Partial differential equation-based dense 3D structure and motion estimation from monocular image sequences. IET Computer Vision 6.3 (2012): 174-185.
- Hu, Xiaoyan, and Philippos Mordohai. Robust Probabilistic Occupancy Grid Estimation from Positive and Negative Distance Fields. IEEE Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012.
- Joubert, Daniek, and Willie Brink. A mesh-based approach to incremental range image integration. Annual Symposium of the Pattern Recognition Association of South Africa. 2011.
- Galindo, Patricio, and Rhaleb Zayer. Distortion driven variational multi-view reconstruction. IEEE Conference on 3D Vision (3DV), 2014.
- Vu, Hoang Hiep. Large-scale and high-quality multi-view stereo. PhD Thesis. Paris Est, 2011.
- Jančošek, Michal. Large Scale Surface Reconstruction based on Point Visibility. PhD Thesis. Czech Technical University, 2014.