SHAPE REPRESENTATIONS FOR OBJECT RECOGNITION

Alexander Toshev

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2011

Kostas Daniilidis
Professor of Computer and Information Science
Supervisor of Dissertation

Ben Taskar
Assistant Professor of Computer and Information Science
Co-Supervisor

Jianbo Shi
Associate Professor of Computer and Information Science
Graduate Group Chairperson and Co-Supervisor

Dissertation Committee:

Camillo J. Taylor, Associate Professor of Computer and Information Science

Daniel D. Lee, Associate Professor of Electrical and Systems Engineering

Pietro Perona, Professor of Electrical Engineering and Computation and Neural Systems

Longin Jan Latecki, Professor of Computer and Information Sciences

Shape Representations for Object Recognition

Acknowledgements

First and foremost I would like to thank my advisors. Prof. Kostas Daniilidis is a great mentor, who was always supportive in my research endeavors and has thought me a lot about computer vision and research in general. I have learned immensely from the insightful thinking and unique scientific approach of Prof. Jianbo Shi. I would like to thank Prof. Ben Taskar for all the inspiration, energy and support during the last three years of my work. I owe special thanks to my dissertation committee members for the review of my work: Prof. CJ Taylor, Prof. Daniel Lee, Prof. Pietro Perona, and Prof. Longin Jan Latecki.

This work would not have been possible without all the great people at the GRASP lab. I would like to thank Prof. Philippos Mordohai for the continuing support and the motivation he has given me during his tenure at Penn. I have learned a lot from Ameesh Makadia who has given me the unique opportunity to work at Google. Many thanks to Ben Sapp for the great collaboration and the great time we have had together.

Although not directly involved in my work, I owe special thanks to many of my colleagues for the inspiring discussions, the nurishing scientific environment and the time spent in and outside the lab: Timothee Cour, Qihui Zhu, Praveen Srinivasan, Gang Song, Katerina Fragiadaki, Sandy Patterson, Elena Bernardis, Qihui Zhu, Joao Graca. I was lucky to enjoy the company and great conversations with my cubicle mates and friends Arvind Bhusnurmath, Babak Shirmohammadi, David Weiss, Paul Vernaza, Berkay Deniz Ilhan, Allison Mathis, Michael Zavlanos. This work would

not have been possible without the priceless administrative support of Mike Felker and Charity Payne.

Thank you all for making my doctoral studies at Penn and my stay in Philadelphia such an enriching and precious experience!

ABSTRACT

SHAPE REPRESENTATIONS FOR OBJECT RECOGNITION

Alexander Toshev

Kostas Daniilidis

The problem of object recognition has been at the forefront of computer vision research in the last decade. The most successful approaches have used mainly edge- or texture-based representations. The shape of the object outline, albeit widely used for pre-segmented objects, has found limited applicability to the detection problem in real images. The fact that shape is a truly holistic global percept is challenging because background structure and interior object contours can easily clutter a global shape descriptor and render it unusable. Therefore, figure-ground organization, which segments the object of interest and removes the cluttering contours, is of paramount importance. However, purely bottom-up segmentation rarely provides a good object outline suitable for shape-based detection.

In this thesis, we study a novel shape representation, called a chordiogram, which allows us to address the above challenges. The chordiogram is a holistic shape descriptor capturing global geometric relationships between object boundaries. Based on the chordiogram, we introduce a boundary structure segmentation model which efficiently integrates region and boundary grouping principles with shape-based matching. This method uses holistic shape for simultaneous object segmentation and detection in highly cluttered scenes. We apply it on established recognition benchmarks and achieve state-of-the art results.

Further, we study the applicability of shape for object detection in videos. We show that shape-based representations can be used not only to robustly detect moving objects but also to provide a rough estimate of their pose. For this purpose, we utilize freely available large datasets of 3D synthetic models.

Beyond linking shape matching with perceptual grouping, we study the interplay between feature matching and perceptual grouping. We introduce co-salient

regions – coherent, corresponding segments in two or more images – and describe two algorithms for their detection. Co-salient regions are applied to two problems – wide-baseline stereo and motion segmentation. In the former problem we show how to estimate correspondences between regions and improve feature matches, while in the latter segments representing same object parts are tracked across multiple frames in a video.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the basic and most fundamental problems being addressed in computer vision is the problem of object recognition and detection. It is the question of what objects or object types we see in an image and how to determine their precise location and segmentation. Answering this question with high accuracy will have an enormous impact and diverse consequences in the field of machine perception and beyond. It is a fundamental step towards image and video understanding.

Over the last two decades computer vision research has focused an enormous effort on solving this problem. Although we have witnessed great progress, especially in the last ten years, computer performance still remains unsatisfactory. One of the unresolved aspects of these challenges is the proper representation which should be used in order to describe an object. Researchers have proposed a variety of representations capturing various object properties.

In this thesis we analyze shape for the purpose of recognition. We provide a new perspective on how to operationalize shape and develop a novel representation and a computational model. In support of this model, we provide empirical evidence with respect to established benchmarks of still images as well as videos.

1.1 Representations for Object Recognition

In the last ten years a myriad of object representations have been proposed. The most widely used are keypoint features [Lowe, 2004, Mikolajczyk and Schmid, 2004, Tuytelaars and Gool, 2004]. These descriptors are based on edges and tend to capture local fine structure. Their success owes to the fact that they model local object parts in a discriminative and repetitive way [Zhang et al., 2007b]. This enables the features to capture salient object parts while dealing at the same time with clutter and occlusion.

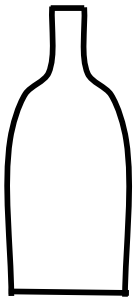
These local features have been thoroughly analyzed for the task of recognition [Zhang et al., 2007b]. They have been used individually as well as they have been combined in recognition frameworks. The consensus among researchers is that no single feature alone is sufficient for object recognition. The reason is that each feature captures specific object properties, which are manifested differently for objects from different classes. This realization has motivated the exploration for further representations.

Examples of other representations, which capture fine image structure, are the texture descriptors based on filter bank responses [Varma and Zisserman, 2005], [Leung and Malik, 2001]. They have been applied to object recognition [Malisiewicz and Efros, 2008], [Gu et al., 2009] as well as scene segmentation at a global scale [Shotton et al., 2009].

Later, edge-based descriptor have been proposed, which have a support going beyond an object part and covering the whole object. For example, Histogram of Gradients has been applied for both human detection [Dalal and Triggs, 2005] and general object recognition [Felzenszwalb et al., 2008].

Although texture- and edge-based descriptors are successful for representing objects, there are object classes which inherently can be only poorly described in this manner. For example, if we study the examples of a specific object class, as shown in Fig. 1.1, it is easy to see that neither color nor texture is very repeatable across

(a) Examples of bottles.



(b) Bottle shape model.

Figure 1.1: What are the common properties among these examples of bottle?

these examples. The only property, which seems to be shared is the shape.

1.2 Shape Representations

In this thesis we focus on shape, which is generally defined as the outline of an object. In some cases, shape includes internal contours as well. From perceptual point of view, there have been two major paradigms for capturing object shape, which we describe below.

1.2.1 Holism

The Gestalt school of perception, led by Max Werhheimer, Wolfgang Koehler and Kurt Koffka, has established the idea of *holism* in visual perception [Palmer, 1999, Koffka, 1935]. This principle suggests an object should be perceived as a whole and not merely as a collection of individual parts. Although this paradigm was formulated for general perception, it is even more applicable to shape. The main reason is that shape is locally not discriminative – while a small region in an image may contain a rich and complex texture pattern, a contour segment in a small region is much less informative.

As a result, when it comes to the design of a computational model for a holistic shape representation we try not to identify local contours as a means for description, but attempt to describe each object contour in the context of the whole object. In other words, the contribution of an edge or a contour segment to the whole object representation depends on all other object contours. We call this view on holism *global dependence*.

There have been a wide range of shape representations consistent with the principle of holism. Such representations are based on a global transform of the input shape. For example, Fourier coefficients have been used to characterize a closed contour [Zhang and Lu, 2003]. Similarly, Zernicke moments were applied to capture shape contour and interior [Zhang and Lu, 2003]. Smoothing a curve and detecting curvature zero-crossings led to Curvature Scale Space Image [Mokhtarian et al., 1997].

Another class of holistic shape representations was initiated by the development of the Medial Axis Transform [Blum, 1973], which is defined as the set of centers of maximally inscribed circles in a closed shape. This set can be thought of as a skeleton of the shape, which is computed globally, and reveals geometrical as well as topological shape properties. Depending how those properties are captured, the medial axis has led to the development of Shocks, Shock graphs [Kimia et al., 1995, Siddiqi et al., 1999, Sebastian et al., 2004] as well as M-reps applied to medical imaging [Pizer et al., 1999]. To deal with the instability of the medial axis to small boundary protrusions a more robust transform based on the Poisson equation has been proposed [Gorelick and Basri, 2009].

Unfortunately, the above holistic representations have had limited success when it comes to real scenes containing background clutter and multiple objects. The main reason is that holism implies a global support. Therefore, holistic representations are susceptible to clutter. As a result, the above approaches expect an already segmented shape as an input, which is not always a realistic setting, especial for real world images.

1.2.2 Structuralism

A different paradigm for object representation is the assumption that each object can be decomposed and this is described in terms of a set of atoms, chosen from a small atom dictionary. Due to the nature of shape, which cannot be easily described locally, the aforementioned atoms should be the shape primitives which capture semi-local shape parts.

From a historical perspective there have been quite a few principled structural theories for shape perception. One type of shape primitive is the generalized cylinder which can be described as a base being swept along an axis [Marr, 2010, Binford, 1971]. By combining generalized cylinders with different parameters one can generate a wide range of objects. Another theory based on a discrete set of predefined 3D shape primitives, called geoms, is the Recognition by Components Theory [Biederman, 1987]. Another attempt is superquadrics which represent more natural shapes
[Pentland, 1986].

The above theories were not successfully applied to real-world vision problems. The main challenges are twofold. On one side, the theories assume that one can extract the shape primitives in images, which is not always the case. On the other side, even if one can obtain good primitive candidates from an image, the search for the correct shape is not always straightforward and tractable
[Grimson and Lozano-Perez, 1987].

1.2.3 Computational Models of Shape

The last decade has witnessed a significant development in shape representations. Most of the proposed approaches lie somewhere between being holistic and structural. On one side, they try to capture global properties while being invariant to shape deformations. On the other side, some of the global properties have to be sacrificed for the sake of having a tractable inference and dealing with problems posed by real images: clutter and occlusion.

Most of the shape representations are defined in terms of *tokens* or landmarks, which can be very local *points*, or semi-local *contours*. The *configuration* of such tokens can be captured in a purely *statistical* fashion or using a *template*. Usually such configurations express the structural organization of the tokens in the plane, but also exploit the fact that these points are *linearly ordered* along a curve, in cases where a curve parameterization is provided. Finally, the support of the shape representation can be *semi-local* or *global*.

Template-based methods capturing unordered point sets. The study of geomertic configurations of landmarks for describing shape has been pioneered by [Kendall, 1989]. Early template-based methods for describing point configurations are the Chamfer distance [Borgefors, 1986] and the Hausdorff distance
[Huttenlocher et al., 1993]. Such simple techniques are not holistic and are susceptible to clutter. A richer set of geometric features in conjunction with graph matching techniques have been used by [Leordeanu et al., 2007] to match an edge configuration to a template. A parametric statistical framework, which models the shape deformation of the point set is the Active Shape Model [Cootes, 1995].

Template-based methods capturing linearly ordered point sets. The above methods capture an unorganized point set. In many cases, however, one is provided with a linear grouping of the points in a curve [Zhu et al., 2007]. This linear ordering has been exploited in the design of algorithms which align curves [Sebastian et al., 2003, Felzenszwalb and Schwartz, 2007],
[Mcneill and Vijayakumar, 2006]. Such methods rely on perceptual contour grouping to obtain object contours. However, they may not capture the full object outline.

Statistics-based methods capturing point sets. A principally different way to capture the configuration of a set of points is to capture their statistics. For example, geometric hashing has been used to describe purely geometric properties

[Lamdan et al., 1990] as well as topological properties [Carlsson, 1999] at a global scale. A very successful descriptor, called Shape Context [Belongie et al., 2002] captures a semi-local distribution of edges. Its descriptive power has been extended to more deformed and articulated shapes by [Ling and Jacobs, 2007]. A different work has related the Shape Context to contour grouping and based on a holistic matching model was designed [Zhu et al., 2008, Srinivasan et al., 2010].

Contour configurations. A different type of token are contour segments. They provide richer information than individual edges and can be extracted using edge linking, contour grouping or segmentation. Boundary fragments combined with a voting scheme have been applied to object recognition [Opelt et al., 2006], [Shotton et al., 2005]. [Ferrari et al., 2006] search in a contour network for contour chains which resemble the model. In a subsequent work, [Ferrari et al., 2008] define a descriptor for groups of adjacent contour segments and use it in conjuction with an SVM classifier. [Lu et al., 2009] explore particle filtering to search for a set of object contours. Dynamic programming has been applied also by [Ravishankar et al., 2008] in a mutli-stage framework to search for a chain of object contours.

The above approaches use semi-local tokens and combine them using global models. These models search for a matching configuration in a scene and thus deal with clutter. To make the search tractable, these models usually do not take into account all dependences among contours and thus lose some of the global relationships between contours. In addition, the above approaches recover some object contours, however none of them reasons over regions or attempts to recover full figure/ground organization.

A selection of the above approaches are compared with respect to holism and robustness to clutter in Fig. 1.2.

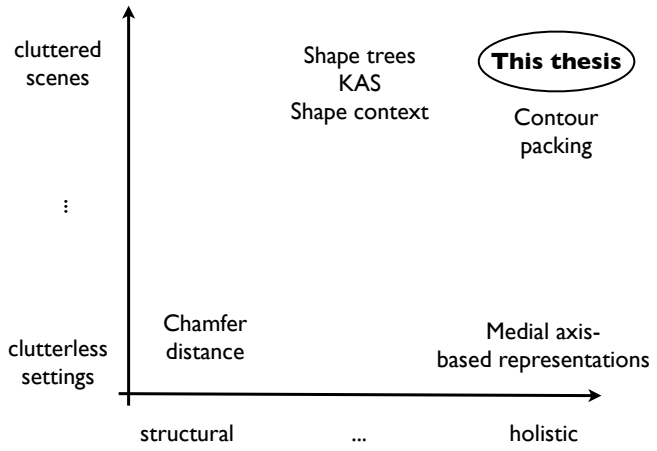


Figure 1.2: Representative selection of shape representations and their properties with respect to holism and robustness to clutter. The approach, presented in this thesis, uses a holistic shape descriptor which targets highly cluttered scenes.

1.3 Contributions of the Thesis

This thesis addresses the question of how to represent shape for the purpose of describing, segmenting, and detecting objects in still images as well as videos. We focus on designing a shape descriptor and integrating it together with grouping principles into an object segmentation and detection model. More precisely, the contributions of this work are as follows:

Shape representation: We introduce a novel shape descriptor, called a *chordiogram*, which captures the object outline and its interior. It possesses the following properties, whose combination is novel and contributes to the descriptor's state-of-the-art performance:

- It captures the object shape as a whole and therefore falls into the category of holistic descriptors.
- The definition of the chordiogram enables one to relate the descriptor to

region and boundary grouping in the image.

By combining both properties, we can simultaneously segment and match an object using its shape. This allows us to deal with clutter and apply the descriptor to real scenes.

In addition, it is worth mentioning that the chordiogram is efficiently computable and thus can be used to obtain efficiently a similarity score between two shapes. However, the introduced descriptor can be related to graph matching and, if necessary, can be used to obtain correspondences between points on the shapes.

Perceptual grouping: We show how to use perceptual grouping to improve matching in two computer vision settings:

- We are able to incorporate both region and boundary grouping principles into a holistic shape detection approach.
- We introduce the notion of co-salient regions, defined as coherent segments from two or more images, which exhibit strong appearance similarity. By detecting co-salient regions in pairs of images, we are able to improve local feature matches in the context of wide-baseline stereo.

In addition, we extend the the estimation of co-salient regions to multiple images and apply it to video segmentation.

Object detection: We address the problem of shape-based object detection in still images and videos.

- We introduce a shape detection model, called BoSS, which applies the chordiogram. It allows for a simple and yet effective approximate computational solution. BoSS is shown to achieve state-of-the art performance on several established benchmarks.

- We utilize datasets of synthetic 3D models, freely available on the web, for analysis of videos. We present a framework, which uses those models to detect and provide a rough pose estimate for moving objects in videos.

1.4 Thesis Outline

This thesis is structured as follows:

Chapter 2: We motivate and introduce the chordiogram. The properties of the descriptor are analyzed. In particular, we provide two versions of the descriptor, each of which is invariant to different transformations – translation and rotation. Moreover, we provide a theoretical relationship between chordiogram and graph matching. In addition, an empirical analysis is presented and the descriptor is compared to other representations.

Chapter 3: We introduce the **Bo**undary **S**tructure and **S**egmentation Model (BoSS), which combines the chordiogram matching with perceptual grouping principles. Both region and boundary grouping principles are explained. An efficient and simple approximate optimization is described which is based on the Semidefinite Programming relaxation. Finally, we empirically analyze and compare the performance of the model on several datasets.

Chapter 5: The notion of co-salient regions is motivated and defined. We provide a framework for their estimation and two concrete algorithms which can be used for two applications. The first one is wide-baseline stereo, in which we simultaneously detect corresponding regions and improve feature matches between pairs of images. In the second application, we segment video by processing all video frames jointly – a co-salient region in this application is a segment representing a part of a scene which is being tracked across multiple frames.

Chapter 4: We turn our attention to object detection in videos. For this purpose,

we use shape matching between 3D models and objects in videos. We discuss how to extract a shape representation of a 3D model suitable for matching to an image as well as how to use motion to extract an object silhouette. We present results on videos of moving objects using the chordiogram as well as shape context as a representation.

Chapter 2

Shape Representation

When it comes to visual perception, there are many cues one can use to describe an object, such as color, texture, shape, etc. It is natural to ask which one carries the most information, i. e. which one alone can be sufficient to identify and characterize an object. Consider for a moment a swan – describing it by its color, which can be either black or white, may not suffice to identify it, since there are many objects which share this property. The swan shape, however, defined as a simple drawing of the outline, seems to be enough to recognize this object with high certainty. This examples motivates us to consider shape as one of the most powerful cues for visual perception [Palmer, 1999] and a natural basic level of abstraction of an object category [Rosch et al., 1976]. This observation is supported by empirical evaluation and comparison of different cues on established datasets [Gu et al., 2009].

In this chapter we lay the groundwork for our shape-based approach towards object detection. After introducing the basic principles, that we believe a shape representation should satisfy, we formulate our descriptor and its variants. Further, analysis of its properties, relations to other representations, and performance evaluation is presented.

2.1 Properties of a Shape Representation

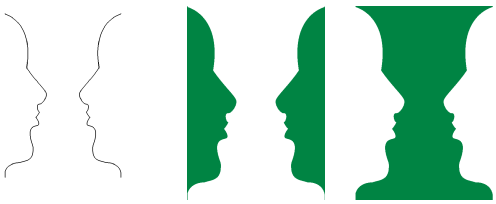
In this work we capture the shape of the object by describing the object outline and its relation to the object interior:

Object boundary: The object boundary can be considered to be a closed curve. This is based on the assumption that the object is completely included in the image and is not occluded. In real images, though, this is assumption is often violated and thus robustness to occlusion is a desirable property of any shape representation.

Contour-based shape descriptions are widely used in computer vision. Some of them, such as Shape Context [Belongie et al., 2002], k-adjacent segments [Ferrari et al., 2008], and hierarchical shapes [Felzenszwalb and Schwartz, 2007], can capture an open curve and thus are suitable to describe a portion of the object outline. Others, such as shock graphs [Siddiqi et al., 1999], [Sebastian et al., 2004], can be defined only for closed curves.

Interior: Although the object boundary carries the main shape information of an object, it is important to capture also the object interior. To understand this consider the example in Fig. 2.1 of the Rubin's vase [Rubin, 1915] which allows for two interpretations – two faces or a vase – if one observes the image contours. To resolve this ambiguity the object interior needs to be selected. To address this problem, some of the contour-based descriptors, such as the ones based on the medial axis transform, relate the object boundary to the object interior [Blum, 1973, Siddiqi et al., 1999, Sebastian et al., 2004].

In this work we aim at capturing the object boundary as well as its interior. In addition, we propose a descriptor which does not require a closed curve as an input and thus can deal with occlusions and partial views.



(a) Image contours (b) Face interpretation (c) Vase interpretation

Figure 2.1: Rubin's vase allows for two interpretations with the same set of image contours.

Holism. In this work, we follow the view of the Gestalt school of perception. According to the principles of Gestaltism, objects are perceived as a whole. This notion is best captured in the words of Koffka [Koffka, 1935] (p. 176): 'The whole is different from the sum of its parts' and inspires the use of *holistic* object representations. Evidence for this principle are the so called emergent properties of part configurations [Palmer, 1999] – a configuration of parts can have properties, which are not contained in the individual parts. For example, in Fig. 2.1 the centaur, which has a horse and a human part is neither a horse nor a human but a completely new creature. This interpretation is not only based on the properties of the individual parts (human upper body and horse torso in this case), but mainly on the interaction among the parts (a human upper body being attached on top of a horse torso).

The holistic approach is especially true for shape, where local parts are not descriptive since they consist of simple contour segments. Contrary to edge- and texture-based local descriptors, for which a patch in the image can contain rich information, shape has locally very limited expressiveness – usually it is locally a curve, potentially with a few high curvature points. For example, in Fig. 2.1 the locally salient corners carry virtually no discriminative information to differentiate between the two objects.

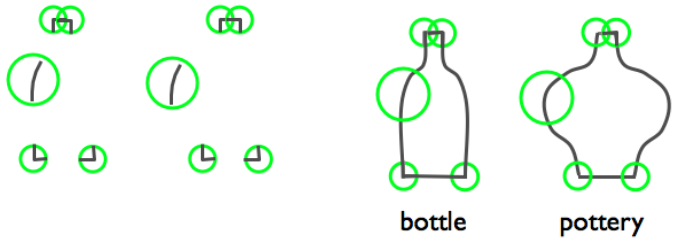
If we try to increase the support of the local parts, we can see that they become expressive only if they capture a big portion of the outline. At this global

Figure 2.2: A creature which is part human and part a horse is a completely new creature.

level, however, at which part descriptors highly overlap, we can observe global interactions among parts – we can no longer assume independence in the appearance of the different parts, as it is commonly done for part-based approaches. Hence, *global dependence* among all shape parts emerges and this is what we will regard computationally as holism.

The above interpretation of holism as global dependence among all shape parts does not give a direct prescription for the design of shape descriptors, but can only serve as a high-level motivation. From a computational standpoint, a holistic descriptor should not merely add up the evidence from the individual object contour segments, but compute properties which are a non-additive function of all or most of the contour segments. For example, the Fourier contour descriptor [Zhang and Lu, 2003] consists of spectral coefficients which depend on the whole object outline. Similarly, the medial axis [Blum, 1973], which is the basis for shocks and shock graphs, depends on the whole object outline.

A major drawback of such holistic descriptors is that they can easily be affected by clutter. For non-segmented objects, background structure in the image and interior contours may clutter the descriptor and thus severely undermine its representational power. This has limited the applicability of the aforementioned representations to object recognition benchmarks.



(a) Local features' view (b) Boundaries of both objects

Figure 2.3: Two shapes, when viewed through the eyes of local salient part descriptors, are virtually indistinguishable. When, however, both shapes are viewed as whole, we see a difference, which is not conveyed through the salient parts.

Transformation Invariance. A different property of a shape representation is *shape equivalence* – under which conditions should two shapes be perceived as corresponding to the same object type [Palmer, 1999]. This question can be addressed by categorizing the transformations under which a shape should not change its type. A shape representation can be *transformation invariant* with respect to:

Rigid transformations: It is desirable that a shape descriptor is invariant to similarity transformations: translation, rotation and scaling; and their combinations. In addition, reflections do not change the shape identity.

Non-rigid transformation: A shape descriptor should be invariant to small non-rigid transformations. For example, minor stretching of object parts or even the whole object shape do not change the identity of the shape.

Articulations: A more severe class of deformations are articulations. In case of articulations, an object shape can be decomposed into several parts which can undergo large non-rigid transformation and usually the deformations of the different parts are independent. As such, articulations occur only for specific objects, such as humans, animals, etc.

Image artifacts. Object recognition in real images, which contain multiple objects and background structures, pose a different array of challenges, which affect shape-based recognition.

Clutter: Interior contours and background structure can add irrelevant information to any object representation. To address this problem, most of the object recognition methods rely on local features [Zhang et al., 2007a]. In the case of shape-based recognition, semi-local descriptors have been used [Ferrari et al., 2008], [Belongie et al., 2002]. This approach, however, can be only of limited success for shape, which best is described globally.

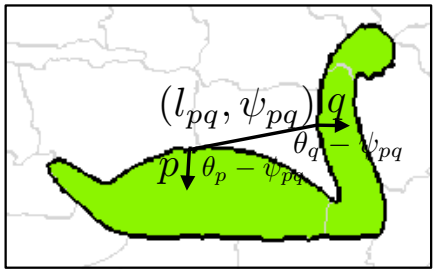
Occlusion and missing parts: Note that these types of challenges are more naturally dealt with using part-based shape representations. In the case of holistic representations, where an object part may affect the whole descriptor, dealing with occlusions becomes harder.

2.2 Chordiogram

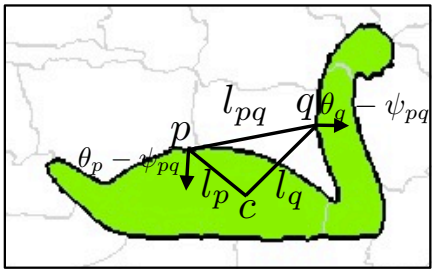
We introduce a novel shape descriptor, called a *chordiogram*, which addresses most of the challenges outlined above [Toshev et al., 2010]. It captures both the object boundary as well as its interior in a holistic fashion. In addition, it is invariant to certain rigid transformations and robust to shape deformations. Most importantly, however, it can be applied in images with severe clutter, which allows for recognition in unsegmented images.

2.2.1 Definition of the Chordiogram

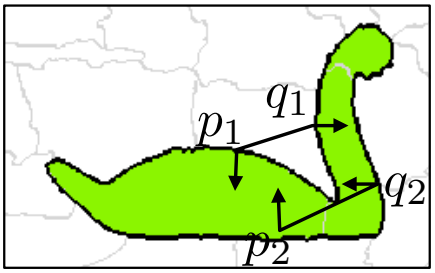
Let us denote by C all the boundary points of a segmented object. To define the chordiogram, consider for a moment a pair of boundary edges p and q from C. We will call such a pair (p, q) a chord. We can think of a chord as a way to express a



(a) Translation-invariant features.



(b) Rotation-invariant features.



(c) Normals.

Figure 2.4: Chord features and orientation of the normals at boundary edges.

dependency between edges p and q. One can define various features which describe the geometry of the chord, which we will denote by $f_{pq} \in \mathbb{R}^D$, as we will see in the subsequent section. These features capture geometrical relationships between the two boundary points.

We describe the shape of a segmented object by capturing the features of all chords. In this way we attempt to capture all dependencies among boundary points and achieve a holistic description. More precisely, the chordiogram ch is defined as a K-dimensional histogram of all chords, where the m^{th} chordiogram element is given by:

$$\text{ch}_m = \#\{(p,q) | f_{p,q} \in bin(m), p, q \in C\} \quad m = 1 \dots K \tag{2.1}$$

To define the chordiogram, one needs to sample points from the shape. In our

definition we take every pixel on the shape.

2.2.2 Chord Features

The exact features that are used to describe a chord determine the properties of the chordiogram. In our work, we use the following two possible sets of chord features.

I. Translation-invariant chord features. One potential chord characterization can be achieved if we focus on the relative geometric configuration of the two boundary edges. More precisely, we define four chord features (see Fig. 2.4(a)):

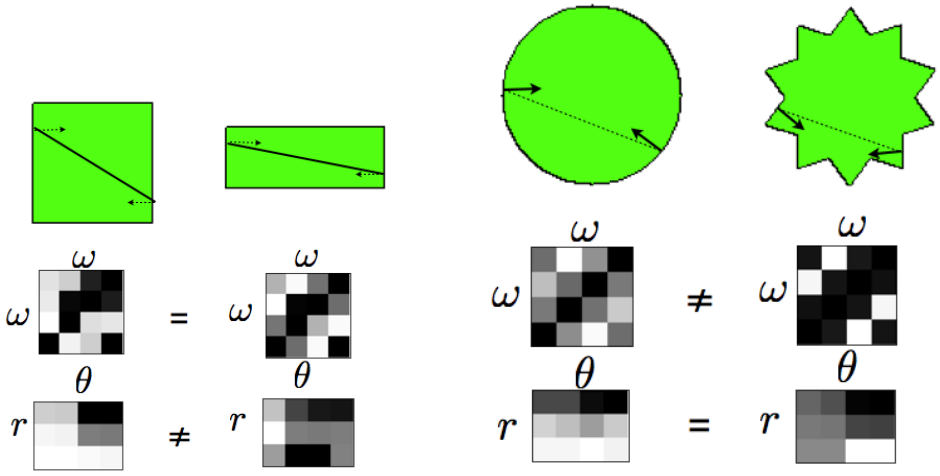
- Chord *length* l_{pq} and *orientation* ψ_{pq} of the vector connecting p and q.
- *Normals* θ_p and θ_q to the object boundary at p and q.

Thus, the *chord features* can be written as:

$$f_{pq}^{(t)} = (l_{pq}, \psi_{pq}, \theta_p - \psi_{pq}, \theta_q - \psi_{pq})^T$$

The normals are defined such that they point towards the interior of the object. In this way not only the contour shape at points p and q is captured but also the relation of the interior to the chord. For example, in Fig. 2.4(c) the chords at the two L-junctions at the bottom of swan's neck differ because the object interior is positioned differently w. r. t. the two junctions.

Since the features are real-valued, to compute the above histogram one needs to quantize the features into bins. The lengths l_{pq} are binned in b_l bins in a log space, which allows for larger shape deformation between points lying further apart. The length h of the largest bin determines the scale of the descriptor – every two boundary points lying within distance h will be captured by the descriptor. To guarantee that the descriptor is global, we set h equal to the diameter of the object in case of pre-segmented object masks. The remaining three features are angles lying in $[0, 2\pi)$ and are binned uniformly – the chord orientation in b_r bins; the normal angles are



(a) Coarse shape. (b) Fine shape.

Figure 2.5: For each pair of shapes (upper row), we show the chordiogram computed over the normal features only (middle row) and over the chord length and orientation (lower row).

binned in b_n angles. This binning strategy results in a $N = b_l \times b_r \times b_n^2$ dimensional shape descriptor at scale h. The chord features are summarized in Table 2.1.

Chord Features Analysis The chord features determine the invariance of the chordiogram to geometric transformations. Since we do not capture absolute rotation information, the resulting descriptor is translation invariant. However, the chord orientation prevents the descriptor from being rotation invariant. Similarly, the chord length prevents the chordiogram from being scale invariant. This design choice is motivated by the fact, that translation is the largest possible dimension of a similarity transformation we have to search along during detection. Moreover, the above version of the chordiogram is tailored towards image datasets, which exhibit the characteristics of personal photo collections – users tend to take pictures of objects in their natural pose, which usually means that we do not need to search over possible rotations.

Note, that we potentially could eliminate features which are not invariant to

	feature	binning	# bins	invariance		
				rotation	scale	translation
l_{pq}	chord length	log space	b_l	yes	no	yes
l_p	distance to center	uniform	b_d	yes	no	no
ψ_{pq}	chord orientation	uniform	b_r	no	yes	yes
$\theta_p - \psi_{pq}$	relative normal	uniform	b_n	yes	yes	yes

Table 2.1: Summary of the chord features and their properties. Note that both the chord length and distance to object center depend also on the scale, defined as the boundary of the largest bin.

other types of transformations. However, this would decrease the expressiveness of our representation.

The chord features are chosen such that they completely describe the geometry of a chord. When it comes to the chordiogram, the features capture different shape properties. The chord length and orientation capture global coarse shape properties, while the fine information is captured by the normals.

To see this, consider the example given in Fig. 2.5. We can restrict the computation of the chordiogram only over a subset of the features. If we only use the normals at the boundary points, then the fine boundary information shown in Fig. 2.5(b) can be distinguished. If we, however, use only the chord length and orientation, then we can discriminate based on coarse shape, as visualized in Fig. 2.5(a).

II. Rotation-invariant chord features. In certain applications, such as videos or multiple images of the same scene, we could use motion information or stereo to detect the rough location and support of a foreground object (see Chapter 4). In such situations, translation and scale invariance is irrelevant, while we need to deal with rotation.

To introduce a rotation-invariant variant of the chordiogram, consider the center of mass of the object outline defined as

$$c = \frac{1}{|C|} \sum_{p \in C} x_p \quad \text{where } C \text{ are all boundary points.}$$

For each boundary point $p \in C$ denote by $l_p = |pc|$ the distance to the object center c. Then the rotation-invariant features are (see Fig. 2.4(b)):

- Chord *length* l_{pq} of the vector connecting p and q and *distances to center* l_p and l_q.

- *Normals* θ_p and θ_q to the object boundary at p and q. To achieve rotation invariance of these features, the angles are normalized with respect to the chord orientation.

Thus, the *chord features* can be written as:

$$f_{pq}^{(r)} = (l_{pq}, l_p, l_q, \theta_p - \psi_{pq}, \theta_q - \psi_{pq})^T$$

The distances l_p and l_q are binned uniformly into b_d bins, while the remaining features are binned as above. This gives us a $N = b_l \times b_d^2 \times b_n^2$ dimensional descriptor. The chord features are summarized in Table 2.1.

2.3 Properties and Analysis of the Chordiogram

In this section we study the chordiogram with regard with the properties outline in Sec. 2.1.

Boundary and interior. An important difference to most contour-based shape representations, is that the chordiogram captures the contour orientation relative to the object interior. Orienting the boundary normals with respect to the interior allows us not only to capture different interpretations of a contour, as shown in Fig. 2.1 in Sec. 2.1, but also contributes to better discrimination, for example, between concave and convex structures (configurations f_{pq} and $f_{p'q'}$ respectively in Fig. 2.4(c)), which otherwise would be indistinguishable.

2.3.1 Gestaltism

The introduced descriptor is a *global* and *holistic* shape representation. To see this note that we take into account all possible chords – long chords as well as short chords. Thus we capture short as well as long geometric relations, which makes the descriptor global.

To give some intuition for the holistic nature of the descriptor, consider the example in Fig. 2.6. Denote by $\mathrm{ch}^{\mathrm{horse}}$ and $\mathrm{ch}^{\mathrm{centaur}}$ the chordiograms of the two shapes. Further, denote by $\mathrm{ch}_i^{\mathrm{horse}}$ and $\mathrm{ch}_i^{\mathrm{centaur}}$ the chordiograms of the i^{th} part, $i \in \{1, 2\}$. The distance $d(\cdot, \cdot)$ between two shapes, expressed with their chordiograms u and v, is defined in terms of the L_1 norm:

$$d(u, v) = \left|\left| \frac{u}{||u||_1} - \frac{v}{||v||_1} \right|\right|_1 \tag{2.2}$$

For the given examples and parameters of the descriptor set as $b_l = 4$, $b_r = 8$, $b_n = 8$, we obtain the following the following distance:

$$d(\mathrm{ch}^{\mathrm{horse}}, \mathrm{ch}^{\mathrm{centaur}}) = 0.72$$

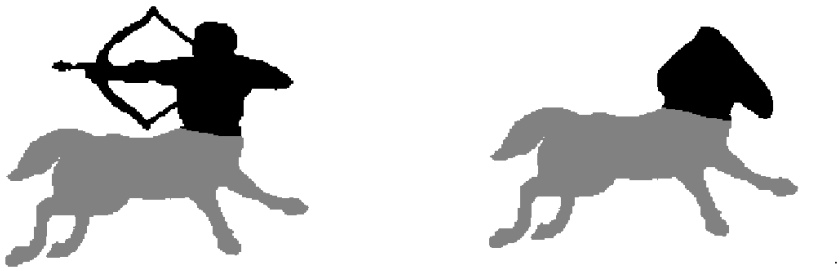
If we ignore the holistic portion of the descriptor, which relates both parts, and evaluate the distance between the parts only, we obtain:

$$d(\mathrm{ch}_1^{\mathrm{horse}} + \mathrm{ch}_2^{\mathrm{horse}}, \mathrm{ch}_1^{\mathrm{centaur}} + \mathrm{ch}_2^{\mathrm{centaur}}) = 0.46$$

This shows that the chordiogram captures not only the shape of the individual parts, but also their mutual relation. If we change one part, then the interpretation of the other parts change as well, although they remain the same. This results in a larger shape distance compared to an additive equivalent.

2.3.2 Deformation Analysis

In this section we analyze the performance of the chordiogram with respect to various deformations, such as *rigid transformations*, *local boundary deformations*, *boundary noise* and *occlusion*. The analysis is performed using the following setup.



(a) Centaur. (b) Horse.

Figure 2.6: Each of the two examples consists of two parts. Part 1 is the upper body (colored in black), while part 2 is the torso (in gray).

Figure 2.7: Example shapes used in the analysis of the chordiogram. We show 5 out of 10 examples for each of the ten classes: *octopus*, *pagodas*, *panda*, *pigeon*, *pizza*, *platypus*, *pyramid*, *rhino*, *rooster*, *saxophone*.

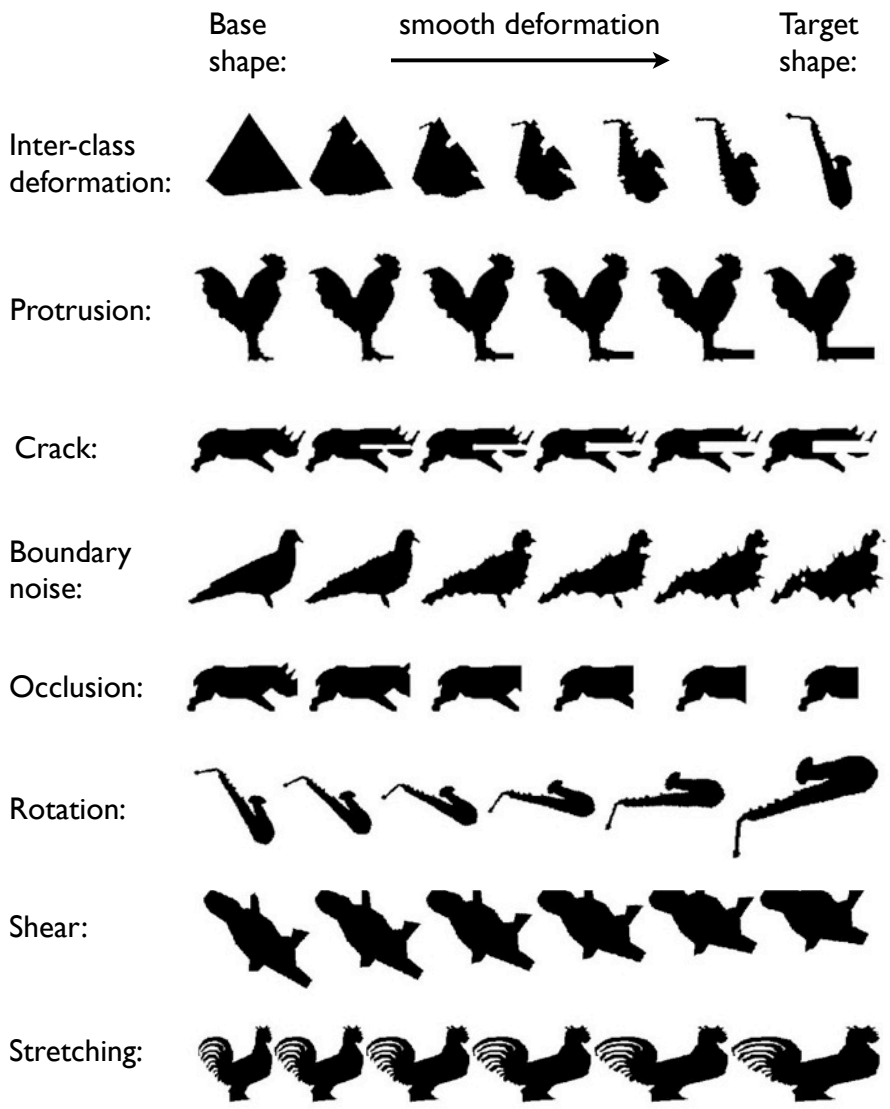


Figure 2.8: Examples of deformation paths of a reference shape to a target shape for each of the used deformation types.

Deformation Analysis Setup. We use a subset of the MPEG-7 CE-Shape 1 [Latecki et al., 2000] dataset, as presented in Fig. 2.7. We use 10 different classes and 10 examples per class. We will call those shapes *reference shapes*. For each of the analyzed deformations and each reference shape, we create a *target shape* which is a deformed version of the reference shape. For example, if the reference shape is a rooster and we analyze protrusions, we deform the rooster by introducing a protrusion (see second row of Fig. 2.8). Moreover, for each pair of a reference and a target shape, we introduce a *deformation path* which is a smooth morphing of the reference shape into the target one (see Fig. 2.8 for examples of several deformation paths).

In this setup, the target shape is considered a version of the reference shape after undergoing a severe deformation. The deformation path contains milder deformations, and the further one walks along the path, the more severe the deformation of the reference shape becomes.

By computing a chordiogram-based distance between the reference shape and each shape on the deformation path, we can analyze the degradation of our descriptor with respect to the type of deformation used to create this path and the degree of deformation. In particular, we use L_1 distance:

$$d_{\gamma,\text{deformation type}}(\text{shape}) = ||\text{ch}(\text{shape}) - \text{ch}_{\gamma,\text{deformation type}}(\text{shape})||_1$$

where the second descriptor is the chordiogram of the reference shape after undergoing a deformation of a specified type and degree $\gamma \in [0, 1]$. In this setup, $\gamma = 0$ corresponds to no deformation, while $\gamma = 1$ is full deformation, as defined below.

For the experiments below, we use the translation invariant version of the chordiogram with parameters $b_l = 5$, $b_r = 4$, and $b_n = 8$.

Deformations. We create the above deformation paths with respect to the following deformations (examples of each of the deformations is given in Fig. 2.8):

Rigid transformations: We use the following three rigid transformations:

Rotation: We rotate the object counterclockwise where $\gamma = 1$ corresponds to a rotation of 90 degrees.

Shear: For a shear factor $\lambda = 0.5\gamma$, we create a deformed shape $\{p_1, \ldots, p_n\}$ from the reference shape $\{p_1^r, \ldots, p_n^r\}$ such that for each point p_k there is a point p_m^r on the reference shape with $p_m^r = \begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix} p_k$.

Stretching: We stretch the reference shape along the x coordinate. The maximal stretch, corresponding to $\gamma = 1$, is 2.5.

We do not analyze translation and scaling since by having an already segmented object we have already dealt with such deformations.

Boundary deformations: We also introduce several local as well global deformations of the object boundary:

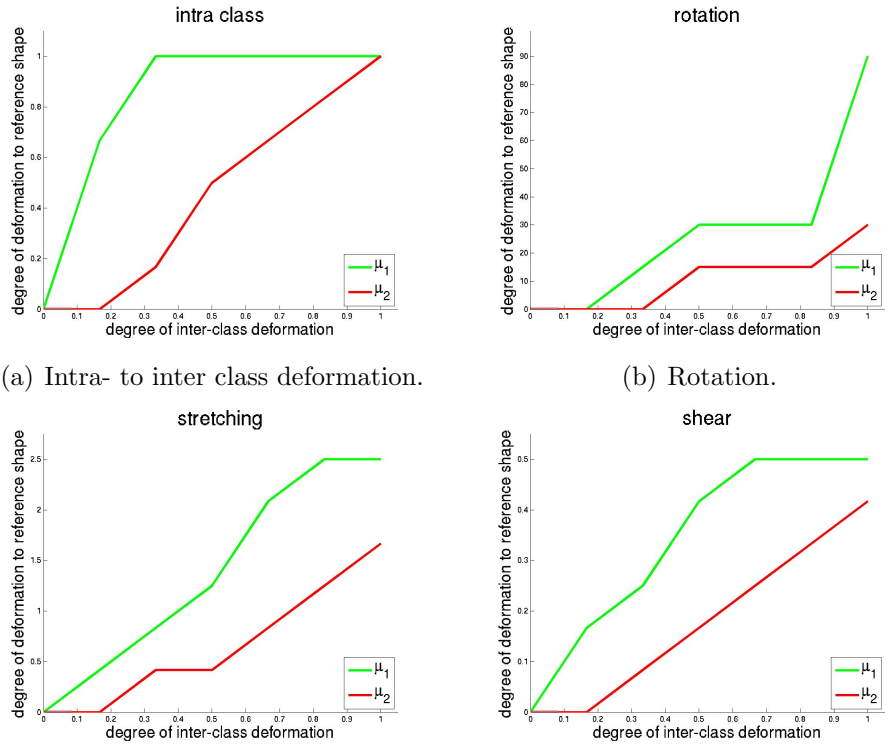
Protrusion: In order to analyze the effect of local boundary deformations, we deform the reference shape by using one horizontal rectangular protrusion. The length of the maximal protrusion is 0.5 of the length of the object.

Crack: Similarly to the protrusion, we introduce a crack with maximal length of 0.5 of the object length.

Boundary noise: In order to analyze the effect of global boundary noise, we perturb the object boundary. Each boundary point is moved along its normal at distance which is sampled from a normal distribution $\mathcal{N}(0, 0.1l\gamma)$, where l is the object length.

Occlusion: Occlusion is modeled by occluding the right side of the object with a rectangle. The maximal occlusion is covering half of the object.

The above deformations were chosen to be representative for rigid and non-rigid transformations to which a shape descriptor should be robust. The degree to which they are applied, as described above, was intended to introduce a severe change of



(a) Intra- to inter class deformation.

(b) Rotation.

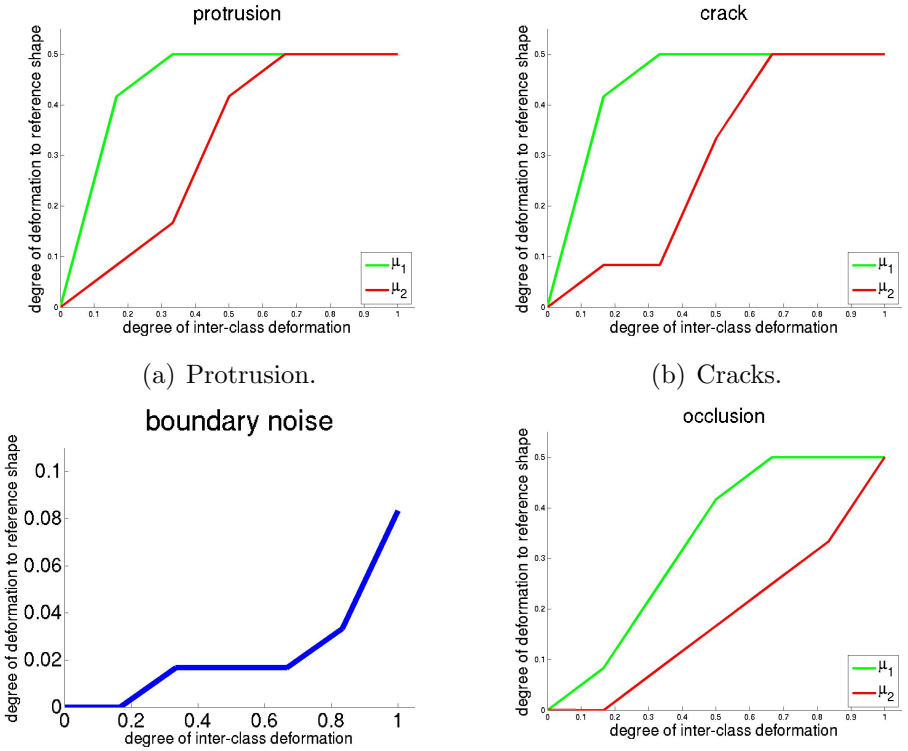
(c) Stretching.

(d) Shear.

Figure 2.9: Degradation of the chordiogram distance for rigid-transformations.

the object shape. However, this change should not distort the perception of the shape – a deformed object should be more similar in shape to itself than to any object of a different class. Therefore, to evaluate the degradation of the chordiogram distance while deforming a shape, we compare this degradation to the one which will be observed if we morph the object to an object of a different class. For this purpose, we define an inter-class deformation path. It is computed by randomly choosing for each reference shape a shape of a different class.

Analysis. The degradation of the chordiogram distance for each deformation is shown in Fig. 2.9 and Fig. 2.10. The X axis of each plot shows a point in the inter-class deformation path. Suppose that γ denotes a point on the X axis, i. e. a



(a) Protrusion.

(b) Cracks.

(c) Boundary noise.

(d) Occlusion.

Figure 2.10: Degradation of the chordiogram distance for boundary deformations and occlusion.

degree of morphing of an object of one class to another object of a different class. For each γ and deformation type, we plot in green the maximally tolerable degree of deformation $\mu_1(\gamma)$ – the degree of deformation which yields on average lower chordiogram distances than the distances of the γ-inter-class deformation:

$$\mu_1(\gamma) = \max\{\beta | \mathbb{E}_{\text{shape}}(d_{\beta,\text{deform. type}}(\text{shape})) \leq \mathbb{E}_{\text{shape}}(d_{\gamma,\text{intra-class}}(\text{shape}))\}$$

where the above expectation is computed over all reference shapes.

Since the mean does not tell us anything about the variance of the data, we use a second definition of a maximally tolerable degree of deformation – its average chordiogram distance is smaller than the average γ-inter-class deformation by at

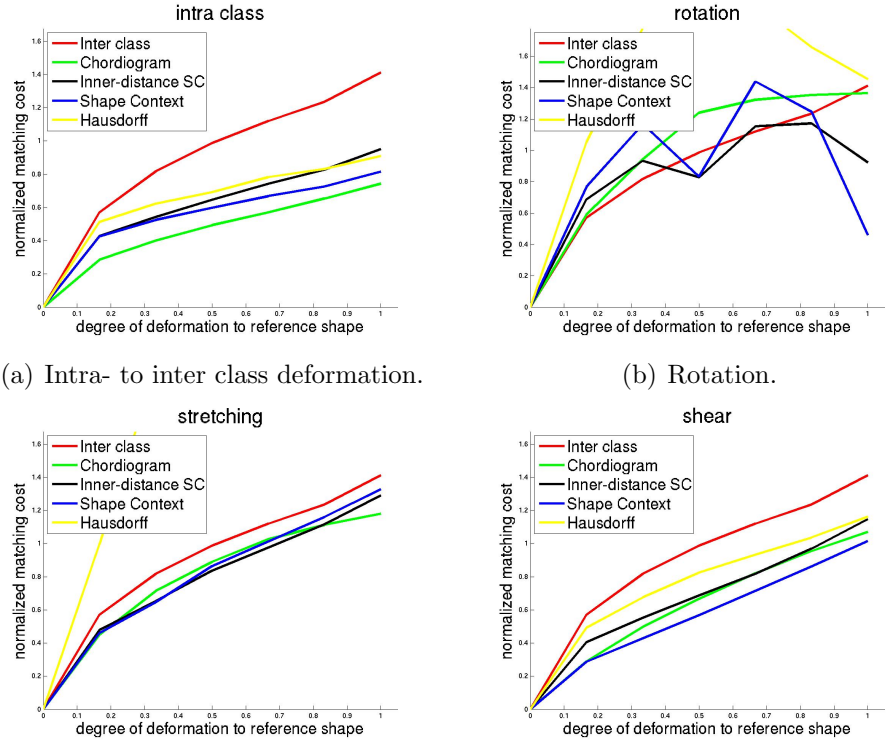
least the sum of the standard deviations σ of both deformations:

$$\mu_2(\gamma) = \max\{\beta | \mathbb{E}_{\text{shape}}(d_{\beta,\text{deform. type}}(\text{shape})) + \sigma_{\text{shape}}(d_{\beta,\text{deform. type}}(\text{shape})) \leq \mathbb{E}_{\text{shape}}(d_{\gamma,\text{intra-class}}(\text{shape})) - \sigma_{\text{shape}}(d_{\gamma,\text{intra-class}}(\text{shape}))\} \quad (2.3)$$

We plot the second curve μ_2 in red.

In Fig. 2.9(a) we compare the inter-class with the intra-class chordiogram distance. We can see that on average the chordiogram distance for objects of the same class is smaller than the distance of objects morphed with $\gamma = 0.3$ to objects of different classes. In addition, we plot in Fig. 2.9(b)-(d) the degradation of the chordiogram distance with respect to rigid transformations. Using the μ_2 measure of deformation tolerance, the maximally tolerable stretch is 1.7, shear 0.42, and rotation of 30 degrees. This means, that deformations stronger than the aforementioned parameter will make an object look as dissimilar to the original object as an object of different class. The above values are a direct function of the chordiogram parameters. Making the chordiogram bins larger would make the descriptor more tolerable to rigid transformations. However, its inter-class versus intra-class separability will decrease.

Further analysis is presented in Fig. 2.10. We can see that the chordiogram is robust to protrusion and cracks which extend up to half of the object length. This can be explained with the fact that both protrusions and cracks are local deformation. Although they add a large new part to the shape, most of the shape remains unchanged which is captured by the chordiogram. Boundary noise, however, is less tolerated by the chordiogram. We can see that perturbing the points by $0.03l$, where l is the object length, can make an object look almost as an object of a different class ($\gamma = 0.85$ of the average inter-class deformation path). This can be explained by the fact that noisy boundaries affect strongly boundary normals which an integral part of the descriptor. Finally, occluding an object will approximately linearly degrade the chordiogram distance. Occlusion of 50% of the object makes it indistinguishable



(a) Intra- to inter class deformation.

(b) Rotation.

(c) Stretching.

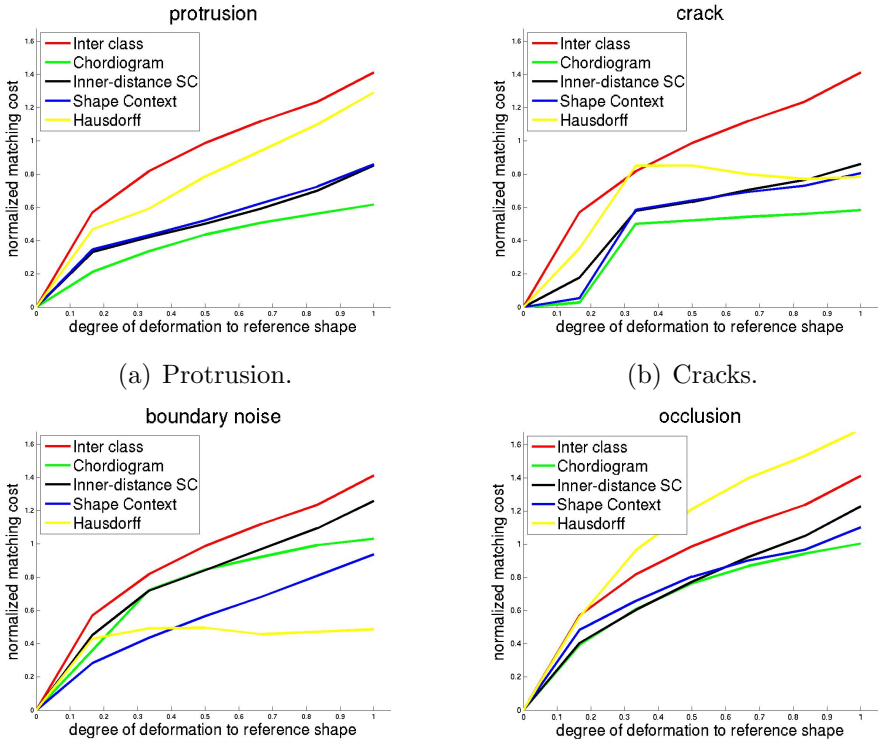
(d) Shear.

Figure 2.11: Degradation of the chordiogram distance for rigid-transformations. We plot average deformation paths for four different shape matching techniques.

from object of different classes.

2.3.3 Comparative Deformation Analysis

We compare the performance of the chordiogram under the deformations used in Sec. 2.3.2 with three other shape representations and matching techniques – Hausdorff distance (HD) [Huttenlocher et al., 1993], Shape Context (SC) [Belongie et al., 2002] and Inner Distance Shape Context (IDSC) [Ling and Jacobs, 2007]. For each of the aforementioned approaches, we compute a matching cost along a deformation path for each shape and deformation type. We



(a) Protrusion.

(b) Cracks.

(c) Boundary noise.

(d) Occlusion.

Figure 2.12: Degradation of the chordiogram distance for boundary deformations and occlusion. We plot average deformation paths for four different shape matching techniques.

denote this cost by

$$d_{\gamma,\text{deform.},\text{algo}}(shape)$$

where algo is one of the above shape matching techniques. In the case of the chordiogram, we use the L_1 distance as explained in Eq. 2.2.

Since the costs of the different techniques are on different scales, we normalize them in such way that the inter-class deformation cost is the same as the one of the chordiogram:

$$d'_{\gamma,\text{deform.},\text{algo}}(shape) = \frac{d_{\gamma,\text{deform.},\text{algo}}(shape)\, d_{\gamma,\text{inter-class},\text{chordiogram}}(shape)}{d_{\gamma,\text{inter-class},\text{algo}}(shape)}$$

We plot d' in Fig. 2.11 and Fig. 2.11 for the deformations introduced in Sec. 2.3.2.

Figure 2.13: Overview of the MPEG-7 dataset. We show a single example from each class.

We can see that the chordiogram performs similarly to SC and IDSC with respect to rigid transformations, while HD performs much worse. This is a result of the fact that all three descriptors are histogram-based and thus they exhibit similar performance to rigid-transformations. When it comes to cracks and perturbations, the chordiogram performs slightly better than SC and IDSC.

2.3.4 Recognition Experiments

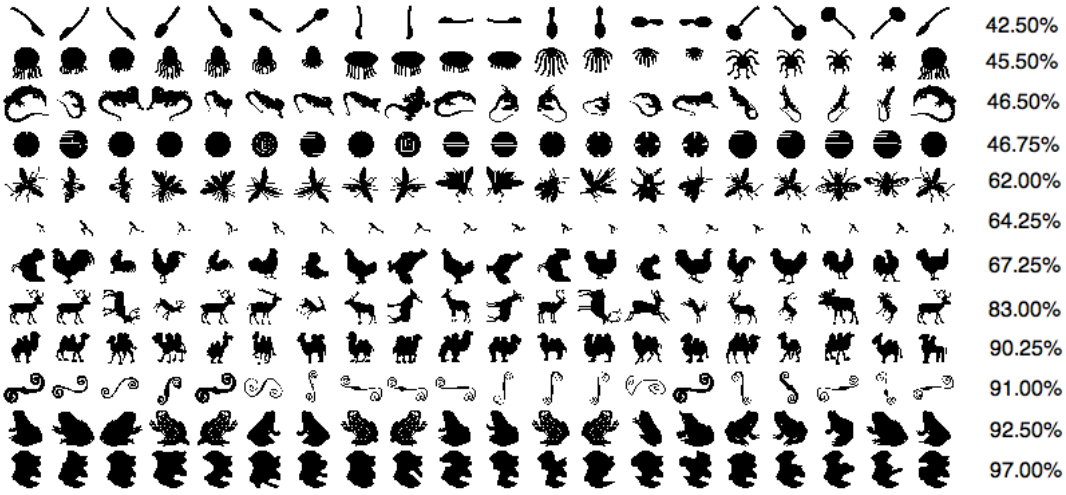


Figure 2.14: Example performance of the translation invariant chordiogram on MPEG 7. We show several classes and the achieved performance on the right.

To evaluate the performance of the chordiogram for the task of object recognition, we perform experiments on the MPEG-7 CE-Shape 1 part B dataset [Latecki et al., 2000].

(a) Translation invariant chordiogram.

level	parameters			dimension	Bulleye
	b_l	b_r	b_n		
	4	8	8	2048	80.85%

(b) Rotation invariant chordiogram.

level	parameters			dimension	Bulleye
	b_l	b_d	b_n		
1	2	1	2	32	55.19%
2	4	3	4	576	74.78%
3	8	7	8	25088	78.85%
4	16	15	16	921600	77.99%
5	32	31	32	31490048	75.99%
PMK					79.52%

Table 2.2: We present parameters of the chordiogram and the corresponding Bulleye score obtained on the MPEG7 dataset. Top (a): translation-invariant variant of the chordiogram. These parameters will be used in the experiments in the subsequent chapters. Bottom (b): rotation-invariant chordiogram with several parameter settings, called also levels. In addition, we aggregate the descriptors over the different levels using Pyramid Matching Kernels and show the corresponding score.

This is a an established dataset which is used to evaluate shape-based classification and retrieval. It consists of 1400 binary object masks representing 70 different classes, each class having 20 examples (see Fig. 2.13). The recognition rate reported for this dataset is the Bulleye score: each shape is matched with all shapes and the percentage of the 20 possible correct matches among the top 40 matches is recorded; the score is the average percentage over all shapes.

In our experiments we use L_1 distance between chordiograms extracted from each shape. We test both the translation- and rotation-invariant versions of the chordiogram (see Sec. 2.2.2).

The achieved Bulleye scores are summarized in Table 2.2. To compute a distance between each two objects, we first scale-normalize all the objects. Since the translation invariant chordiogram is not rotation invariant, we rotate each image b_r times using b_r rotations of angle $\{0, \frac{2\pi}{b_r}, \ldots, (b_r - 1)\frac{2\pi}{b_r}\}$ around the object center of mass

Method	Bulleye score
CSS [Mokhtarian et al., 1997]	75.44%
Visual parts [Latecki and Lakamper, 2000]	76.45%
Shape Context + TPS[Belongie et al., 2002]	76.51%
Aligning curves [Sebastian et al., 2003]	78.16%
Rotation inv. chordiogram	**79.52%**
Generative Models [Tu and Yuille, 2004]	80.03%
Translation inv. chordiogram	**80.85%**
Inner-distance SC + DP [Ling and Jacobs, 2007]	85.40%
Hierarchical Procrustes [Mcneill and Vijayakumar, 2006]	86.35%
Shape Tree [Felzenszwalb and Schwartz, 2007]	87.70%

Table 2.3: Comparison of the performance of the chordiogram with other shape matching methods on the MPEG dataset.

and compute the chordiogram. Thus, we obtain b_r descriptors $\{\mathrm{ch}_i^{(1)}, \ldots, \mathrm{ch}_i^{(b_r)}\}$ for the i^{th} object corresponding to two different rotations. The distance between tho images i and j is defined as the smallest distance in L_1 sense among all rotated chordiograms:

$$d(i,j) = \min_{\theta_i,\theta_j}\{||\mathrm{ch}_i^{(\theta_1)} - \mathrm{ch}_j^{(\theta_j)}||_1 | \theta_i, \theta_j \in \{1, \ldots, b_r\}\}$$

Using the above setup, we achieve a score of 80.85%.

In addition, we evaluate the performance of the rotation invariant chordiogram, which is more suited for this experiment. The results show that the bin sizes are not so crucial and different parameters lead to similar scores (see Table 2.2(b)). Moreover, if we combine them using Pyramid Matching Kernel approach [Grauman and Darrell, 2007], we achieve score of 79.52%.

A comparison to other shape matching approaches is presented in Table 2.3. We outperform most of the approaches with exception of Shape Trees [Felzenszwalb and Schwartz, 2007], Hierarchical Procrustes [Ling and Jacobs, 2007] and Inner Distance Shape Context [Ling and Jacobs, 2007]. The main reason is that the latter approaches are based on metrics which are computed along the shape contour, while our approach uses Euclidean distances to capture shape. As a result

it does not deal as well with non-rigid deformations as the above methods. However, as we will see in the subsequent chapters, the particular definition of the chordiogram allows it to be combined with segmentation and in this way to be applied to cluttered images, while the above methods assume that the object has been pre-segmented. An additional advantage of the chordiogram is that its distance is simply a L_1 norm computation, while the above approaches require an inference of some sort.

To better understand the performance of our representation, we show several classes from MPEG in Fig. 2.14 and their Bulleye score. We can see that the chordiogram performs well on classes which have medium/large deformation (misk in last row, frog) and medium articulations (camel, deer). However, the approach has problems with classes which are highly deformed (lizard), highly articulated (octopus), or are characterized by small details (head of fork, device9 in fourth row).

2.4 Related Work

2.4.1 Shape Representations

The chordiogram has been inspired by and exhibits similarities to several other representation techniques in computer vision.

Shape representations. The most closely related shape descriptor is the Shape Context (SC) [Belongie et al., 2002]. In the context of the chordiogram, the SC can be thought of capturing the relationship of all boundary points to only one boundary point, which is the descriptor offset. These relationships are parameterized in a similar fashion and captured in a histogram. The fact that SC is defined with respect to a fixed offset, makes it more descriptive. However, contrary to the chordiogram, the SC is not global and not translation-invariant. More in historic perspective, pairwise geometric histograms have been used to capture relations of all lines in an image to a reference line [Evans et al., 1992].

The descriptor is also inspired by [Carlsson, 1999], which captures properties of set of points. These sets are larger than pairs and the properties are related to the order structure among points and tangent lines at those points. The major difference is that the resulting properties are of a topological nature, while our approach has a metric nature.

Shape representations in 3D. Histograms of geometric properties of sets of points have been used to match 3D models [Osada et al., 2002]. These histograms can be interpreted as distributions of shape functions, where each function represents a property of a small set of points (pairs, triplets or four points). One of the motivations of this work is the fact that 3D objects lack natural parameterization as the arc length used for 2D contours. Note, however, that in case of unsegmented objects in cluttered images we are faced with ambiguity – although we know how to parameterize a single contour, we do not know which contours comprise the object and thus we cannot parameterize directly the object boundary. As we will see in the next chapter, the histogram nature of our representation allows us to select the object boundary and resolve the above ambiguity.

2.4.2 Shape Part Correspondence

A common paradigm in shape matching is to try to quantify the similarity between two shapes by establishing correspondences between points on the shapes. The correspondence between the points serves as an explanation of the match, while the quality of the match is determined using a matching model [Yoshida and Sakoe, 1982, Basri et al., 1998]. The chordiogram, as defined in Sec. 2.2.1, does not capture any absolute boundary point information as part of the chord features, neither it captures any location relations among chords. As a result, it is not clear whether the chordiogram, as a histogram, can be used to establish correspondences among boundary points of two shapes.

In this section we relate the chordiogram to the graph matching problem, which is a widely used approach to the correspondence problem [Shapiro and Haralick, 1979, Gold and Rangarajan, 1996, Umeyama, 1988a], and obtain the following insights:

1. We provide a different interpretation of the chordiogram matching as bipartite matching among chords. We show that the chordiogram can be used to compute the cost of this bipartite matching efficiently without recovering any explicit correspondences.

2. We bound the chordiogram matching from above with the cost of a graph matching among points on the shape. This relates our descriptor to correspondence estimation.

3. Finally, we show how to estimate correspondences between shapes starting from the bipartite matching interpretation of our descriptor.

Next we set up the notation and tools needed for the subsequent analysis.

Graph matching. Suppose that the two shapes, whose similarity needs to be assessed, are defined in terms of point sets:

$$P^s = \{p_1^s, \cdots, p_n^s\} \quad \text{for} \quad s \in \{1, 2\}$$

For simplicity, we assume that both point sets have the same cardinality n. In this case, we can think of a shape as a complete graph, whose nodes are the above point set and the edges are the chords (see Sec. 2.2.1).

Chord distances. Furthermore, a chord (i, j) from shape s, and in this way the edge (i, j) from graph s, can be described by the bin into which it falls using a predefined binning scheme b. This can be written as a trivial chordiogram $\mathrm{ch}_{ij}^{b,s}$, in which only the chord (i, j) with a feature vector f_{ij}^s gets binned (see Eq. 2.1):

$$(\mathrm{ch}_{ij}^{b,s})_m = \begin{cases} 1 & \text{if } f_{ij}^s \in \mathrm{bin}_b(m) \\ 0 & \text{otherwise} \end{cases}$$

Denote further by $\mathrm{ch}^{b,s}$ the chordiogram for shape s using binning scheme b and $N = \binom{n}{2} = ||\mathrm{ch}^1|| = ||\mathrm{ch}^2||$ the number of chords.

In the following exposition we will use a sequence of nested binning schemes, as defined in [Indyk and Thaper, 2003]. Suppose that Δ is the diameter of the chord set of both shapes, where the diameter is defined in terms of the L_1 distance on the feature vector f_{ij} of a chord (i, j). Then the b^{th} binning scheme is defined by partitioning each feature space using a grid of size 2^b. The values of b are such that they define together a fine to coarse hierarchical binning: the grid cell length at level b is 2^b, $b \in \{1/2, 1, 2, 4, \ldots, 2^B\}$ with $B = \lceil \log_2 \Delta \rceil$.

Using the above descriptors of a chord, we can define the following three distances $W_{ij;kl}$ between chords (i, j) and (k, l) from two different shapes, which characterize their dissimilarity:

L_1 in original feature space:

$$W^{\text{orig}}_{ij;kl} = ||f^1_{ij} - f^2_{kl}||_1 \tag{2.4}$$

Bin-based distance: For a particular binning scheme b , one can declare two chords similar if they lie in the same bin, and dissimilar otherwise. This can be expressed as follows:

$$W^{\text{bin}}_{ij;kl} = ||\mathrm{ch}^{b,1}_{ij} - \mathrm{ch}^{b,2}_{kl}||_1 \tag{2.5}$$

Multilevel bin-based distance: In addition to the above bin comparison distance, one can combine multiple binning schemes into a single distance:

$$W^{\text{mbins}}_{ij;kl} = \sum_{b=-1}^{B} \alpha_b ||\mathrm{ch}^{b,1}_{ij} - \mathrm{ch}^{b,2}_{kl}||_1 \quad \text{with weights} \quad \alpha_b \tag{2.6}$$

Graph matching formulation. We would like to recover one-to-one correspondence between both graphs. For this purpose, we define a correspondence indicator variable

$$x_{ik} = \begin{cases} 1 & \text{if } p^1_i \text{ and } p^2_k \text{ are in correspondence;} \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

Then, a matching problem, which evaluates the structural similarity between the graphs, can be formulated as follows:

$$\text{(GM)}: \qquad \min_x x^T W x = \sum_{ijkl} W_{ij;kl} x_{ik} x_{jl} \tag{2.8}$$

$$\text{subject to} \quad \sum_k x_{ik} = 1 \text{ for all } i; \quad \sum_i x_{ik} = 1 \text{ for all } k \tag{2.9}$$

$$x_{ik} \in \{0,1\} \text{ for all } i,k \tag{2.10}$$

where w can be any positive chord distance, such as the ones defined in Eq. (2.4–2.6). The constraints (2.9) guarantee one-to-one correspondence, while the integral constraints (2.10) assure that the solution to the problem is a correspondence indicator variable, as defined in Eq. (2.7).

Relaxation of the graph matching formulation. Following [Chekuri et al., 2005], we reformulate the above problem into an equivalent one, in which we introduce a new set of variables $X : X_{ijkl} = x_{ik} x_{jl}$. These variables can be thought of as correspondence variables between chords. Then problem (GM) from Eq. (2.8) reads:

$$\text{(GMC)}: \quad \min_X W \cdot X = \sum_{ijkl} W_{ij;kl} X_{ijkl} \tag{2.11}$$

$$\text{subject to} \quad \sum_{k,l} X_{ijkl} = 1 \text{ for all } i,j; \quad \sum_{i,j} X_{ijkl} = 1 \text{ for all } k,l \tag{2.12}$$

$$\sum_l X_{ij_1kl} = \sum_l X_{ij_2kl} \text{ for all } i,k,j_1,j_2 \tag{2.13}$$

$$\sum_j X_{ijkl_1} = \sum_l X_{ijkl_2} \text{ for all } i,k,l_1,l_2$$

$$X_{ijkl} \in \{0,1\} \text{ for all } i,k \tag{2.14}$$

Constraints (2.12) stem directly from the definition of X and the constraints (2.9) on x. Further, the constraints (2.13) assure that corresponding chords agree on a unique correspondence between the points. This constraint can be derived from the

following relationship between point and chord correspondences:

$$x_{ik} = \sum_{l} X_{ijkl} \quad \text{for all } j \tag{2.15}$$

To solve the integer program (GMC), one can relax the integral constraints (2.14) to non-negativity constraints. As a result, one obtains the following exactly solvable linear program [Chekuri et al., 2005], which we will call point matching (PM) indicating that it aims to recover point correspondences:

$$\text{(PM)}: \quad \min_{X} W \cdot X \quad \text{subject to} \quad X \in \mathcal{P}_{\text{PM}} \tag{2.16}$$

where $\mathcal{P}_{\text{PM}}$ denotes the following polytope:

$$\mathcal{P}_{\text{PM}} = \{ \quad \sum_{k,l} X_{ijkl} = 1 \text{ for all } i,j; \quad \sum_{i,j} X_{ijkl} = 1 \text{ for all } k,l; \tag{2.17}$$

$$\sum_{l} X_{ij_1kl} = \sum_{l} X_{ij_2kl} \text{ for all } i,k,j_1,j_2; \tag{2.18}$$

$$\sum_{j} X_{ijkl_1} = \sum_{l} X_{ijkl_2} \text{ for all } i,k,l_1,l_2;$$

$$X \geq 0\} \tag{2.19}$$

A different relaxation would be to retain the integral constraints (2.14), but to remove the constraints (2.13). This corresponds to bipartite matching among the chords of the two shapes:

$$\text{(CM)}: \quad \min_{X} W \cdot X \quad \text{subject to} \quad X \in \mathcal{P}_{\text{CM}} \tag{2.20}$$

with

$$\mathcal{P}_{\text{CM}} = \{ \quad \sum_{k,l} X_{ijkl} = 1 \text{ for all } i,j; \quad \sum_{i,j} X_{ijkl} = 1 \text{ for all } k,l; \tag{2.21}$$

$$X_{ijkl} \in \{0,1\} \text{ for all } i,j,k,l\} \tag{2.22}$$

The latter program does not guarantee that the resulting chord correspondence can be directly translated to point correspondences. However, it is an integer program, which can be solved exactly using Max-Flow algorithms.

Relations between graph matching and chordiogram distance. Using the above definition of graph matching and its relaxations, one can show that the chordiogram matching is closely related to the correspondence problem between two shapes. First, we show the relationship between the chordiogram and bipartite matching among chords:

Theorem 1. *Define the set*

$$\mathcal{P}^*_{CM} = \{X \in \mathcal{P}_{CM} | \sum_{\substack{(i,j)\in bin_b(m) \\ (k,l)\in bin_b(m)}} X_{ijkl} = \min\{ch^1_m, ch^2_m\} \text{ for all bins } m \text{ and schemes } b\} \tag{2.23}$$

as a subset of $\mathcal{P}_{CM}$ for which the chord correspondence variable X is constrained through the chordiograms.

Then we can show that each $X^* \in \mathcal{P}^*_{CM}$ is a minimizer of the problem (CM) with data terms W^{mbins} and the minimum of this problem is analytically computable using the chordiogram:

$$W^{mbins} \cdot X^* = \sum_{b=-1}^{B} \alpha_b ||ch^{b,1} - ch^{b,2}||_1$$

for weights $\alpha_b = 2^b$.

Furthermore, we can relate the chordiogram matching to point matching between shapes:

Theorem 2. *Suppose that $X^*_{cm,orig}$ is the minimizers of problem (CM) in Eq. (2.20) using the data terms W^{orig}. Further, X^*_{pm} is the minimizer of problem (PM) in Eq. (2.16) using the data terms W^{mbins}. Then, the following relationship holds:*

$$C(W^{orig} \cdot X^*_{cm,orig}) \leq \sum_{b=-1}^{B} \alpha_b ||ch^{b,1} - ch^{b,2}||_1 \leq W^{mbins} \cdot X^*_{pm}$$

for a positive constant C.

The proof of both theorems is given in Appendix A. There are several insights we gain from the above theorems which relate our shape representation to matching points on the two shapes.

1. As shown in Theorem 1, the chordiogram distance is a minimizer of a bipartite matching among chords for a specific form of the chord distances. Thus, it quantifies the best possible correspondences among chords on two shapes without explicitly giving those correspondences. In addition, the chordiogram distance does not require any inference and is thus more efficient.

2. As shown in the first inequality of Theorem 2, the chordiogram over several binning schemes is an upper bound of the bipartite matching for which the similarities are defined in the original chord feature space. This shows that by choosing several binning schemes for the chordiogram, we can obtain an approximation to the original distance in the chord feature space.

3. As shown in the second inequality of Theorem 2, the distance based on our shape descriptor is a lower bound of the linear programming approximation for establishing correspondences among points on two shapes.

Correspondence recovery. The above theorem is based on the fact that we can think of chordiogram matching as a different relaxation of the original graph matching formulation. This allows for recovery of point correspondences – if we have $X \in \mathcal{P}_{PM}$, then we can use Eq. (2.15) for an arbitrary j to estimate point correspondences. To obtain such an X, however, we will not solve (PM) directly, but rather use the solution for (CM) obtained from the chordiogram matching. More precisely, we will try to find $X \in \mathcal{P}_{PM}$ closest to any minimizer of (CM):

$$\min_{X}\{||X - X^*_{\text{cm}}||_2 | X \in \mathcal{P}_{PM}, X^*_{\text{cm}} \in \mathcal{P}^*_{\text{CM}}\} \tag{2.24}$$

Note the above problem is an integer quadratic program, and thus NP-hard. To obtain an approximate solution, one can relax the above problem by replacing the

Figure 2.15: Examples of recovered correspondence on pairs of shapes. Points, colored in the same color, are in correspondence.

integral constraints with nonnegativity constraints in the definition of $\mathcal{P}^{*}_{\mathrm{CM}}$:

$$\begin{aligned}\mathcal{P}^{**}_{\mathrm{CM}} = \{ \; & \sum_{k,l} X_{ijkl} = 1 \text{ for all } i,j; \quad \sum_{i,j} X_{ijkl} = 1 \text{ for all } k,l; \\ & X_{ijkl} \geq 0 \text{ for all } i,j,k,l; \\ & \sum_{\substack{(i,j)\in \mathrm{bin}(m) \\ (k,l)\in \mathrm{bin}(m)}} X_{ijkl} = \min\{\mathrm{ch}^1_m, \mathrm{ch}^2_m\} \text{ for all bins } m\}\end{aligned}$$

The above polytope $\mathcal{P}^{**}_{\mathrm{CM}}$ is a convex set and if we replace $\mathcal{P}^{*}_{\mathrm{CM}}$ for $\mathcal{P}^{**}_{\mathrm{CM}}$ in problem (2.24), then we obtain a convex program. The correspondence recovery procedure is summarized in Algorithm 1.

Algorithm 1 Correspondence estimation from chordiograms.

Require: Chordiograms ch^1, ch^2 of two shapes.
1: Define $\mathcal{P}^{**}_{\text{CM}}$ using ch^1 and ch^2.
2: Solve program (2.24) and obtain minimizer $X^* \in \mathcal{P}_{PM}$.
3: Recover correspondence indicator variables x from X^* using Eq. (2.15).
4: Obtain discrete indicators $\hat{x}_{ij} = 1$ iff $j = \arg\max_{j_1}\{x_{ij_1}\}$, and 0 otherwise.

Examples. We show results of the correspondence recovery algorithm on selected pairs of shapes from MPEG 7 dataset [Latecki et al., 2000]. From each shape, defined by the outline of the shape mask, we sample uniformly 30 points, which are to be put in correspondence. The chordiogram is computed using only the sample points. For the optimization problem in step 2 of the algorithm, we use the `CVX` optimization package [Grant and Boyd, 2010]. Results are shown in Fig. 2.15. As we can see, correct correspondences are recovered for most of the points for articulated as well as rigid objects. The main problems arise in cases of strong articulation (see tree in row 3, column 1, where the orientation of the branches differs drastically), or lack of matching points (see elephant in row 1, column 3, where in the left object two legs are visible, while in the right object three legs are visible).

2.5 Conclusion

In this chapter we have introduced a novel shape descriptor, called chordiogram. We showed that this representation is consistent with basic shape principles – it should describe shape boundary and interior, has a global and holistic nature and is robust to deformations. We contrast the chordiogram with other shape representations. In particular, we show that although it is a global histogram, the presented descriptor can be thought of as an approximation of graph matching between points on two shape outlines. In this way, we are able to use it for recovery of correspondences between two matching shapes.

In the next chapters we will build upon the introduced shape representation.

In Chapter 3 we will integrate the chordiogram with segmentation and thus use it in cluttered scenes. In addition, region and boundary grouping principles can be combined with the chordiogram, as shown in the same chapter. Finally, application of the rotation invariant version of our descriptor to object detection in videos will be studied in Chapter 4.

Chapter 3

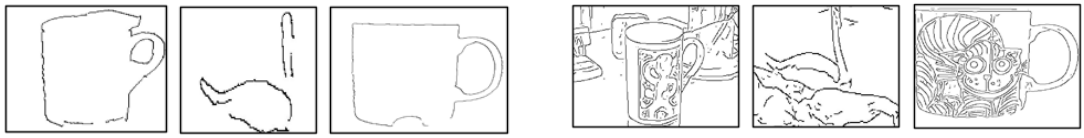
Shape-based Detection

In the previous chapter we introduced a novel shape descriptor, called the *chordiogram*, which is consistent with most of the properties which a shape representation should possess. In particular, this descriptor is holistic and has global support. Unfortunately, such global representations suffer from all the irrelevant structure present in images, such as interior contours and background clutter. This is a major challenge while applying an object representation in real settings, such as scene images of multiple objects and rich background structure.

In this chapter we present a framework for shape detection based on the chordiogram which addresses the problem of clutter [Toshev et al., 2010]. In the next section, we define the shape detection problem. In Sec. 3.2-3.5 we present the formulation of the detection model, whose inference is explained in Sec. 3.6.

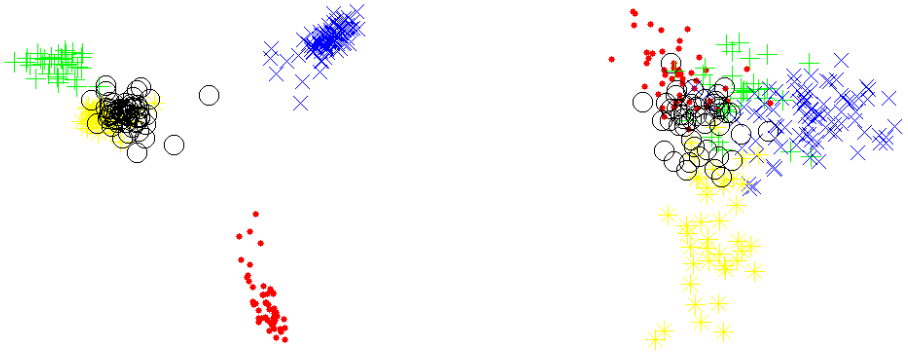
3.1 Problem Formulation

To see the importance of the problems arising from clutter, consider for a moment the ETHZ Shape dataset [Ferrari et al., 2009] which contains 255 images containing objects of 5 different classes. Suppose that an oracle has supplied us with masks and boundaries of those objects. If we compute the chordiogram using only the object



(a) Original images: Only object boundaries.

(b) Original images: All image contours



(c) LDA: Only object boundaries.

(d) LDA: All image contours

Figure 3.1: Top: Examples of object boundaries and all image contours. Bottom: The top 2 principal components of chordiograms computed using Linear Discriminant Analysis for objects in the ETHZ Shape dataset.

boundaries, than the top two principle components reveal a feature space, in which all classes are well separated from each other (see Fig. 3.1). However, if we use all image contours we obtain a feature space which is not anymore separable. Therefore, image clutter presents a major challenge to shape detection approaches, especially when we deal with images of scenes. This is one of the main reasons why well motivated shape representations such as shocks and shock graphs [Kimia et al., 1995, Sebastian et al., 2004] were not successfully applied to cluttered scene images.

In this chapter we show how we can use the chordiogram to perform shape detection in cluttered scenes. More precisely, we unify the problems of shape-based *detection* and object *segmentation* in a single problem. For an input image and a set of object masks, representing the shape of a set of objects, a shape detection method should produce:

Segmentation: A segment which represents an object. This segment should be:

1. *Similar in shape* to one of the object models. This is realized through a top-down process exploiting object-specific knowledge. Evidence from human perception indicates that familiarity with the target shape plays a large role in figure/ground assignment.

2. *Perceptually salient* region. This criterion is executed through a bottom-up process based on general grouping principles, which apply to wide range of object masks. In particular, the perceptual grouping component, which we define, is based on configural cues of salient contours, color and texture coherence, and small perimeter prior.

Detection: An association of the segmented object to one of the object models and a cost of this association based on the shape similarity to the model. This gives us a detection and quantification of this detection.

We attempt to solve for object *segmentation* and *detection* simultaneously. This is motivated by the fact that when it comes to global holistic shape we cannot evaluate the object-model similarity before we have segmented the object. However, segmenting an object assumes an already correct detection. The resulting 'chicken-egg' problem can be naturally addressed by solving for both at the same time.

We propose the **Bo**undary **S**tructure **S**egmentation (BoSS) model, which addresses the problem of detection and segmentation simultaneously in a unified framework. BoSS allows a concise formulation as an integer quadratic program, consisting of two terms a boundary structure matching term defined over superpixel boundaries, and a perceptual grouping term defined over superpixels. The terms are coupled via linear constraints relating the superpixels with their boundary. The resulting optimization problem is solved using a Semidefinte Programming relaxation and yields shape similarity and figure/ground segmentation in a single step.

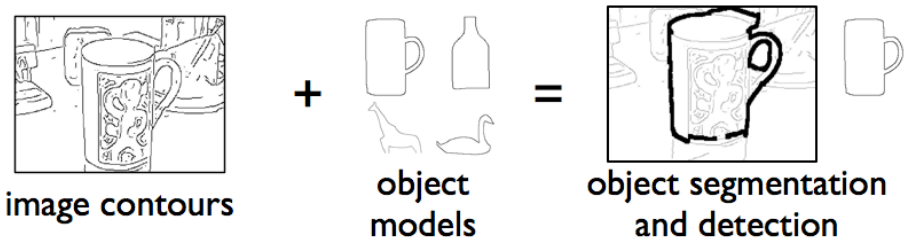


Figure 3.2: Using a cluttered image and a set of .

3.2 Chordiogram Parameterization

In order to relate the chordiogram to image segmentation, we should be able to parameterize it with segment and segment boundary indicator variables.

Oversegmentation. As a starting point for our method, we assume that we have an oversegmentation of the input image. The property we require from the thus obtained segments is that they do not cross object boundaries. In this way, every object in the image is representable as a set of such segments.

The advantages of using oversegmentation are twofold:

- Since the object is representable as a few segments which should be chosen from at most several hundred segments, one needs to reason over the space of segments instead of pixels which leads to a computational speedup.

- Segments are results of a bottom-up perceptual grouping process and thus can be expected to be coherent regions. Thus they represent a stable coarsening of the search space.

Oversegmentation as a preprocessing was first used for human pose estimation [Mori, 2005] and is widely used for object and scene segmentation. To obtain the segments one can use any segmentation algorithm such as Mean Shift [Comaniciu and Meer, 2002], Felzenszwalb's graph based segmentation

[Felzenszwalb and Huttenlocher, 2004], or Normalized Cuts [Shi and Malik, 2000], [Cour et al., 2005].

Segment parametrization. For each segment k obtained via the oversegmentation we introduce a segment indicator variable $s_k \in \{-1, 1\}$:

$$s_k = \begin{cases} 1 & \text{segment } k \text{ is foreground} \\ -1 & \text{otherwise} \end{cases} \tag{3.1}$$

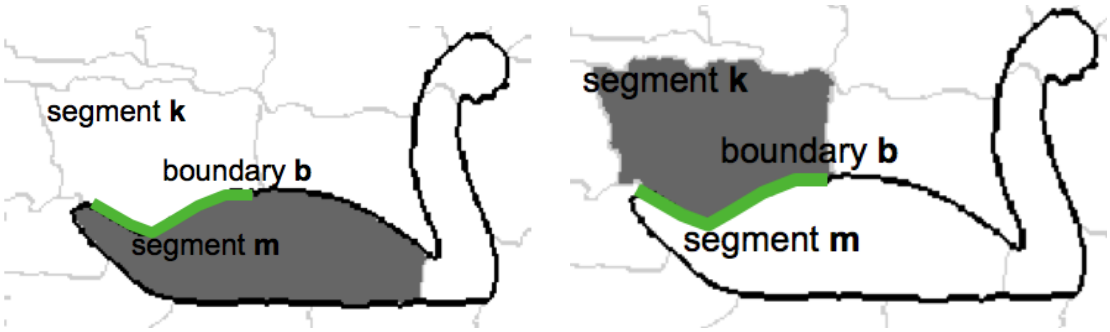
Further, denote by N the number of segments.

Segment boundary parameterization. We denote by $\mathcal{B}$ the set of all boundaries between pairs of neighboring segments, where the number of such boundaries is $M = |\mathcal{B}|$. Note that a contour b is a boundary because exactly one of its neighboring segments k and m is foreground and the other is background (see Fig. 3.3). To differentiate between those two cases, we include in $\mathcal{B}$ for each contour b two boundaries: b^m and b^k. The first denotes the case when m is foreground and k is background; the second case denotes the opposite case.

We introduce boundary indicator variables which indicate whether a segment boundary is an object boundary. This variable not only captures the state of the boundary but tries to explain which segment configuration causes this state. More precisely, for each boundary $b^k \in \mathcal{B}$ we introduce a boundary indicator variable $t_b^k \in \{0, 1\}$:

$$t_b^k = \begin{cases} 1 & \text{segment } k \text{ is foreground} \\ & \text{and segment } m \text{ is background} \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

As a result, there are two variables associated with each boundary. If the boundary designates an object boundary, then exactly one of the variables has value 1. Otherwise both are 0. The relationship between the values of the boundary and



(a) Segment m is foreground (b) Segment k is foreground

Figure 3.3: There are two cases in which boundary b can be an object boundary.

Boundary		Segments	
t_b^k	t_b^m	s_k	s_m
1	0	1	-1
0	1	-1	1
0	0	1	1
0	0	-1	-1

Table 3.1: We present the relationship between boundary and segment indicator variables.

segment variables is summarized in Table 3.1. This relationship can be expressed in terms of two constraints:

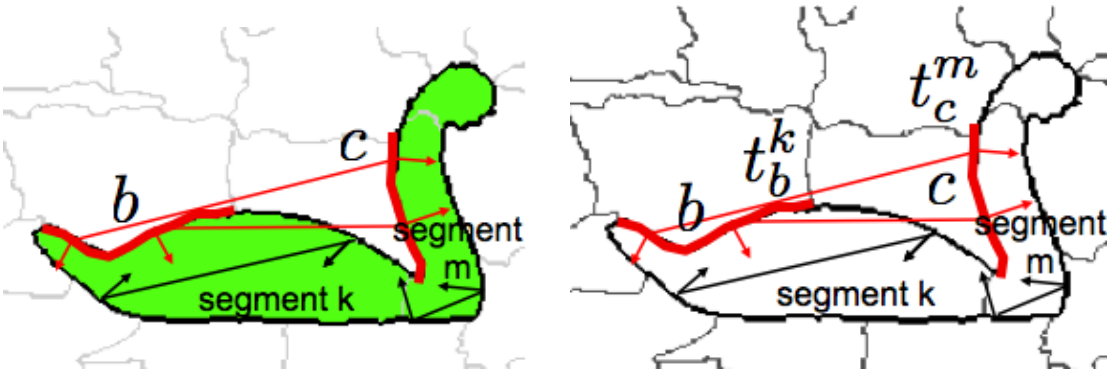
$$t_b^k - t_b^m = 1/2(s_k - s_m) \tag{3.3}$$

$$t_b^k t_b^m = 0 \tag{3.4}$$

Chordiogram additivity. To parameterize the chordiogram using the above variables, it will prove useful to provide an equivalent definition to Eq. (2.1). For a given segmented object, the chords connecting points on two boundaries b and c, caused by segments k and m being foreground respectively, can be described by a chordiogram $\mathrm{ch}_{bc}^{km} \in \mathbb{R}^K$, $b^k, c^m \in \mathcal{B}$:

$$(\mathrm{ch}_{bc}^{km})_l = \#\{(p,q) | f_{pq} \in bin(l), p \in b^k, q \in c^m\} \tag{3.5}$$

The above quantity can be considered as boundary-pair chordiogram. Note that the boundary-pair chordiogram is a subset of the overall chordiogram. Then Eq. (2.1) can



(a) Segmented object (b) Unknown object

Figure 3.4: The chordiogram of an object can be decomposed in terms of chordiograms which relate pair of boundaries, as shown on the left. If the object is not segmented, the boundaries can be selected via the boundary indicator variables.

be expressed as a sum of all boundary-pair chordiograms for all pairs of boundaries. This has the following linear form (see Fig. 3.4, left):

$$\mathrm{ch} = \sum_{b^k, c^m \in \mathcal{B}} \mathrm{ch}_{bc}^{km} \tag{3.6}$$

The above decomposition will be referred to as *chordiogram additivity* – the descriptor can be expressed in an additive form in terms of relations between object parts. Note that this is not a contradiction to the holistic nature of the descriptor since the additive components are *not* object parts, but configurations between parts.

Chordiogram parameterization. If we do not have a segmented object, we can select the object boundaries using the indicator variables (see Fig. 3.4, right) and express the resulting image chordiogram as follows:

$$\mathrm{ch}(t) = \sum_{b^k, c^m \in \mathcal{B}} \mathrm{ch}_{bc}^{km} t_b^k t_c^m \tag{3.7}$$

The value of the l^{th} bin can be expressed as a quadratic function:

$$\mathrm{ch}(t)_l = \sum_{b^k, c^m \in \mathcal{B}} (\mathrm{ch}_{bc})_l t_b^k t_c^m = t^T Q_l t \tag{3.8}$$

for a matrix Q_l which contains the values of the boundary-pair chordiogram: $(Q_l)_{bk;cm} = (\mathrm{ch}_{bc})_l$.

Note that in the above chordiogram parameterization one needs to indicate not only the boundary but also its relationship to the neighboring segments. This information is already contained in the chordiogram, since as defined in Sec. 2.2.1, each chord captures the object interior via the orientation of the normals.

Note that the above formulation is quadratic in the number of segment boundaries. However, the original problem of selecting a set of such boundaries, which comprise the object outline, is exponential in the number of boundaries.

3.3 Shape Matching

After we have parameterized the chordiogram in terms of the boundary indicators (see eq. 3.7), we chose to compare it with the model $\mathrm{ch}^{\mathrm{model}}$ using L_1 distance:

$$match(t, m) = ||\mathrm{ch}^{\mathrm{model}} - \mathrm{ch}(t)||_1 \tag{3.9}$$

The above shape matching cost evaluates the shape similarity between a model and a particular selection of segment boundaries. This motivates us to formulate the problem of shape matching as minimization of the above cost while taking into account the relation between boundaries and segments, as expressed in constraints in Eq. 3.3:

$$(SM): \qquad \min_{t,s} ||\mathrm{ch}^{\mathrm{model}} - \mathrm{ch}(t)||_1 \tag{3.10}$$

$$\text{subject to} \quad t_b^k - t_b^m = \frac{1}{2}(s_k - s_m) \quad \text{for all } b^m, b^k \in \mathcal{B} \tag{3.11}$$

$$t_b^k t_b^m = 0 \tag{3.12}$$

$$t \in \{0,1\}^{2M}, s \in \{-1,1\}^N \tag{3.13}$$

Solving the above optimization problem will result in

Segmentation: The opimal values of the boundary and segment indicators gives the object interior and boundary.

Shape-based detection cost: The minimum of the objective function quantifies the quality of match based on shape similarity.

Therefore, we will perform shape-based detection while at the same time we segment the object.

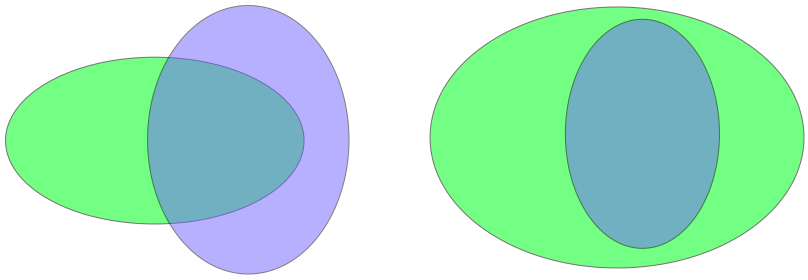
Region-boundary constraints. Many of the contour-based shape matching approaches reason directly over image contours [Ferrari et al., 2009, Ferrari et al., 2008, Lu et al., 2009, Felzenszwalb and Schwartz, 2007]. These works assume contours as a result of bottom-up contour grouping preprocessing step. The use of pre-grouped contours leads to more stable solution and reduces the computational complexity. However, those approaches ignore region information. In the work closest to ours, Zhu et al. [Zhu et al., 2008] rely on bottom-up grouping by selecting object boundaries from a set of long salient contours [Zhu et al., 2007].

In our model from Eq. 3.10 we use segments as additional constraints. This results in a solution which has the following properties with respect to the segment and boundaries:

Segments: A set of regions.

Boundaries: A set of *non-intersecting*, *non-including*, *closed* contours (see Fig. 3.5).

These requirements present a perceptual prior which reduces the search space and thus leads to a more stable solution. The latter requirements on the detected image boundaries are natural. The extracted contours should build a closed boundary, since they represent an object outline. Moreover, an object outline is not self-intersecting and the object interior should not include portions of the outline (non-including).



(a) Intersecting contours (b) Including contour

Figure 3.5: Undesirable contour configurations.

3.4 Perceptual Grouping

Our model can express grouping principles relating regions as well as boundaries.

Region grouping principles. While matching the input image to a model, we need to assure that the resulting figure represents a *perceptually salient segmentation*, i. e. the resulting figure should be a coherent region or set of regions distinct from the background. This property can be expressed using the segment indicator variable, as introduced in Sec. 3.2, and Min-Cut smoothness criterion. If we denote by $w_{e,g}$ the similarity between the appearance of superpixels e and g, then we can express the above region condition by the standard graph cut score:

$$group_r(s) = -s^T W s = -1^T W 1 + 2 \sum_{\substack{e \in \text{figure} \\ g \in \text{ground}}} w_{e,g} \quad \text{for } s \in \{1, 1\} \tag{3.14}$$

where the first term is constant.

Boundary grouping principles. In many cases an edge/contour detector cannot detect all object boundaries since there is no evidence in the image (see Fig. 3.6, right). However, if we use segmentation we can hallucinate object boundaries and recover the missing ones (see Fig. 3.6, left). This comes with the danger that one can hallucinate also non-existing objects in the maze of segment boundaries.

Figure 3.6: Left: input image. Middle: if we use all segment boundaries, than non-existing objects can be easily hallucinated. Right: if we rely on an edge/contour detection, then we can miss correct boundaries, which the segmentation can potentially hallucinate.

To address this issue we propose to use all segment boundaries, while at the same time incurring a cost if we choose hallucinated ones. In this way we will be able to complete the bottom of the bottle in Fig. 3.6 by paying a small cost, while we will never detect the apple since the cost for hallucinating all boundaries will be prohibitively large.

For a boundary segment b, we denote by c_b the percent of the pixels of b *not covered by image edges* extracted using thresholded Pb [Martin et al., 2004]. Then the boundary cost is defined as

$$group_b(t) = c^T t = \sum_{b \in \mathcal{B}} \sum_{k \in \mathcal{N}(b)} c_b t_b^k \quad \text{for } t_b^k \in \{0,1\}^N \tag{3.15}$$

3.5 BoSS Model

The BoSS model combines the costs from the previous sections. It solves for a shape match using cost (3.9) from Sec. 3.3, while at the same time applies grouping

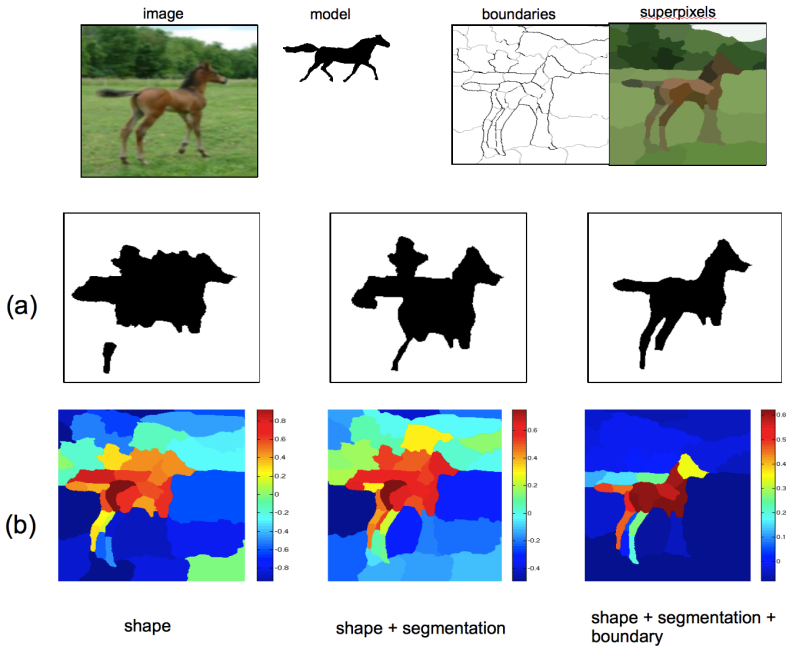


Figure 3.7: For an input image and model, as shown in the first row, our algorithm computes an object segmentation displayed in (a) row. We present three solutions by using only the matching term from Eq. (3.9) in first column; the matching term together with the superpixel segmentation prior (see Eq. 3.14) in second column; and the whole cost function consisting of the matching, segmentation and the boundary term in third column. (b) We also show for the three cost combinations the relaxed values of the segmentation variable s, as explained in Sec. 3.6.

principles as formulated in costs (3.14) and (3.15) from Sec. 3.4:

$$(BoSS): \quad \min_{t,s} match(t,m) + \delta group_r(s) + \gamma group_b(t) \tag{3.16}$$

$$\text{s. t.} \quad t_b^k - t_b^m = \frac{1}{2}(s_k - s_m) \quad \text{for all } b^k, b^m \in \mathcal{B} \tag{3.17}$$

$$t_b^k t_b^m = 0 \tag{3.18}$$

$$t \in \{0,1\}^{2M}, s \in \{-1,1\}^N \tag{3.19}$$

where δ and γ are weights of the different terms. The difference to the program (SM) from Eq. (3.10) lies in the addition of two grouping terms.

Term contributions. We examine the contribution of each term of the model on one concrete example presented in Fig. 3.7. The shown results were obtained using the optimization described in Sec. 3.6. By using only the matching term we are able to localize the object and obtain a rough mask, which however extends the back of the horse and ignores its legs (first column). The inclusion of the superpixel grouping bias helps to remove some of the erroneous superpixels above the object which have a different color than the horse (second column). Finally, if we add the boundary term, it serves as a sparsity regularization on t and results in a tighter segmentation (third column). Thus, the incorrect superpixels above the horse get removed, since they contain hallucinated boundaries not supported by edge response. Additionally, it recovers some of the legs, since they exibit strong edge response along their boundary.

3.6 Inference

Both the Shape Matching problem formulated as an integer quadratic program (SM) in Eq. (3.10) and (BoSS) from Eq. (3.16) are NP-hard. This not surprising since it is the problem of selecting from a set of exponentially many segments such that the resulting region has a desired shape and perceptual properties. To compute an approximate solution, we apply the Semi-definite Programming (SDP) relaxation [Goemans and Williamson, 1995, Boyd and Vandenberghe, 2004]. Since the latter program is a superset of the former, we present an optimization scheme for (BoSS) only.

First, we re-write the objective as a linear function and a set of quadratic constraints. We introduce for the l^{th} bin a variable β_l, which denotes the difference of the model and image chordiogram at this bin. Then the objective of (BoSS) can be

expressed in terms of β and a quadratic constraint for each bin:

$$\min_{t,s,\beta} 1^T\beta - \delta s^T W s + \gamma c^T t \tag{3.20}$$

$$\text{subject to} \quad t^T Q_l t - \mathrm{ch}_l^{\mathrm{model}} \leq \beta_l \tag{3.21}$$

$$\mathrm{ch}_l^{\mathrm{model}} - t^T Q_l t \leq \beta_l \tag{3.22}$$

$$t_b^k - t_b^m = \frac{1}{2}(s_k - s_m) \quad \text{for all } b^k, b^m \in \mathcal{B}$$

$$t_b^k t_b^m = 0$$

$$t \in \{0,1\}^{2M}, s \in \{-1,1\}^N \tag{3.23}$$

where in the first two constraints (3.21) and (3.22) we use the chordiogram parameterization as defined in Eq. (3.8).

To apply the SDP relaxation, we introduce variables T and S, which bring both the quadratic terms (3.21) and (3.22) into linear form: $T = tt^T$; and the quadratic terms in (3.20) into linear form: $S = ss^T$. This allows us to state the relaxation as follows:

$$(BoSS_{sdp}): \qquad \min_{t,s,\beta} 1^T\beta - \delta tr(W^T S) + \gamma c^T t \tag{3.24}$$

$$\text{subject to} \quad tr(Q_l^T T) - \mathrm{ch}_l^{\mathrm{model}} \leq \beta_l$$

$$\mathrm{ch}_l^{\mathrm{model}} - tr(Q_l^T T) \leq \beta_l$$

$$t_b^k - t_b^m = \frac{1}{2}(s_k - s_m) \quad \text{for all } b^k, b^m \in \mathcal{B}$$

$$T_{bk;bm} = 0 \quad \text{for all } b^k, b^m \in \mathcal{B}$$

$$t_b^k = T_{bk;bk} \quad \text{for all } b^k \in \mathcal{B} \tag{3.25}$$

$$diag(S) = 1_n \tag{3.26}$$

$$\begin{pmatrix} T & t \\ t^T & 1 \end{pmatrix} \succeq 0 \tag{3.27}$$

$$\begin{pmatrix} S & s \\ s^T & 1 \end{pmatrix} \succeq 0 \tag{3.28}$$

The above problem was obtained from problem (3.20) in two steps. First, we relax the constraints $T = tt^T$ to $T \succeq tt^T$ and $S = ss^T$ to $S \succeq ss^T$ respectively, which by

Schur complement are equivalent to (3.27) and (3.28) [Boyd and Vandenberghe, 2004]. Second, we weakly enforce the domain of the variables from the constraint (3.23). The $-1/1$-integer constraint on s is expressed as diagonal equality constraint on the relaxed S (see Eq. (3.26)), which can be interpreted as bounding the squared value of the elements of s by 1.The 0/1-integer constraint (see Eq. (3.25)) is enforced by requiring that the diagonal and the first row of T have the same value. Since $T = tt^T$, this has the meaning that the elements of t are equal to their squared values, which is true only if they are 0 or 1. Finally, the boundary-region constraints, one of which is quadratic, naturally translate to linear constraints.

The above problem is a liner program with inequality constraints in the cone of positive semi-definte matrices. As such, it is convex and can be solved exactly with any standard optimization package which supports such problems.

Discretization. Discrete solutions are obtained by thresholding s. Since s has N elements, there are at most N different discretizations, all of which are ranked using their distance to the model. If a threshold results in a set of several disconnected regions, we consider all possible subsets of this set. The algorithm outputs the top 5 ranked non-overlapping masks. Note that we are capable of detecting several instances of an object class since they result in several disconnected regions which are evaluated independently.

BoSS algorithm. The BoSS algorithms starts with an input image and a set of models. It solves the above optimization problem for each image-model pair at each scale. The best matching model gives the object segmentation as well as a detection cost – the chordiogram distance of the model to the obtained segmentation. The final algorithm is presented in Algorithm 2.

Algorithm 2 BoSS algorithm.

Input: model masks $m_1, \ldots, m_k$; image segmentation parametrized by t and s; scales $h_1, \ldots, h_p$.
Initialize: segmentations $S \leftarrow \emptyset$ and their detection costs $D \leftarrow \emptyset$.
for $i = 1 \ldots k$ **do**
  for $j = 1 \ldots p$ **do**
    $m_i^j \leftarrow$ rescale m_i to scale h_j:
    Compute $\mathrm{ch}_{i,j}^{\mathrm{mod}}$ of m_i^j at scale h_j using Eq. (2.1).
    Solve relaxed BoSS problem (3.24) using $\mathrm{ch}_{i,j}^{\mathrm{model}}$.
    Discretize to obtain segmentation $s_{i,j}$; $S \leftarrow S \cup \{s_{i,j}\}$.
    Compute $\mathrm{ch}_{i,j}$ from $s_{i,j}$ at scale h_j using Eq. (2.1).
    Compute detection cost: $d_{i,j} \leftarrow \left|\left| \frac{\mathrm{ch}_{i,j}}{||\mathrm{ch}_{i,j}||} - \frac{\mathrm{ch}_{i,j}^{\mathrm{model}}}{||\mathrm{ch}_{i,j}^{\mathrm{model}}||} \right|\right|_1$.
  end for
end for
$(i^*, j^*) \leftarrow \arg\min_{i,j} d_{i,j}$.
Output: segmentation s_{i^*,j^*} and detection cost d_{i^*,j^*}.

3.7 Experiments

In this section we show detection and segmentation results on several established benchmarks.

Implementation details. We use translation invariant chordiograms with $b_l = 4, b_r = 8, b_n = 8$, resulting in 2048-dimensional descriptor.

To obtain superpixels we oversegment the image using NCuts [Cour et al., 2005] with $n = 45$ segments. The grouping cues used to define the affinity matrix W^{pixels} are color and intervening contours [Yu and Shi, 2003] based on Pb [Martin et al., 2004]. To define the segmentation term (3.14) in our model we can use any affinity matrix. We choose to use the same grouping cues as for segmentation above. For each pair of superpixels k and m we average the pixel affinities to obtain an affinity matrix over the superpixels: $W_{km}^{superpixels} = \frac{1}{a_k a_m} \sum_{p \in k, q \in m} \widehat{W}_{pq}^{pixels}$, a_k and a_m being the number of pixels contained in k and m respectively. Above, $\widehat{W}^{pixels}$ is obtained from the top n eigenvectors E of W^{pixels}: $\widehat{W}^{pixels} = E \Lambda E^T \approx W^{pixels}$, where Λ are the corresponding eigenvalues. This low-rank approximation represents a smoothed version of the

original matrix and reduces the noise in the original affinities. Finally, the weights of the term in Eq. (3.16) were chosen to be $\delta = 0.01$ and $\gamma = 0.6$ on five images from ETHZ dataset and held constant for all experiments.

For the optimization we use SeDuMi [Sturm, 1999] which is based on the Primal-Dual Interior Point Method. To compute the number of variables in the SDP, one can assume that each superpixel has at most C neighboring superpixels. Hence we obtain $M = Cn$ boundary variables. Thus, the total variable number in the relaxed problem is bounded by $n^2 + C^2n^2 \in O(n^2)$. In our experiments, we have $n = 45$ and the value of C is less than 5 which results in less than 200 boundary segment variables. The segmentation of an image takes $5 - 15$ secs on a 3.50 GHz processor.

3.7.1 ETHZ Shape Dataset

Dataset. The ETHZ Shape Dataset [Ferrari et al., 2009] consists of 255 images of 5 different object classes — Applelogos (40 images), Bottles (48 images), Mugs (48 images), Giraffes (87 images) and Swans (32 images). The dataset is designed such that most of the object do not have a distinctive appearance and the only representation, which can be used to detect them, is their outline. As a result, this dataset has been widely used for evaluation of shape-based detection methods. Some of the challenges in this dataset are highly cluttered images – in the background as well as internal spurious contours; wide variation of object scale; multiple instances of an object in the same image. However, the depicted objects are fully included in the images and are not occluded. Also, the used objects vary in shape but are not articulated (the giraffe's legs are not detected).

We apply the BoSS model using hand-drawn object outlines as shape models, one model per class. These models were supplied with the dataset. We use 7 different scales, such that the scale of the model, defined as the diameter of its bounding box, range from 100 to 300 pixels. We use non-maximum suppression – for every two hypotheses, whose bounding boxes overlap by more than 50%, we retain the one

with the higher score and discard the other one.

Detection results. On the ETHZ Shape Dataset we achieve 89.2%/90.5% detection rate at 0.3/0.4 fppi using Pascal criterion[1] and 93.4%/94.2% under 20% overlap criterion[2], as reported in Table 3.2 and Fig. 3.9-3.8.

Reranking In order to compare with approaches on the ETHZ Shape Dataset which use supervision, we use weakly labeled data to rerank the detections obtained from BoSS. We use only the labels of the training images to train a classifier but not the bounding boxes. This classifier can be used to rerank new hypotheses obtained from BoSS.

More precisely, we use half of the dataset as training and the other half as test (we use 5 random splits). We use BoSS to mine for positive and negative examples. The top detection in a training image using a model which represents the label of that image is considered a positive example; all other detections are negative examples. The chordiograms of these examples are used as features to train one-vs-all SVM [Joachims, 1999] for each class. During test time, each detection is scored using the output of the SVM corresponding to the model used to obtain this detection. Note that this is a different setup of supervision which requires less labeling – while we need one hand-drawn model per class to obtain detections via BoSS, we do not use the bounding boxes but only the labels of the training images to score them. We argue that the effort to obtain a model is constant while segmenting images by hand is much more time consuming.

The results are shown in Table 3.2. The weak supervision leads to 94.3%/96.0% detection rate under Pascal criterion, which is an improvement of approx. 5% over BoSS. It is attributed to the discriminatively learned weights of the chordiogram's bins. This corresponds to discriminatively learning object shape variations and

[1]Pascal criterion: the intersection of the hypothesis and ground truth bounding boxes overlap more than 50% with the union of both; 20% overlap detection criterion: the intersection of the hypothesis and ground truth bounding boxes overlap more than 20% with the each of them.

	Algorithm	Apple logos	Bottles	Giraffes
20% over.	BoSS†	86.4%/88.6%	**96.4**%/**98.2**%	**97.8**%/**97.8**%
	[Lu et al., 2009]$^{\dagger\sharp}$	**92.5**%/**92.5**%	95.8%/95.8%	86.2%/92.0%
	[Fritz and Schiele, 2008]*	-/89.9%	-/76.8%	-/90.5%
	[Ferrari et al., 2009]†	84.1%/86.4%	90.9%/92.7%	65.6%/70.3%
Pascal crit.	BoSS†	86.4%/88.6%	96.4%/96.4%	81.3%/86.8%
	BoSS + reranking*	100%/100%	96.3%/97.1%	86.1%/91.7%
	[Maji and Malik, 2009]*	95.0%/95.0%	92.9%/96.4%	89.6%/89.6%
	[Srinivasan et al., 2010]*	95.0%/95.0%	100%/100%	87.2%/89.6%
	[Gu et al., 2009]*	90.6%/-	94.8%/-	79.8%/-
	[Ravishankar et al., 2008]$^{\dagger\circ}$	95.5%/97.7%	90.9%/92.7%	91.2%/93.4%

	Algorithm	Mugs	Swans	Average
20% over.	BoSS†	**84.8**%/86.4%	93.4%/93.4%	**91.2**%/93.0%
	[Lu et al., 2009]$^{\dagger\sharp}$	83.3%/92.0%	**93.8**%/**93.8**%	90.3%/**93.2**%
	[Fritz and Schiele, 2008]*	-/82.7%	-/84.0%	-/84.8%
	[Ferrari et al., 2009]†	80.3%/83.4%	90.9%/93.9%	82.4%/85.3%
Pascal crit.	BoSS†	72.7%/77.3%	93.9%/93.9%	86.1%/88.6%
	BoSS + reranking*	90.1%/91.5%	98.8%/100%	94.3%/96.0%
	[Maji and Malik, 2009]*	93.6%/96.7%	88.2%/88.2%	91.9%/93.2%
	[Srinivasan et al., 2010]*	93.6%/93.6%	100%/100%	95.2%/95.6%
	[Gu et al., 2009]*	83.2%/-	86.8%/-	87.1%/-
	[Ravishankar et al., 2008]$^{\dagger\circ}$	93.7%/95.3%	93.9%/96.9%	93.0%/95.2%

Table 3.2: Detection rates at 0.3/0.4 false positives per image, using the 20% overlap and Pascal criteria. We achieve state of the art results on all categories under the first detection criterion. Under the Pascal criterion, we achieve state of the art rates on the dataset as well. For Applelogos, Swans and Bottles, the results are equal to the ones using the weaker criterion. This is due to the exact localization, which can be achieved when segmenting the object. For Giraffes and Mugs results are slightly lower due to imperfect segmentation (some segments leak into the background or miss parts) – the detections which are correct under the weaker 20% overlap criterion, are not counted as correct under the Pascal criterion. However, there are correctly segmented objects under the Pascal criterion which are ranked lower. The employed reranking helps to recover some of them. († use only hand labeled models. * use strongly labeled training data with bounding boxes, while we use weakly labeled data in the reranking, i. e. no bounding boxes. $^{\sharp}$ considers in the experiments only at most one object per image and does not detect multiple objects per image. $^{\circ}$ uses a slightly weaker detection criterion than Pascal.)

builds on the power of BoSS to deal with clutter.

Segmentation In addition to the detection results, we evaluate the quality of the detected object boundaries and object masks. For evaluation of the former we follow the test settings of [Ferrari et al., 2009][2]. We report recall and precision of the detected boundaries in correctly detected images in Table 3.3. We achieve higher recall at higher precision compared to [Ferrari et al., 2009]. This is mainly result of the fact that BoSS attempts to recover a closed contour and in this way the complete object boundary. These statistics show that the combination of shape matching and figure/ground organization results in precise boundaries ($> 87\%$ for all classes except Giraffes). The slightly lower results for Giraffes is due to the legs which are not fully captured in the provided class models. We also provide object mask evaluation as percentage of the image pixels classified incorrectly by the detected mask (see Table 3.3). For all classes we achieve less than 6% error, and especially classes with small shape variation such as Bottles and Applelogos we have precise masks ($< 3\%$ error).

3.7.2 INRIA Horses Dataset

Dataset. The INRIA horses dataset, has 340 images, half of which contain horses and the other half has background objects. This dataset presents challenges not only in terms of clutter and scale variation, but also in articulation, since the horses are in different poses, and partial occlusions.

We use 6 horse models representing different poses for the INRIA horse dataset (see Fig. 3.12). In these experiments we used 10 scales such that the scale of the model, defined as the diameter of its bounding box, range from 55 to 450 pixels. Similarly to the previous dataset, we use non-maximum suppression – for every two

[2]A detected boundary point is considered a true positive if it lies within t pixels of a ground truth boundary point, where t is set to 4% of the diagonal of the ground truth mask. Based on this definition, one computes recall and precision.

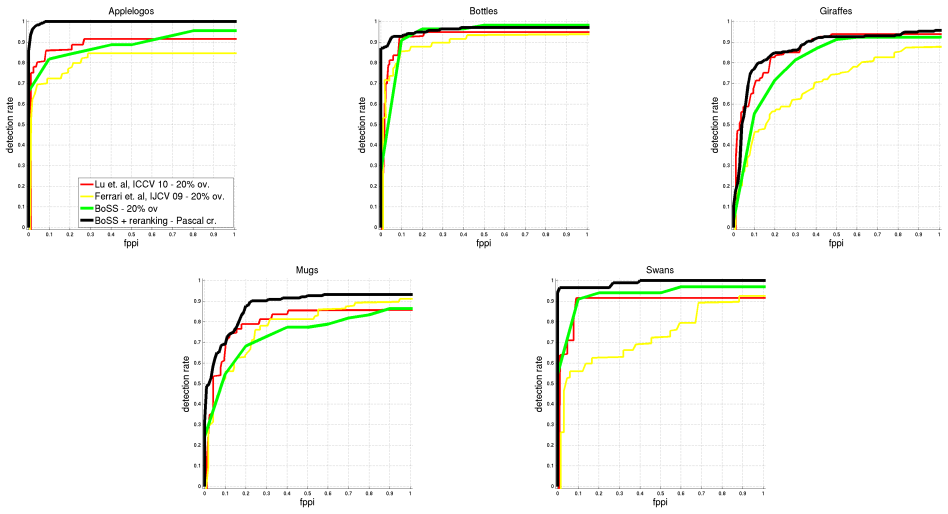


Figure 3.8: Results on ETHZ Shape dataset: detection rate vs false positives per image. Results using BoSS are shown using 20% overlap as well as after reranking using the stricter Pascal criterion. Both consistently outperform other approaches, evaluated using the weaker 20% overlap criterion.

hypotheses, whose bounding boxes overlap by more than 50%, we retain the one with the higher score and discard the other one.

Detection results. On INRIA Horses dataset, we achieve state of the art detection rate of 92.4% at 1.0 fppi (see Fig. 3.10). Examples of detections of horses in different poses, scales and in cluttered images are shown in Fig. 3.12.

3.7.3 Analysis of the Empirical Results

In Fig. 3.11 and Fig. 3.12 we show examples of typical detections in the datasets described above. Our method is capable of detecting objects of various scales in highly cluttered images, even when the object is small and most of the image contours and segments are not part of the object. Note that the translation invariance of the chordiogram allows us to find the object without having to search exhaustively for location. Additionally, the segmentation gives us a pixel-level object localization

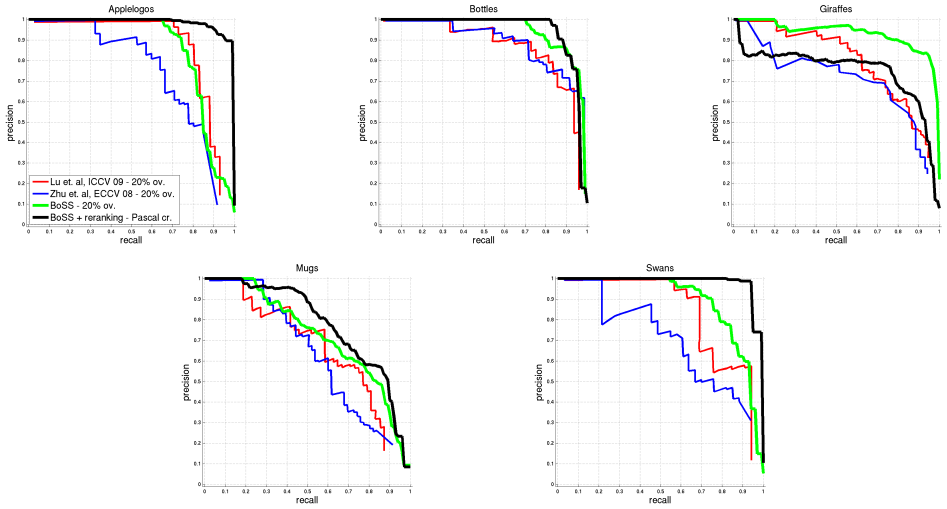
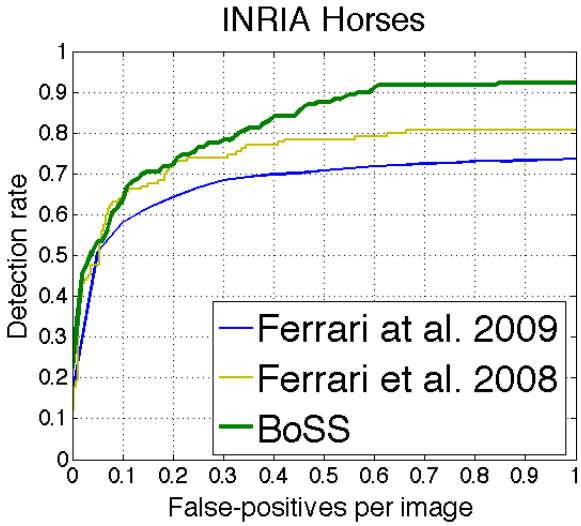


Figure 3.9: Results on ETHZ Shape dataset: precision recall curves. Results using BoSS are shown using 20% overlap as well as after reranking using the stricter Pascal criterion. Both consistently outperform other approaches, evaluated using the weaker 20% overlap criterion.



Method	Det. rate
BoSS	92.4%
[Maji and Malik, 2009]	85.3%
[Ferrari et al., 2008]	80.8%
[Ferrari et al., 2009]	73.8%

Figure 3.10: Detection rate vs false positives per image (fppi) for our and other approaches on INRIA Horse dataset.

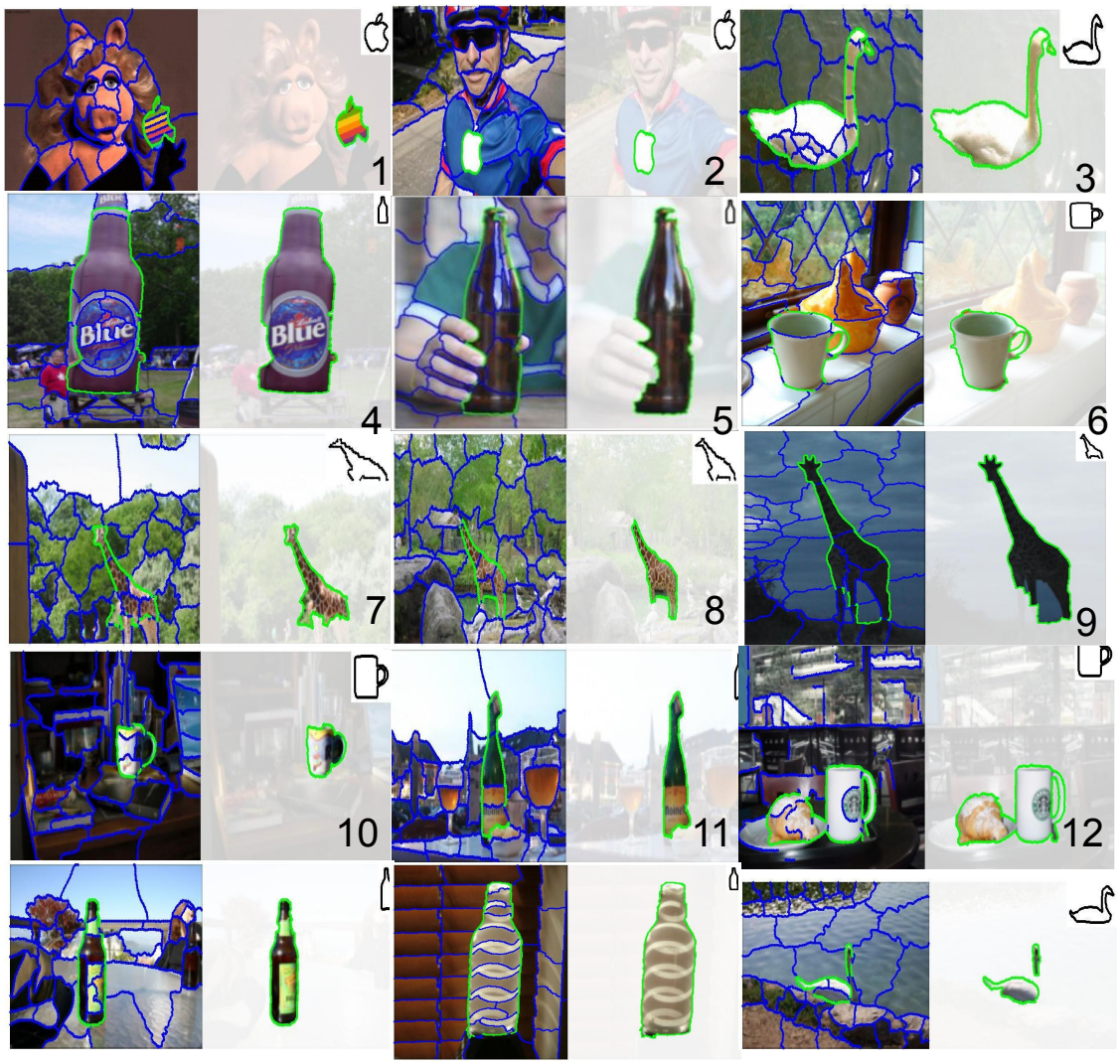


Figure 3.11: Example detection on ETZ Shape dataset. For each example, we show on the left side the selected superpixel boundaries, and on the right the selected object mask.

	boundary precision/recall			pixel error	
	SM	BoSS	[Ferrari et al., 2009]	SM	BoSS
Applelogos	91.9% 97.1%	91.8%/97.5%	91.6%/93.9%	2.0%	1.6%
Bottles	89.4%/91.1%	90.3%/92.5%	83.4%/84.5%	2.8%	2.7%
Giraffes	75.4%/81.3%	76.8%/82.4%	68.5%/77.3%	6.2%	5.9%
Mugs	77.7%/89.1%	86.5%/90.5%	84.4%/77.6%	5.5%	3.6%
Swans	81.0%/86.8%	85.8%/87.6%	77.7%/77.2%	6.7%	4.9%

Table 3.3: Precision/recall of the detected object boundaries and pixel classification error of the detected object masks for ETHZ Shape dataset. We present results using only the Shape Matching cost (see Eq. (3.3)) as well as the full cost – BoSS – which consists of shape matching as well as perceptual grouping terms (see Eq. (3.16)).

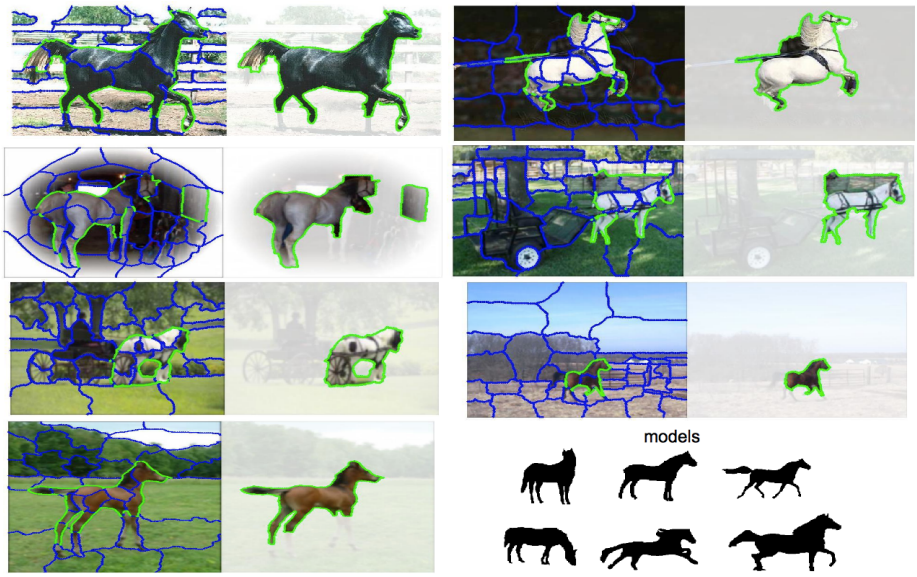


Figure 3.12: Examples of detections for INRIA horses dataset. For each image we show the selected superpixel boundaries on the left and the detected object segmentation on the right. Bottom right: 6 models used in the experiments.

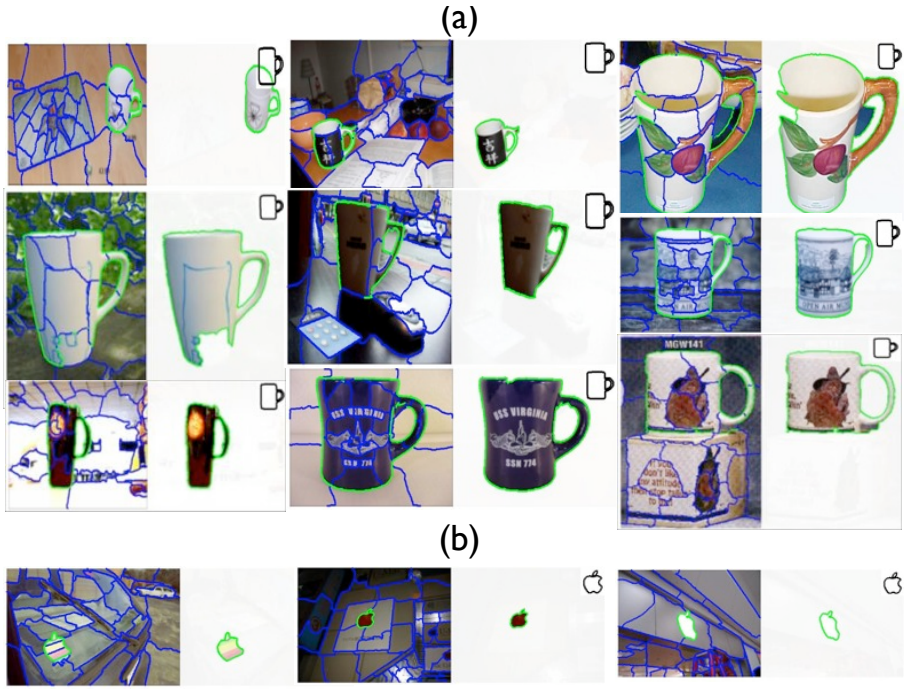
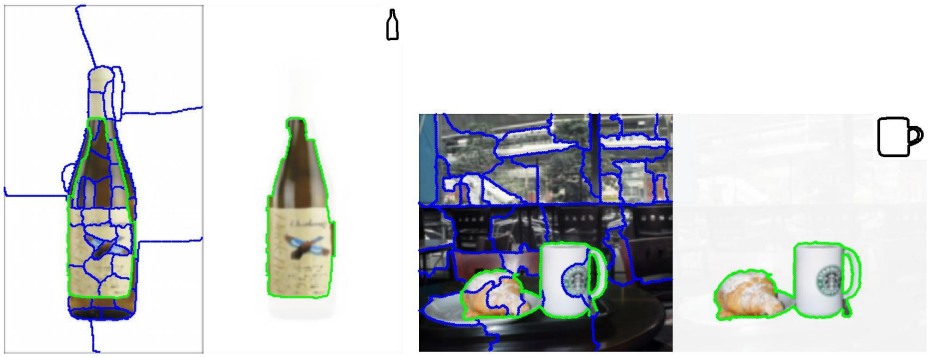


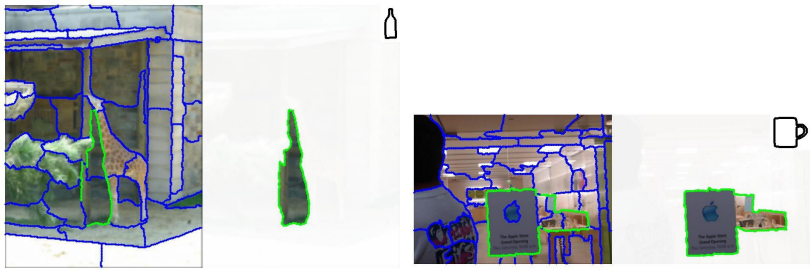
Figure 3.13: Example detection on ETHZ Shape dataset which show the robustness of the chordiogram and BoSS to shape variations. For each example, we show on the left side the selected superpixel boundaries, and on the right the selected object mask. We use the same model to obtain those detections. Note. however, that the detected mugs may have different aspect ratio, shape of the body (rectangle or cone), and shape and size of the handle.

which is much more precise compared to the bounding-box localization used by other methods.

Shape variations. Our approach is robust against local shape variations as well as global transformations. As shown in Fig. 3.13 (a), using a single mug model BoSS obtains detections of objects whose shape deviates from the model in various ways: aspect ration, global shape, shape of parts, etc. In addition, it tolerates global transformations as minor rotations and foreshortening (see Fig. 3.13 (b)).



(a) Inexact segmentation.

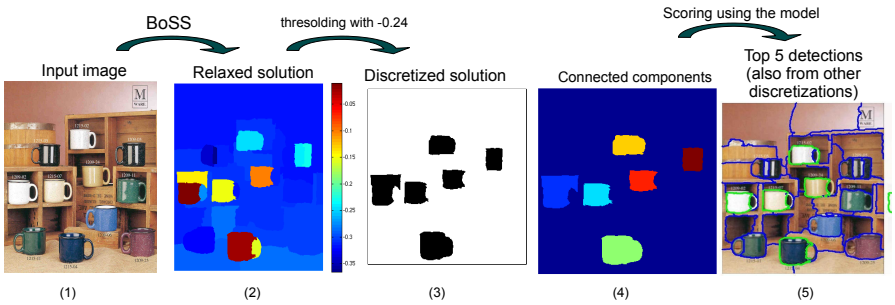


(b) Hallucinations.

Figure 3.14: Examples of missdetections.

Missdetections. The major sources for incorrect detections are accidental alignments with background contours, which we call hallucinations, and partially incorrect boundaries (see Fig. 3.14). The former cause shows the limitation of shape – one can sometimes find a constellation of contours which resemble the model outline. Some of those cases can be ruled out by using perceptual grouping principle. However, in other cases the lack of an appearance model is limiting.

Multiple detections per image. Our algorithm is capable of detecting multiple objects of the same class in an image. This is possible since the sum of the unnormalized chordiograms of all present objects should be a good match for the object model. Hence, a single invocation of the optimization problem should lead to a relaxed solu- tion which captures all detections. For example, after matching the



(a) Outline of the detection of multiple objects per image.

(b) Examples of multiple detections per image.

Figure 3.15: Detection of multiple instances of the same object class in an image.

image in (1) of Fig. 3.15, we obtain a relaxed solution (2) which contains the majority of the objects. During the discretization step we use all the different thresholds, which would give us different discretizations. We subsequently score them by matching them to the model. If a discretization contains several disconnected regions, we score all of them independently. For example, after using threshold -0.24 on (2), we obtain the descrete solution (3), which contains 6 disconnected regions shown in (4). After scoring each of them, we discard the left most mug and retain the remaining regions. The results are shown in (5) (note that we present in (5) the final result after matching over several scales and using all thresholds).

Influence of the different terms. The presented BoSS model (see Eq. (3.16) in Sec. 3.5) consists of top-down shape matching term and a bottom-up perceptual grouping term. Clearly, the former term is stronger since it contains class specific information. Additionally, it operates on pregrouped superpixels, which are a product of a bottom-up process. Therefore, the shape matching term alone (first term in

Figure 3.16: Examples of BoSS without perceptual terms (left) and with perceptual terms (right).

Eq. (3.16)) is not purely top-down.

To analyze the contribution of the perceptual terms, we apply BoSS on the ETHZ Shape Dataset without the perceptual terms (see program SM in Eq. (3.10)) and compare the resulting segmentations and object boundaries to the one obtained using the full BoSS model. The results are compared in Table 3.3. Although SM performs pretty comparable to the full model, its boundary and pixel precisions are slightly below the ones obtain via BoSS – on average SM has 4.6% pixel error, while BoSS reduces it to 3.7%. Perceptual grouping tends to correct shape-based segmentation in cases where the shape match is not very good, but the bottom-up grouping is based on a strong signal. For example, in the upper left image in Fig. 3.16 the bottle does not match well due to the occluding hand. Segmenting only the upper part of the bottle, however, will be penalized by the segmentation term since it encourages grouping of the whole bottle due to homogeneous appearance.

3.8 Analysis of BoSS

In the definition of the BoSS model we assume that we pre-segment the image into segments. All subsequent computations are executed over these segments. Therefore, it is natural to ask how does the quality of the segments and the noise in the image influence the performance of the presented algorithm?

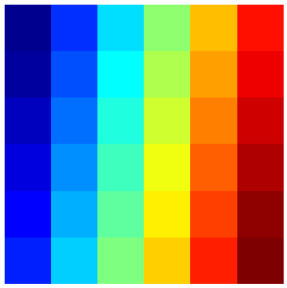
In particular, we will motivate the need for a good segmentation in the case of shape-based detection. The high quality of the segmentation is achieved through the BoSS model and its inference. Further, we will analyze the performance of the BoSS model in the presence of background clutter and object boundary noise. We will analyze the influence of the number of segments. Finally, we will motivate the choice of the chordiogram as a shape representation with the BoSS model.

3.8.1 Grid World Setup

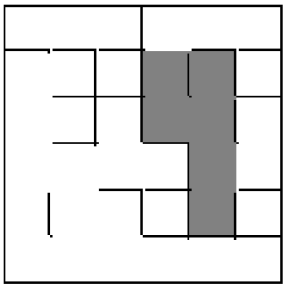
In the subsequent analysis we will use an idealized setup in which we can control the clutter and background noise. In particular, we will use an image whose segmentation is a grid of size $g \times g$. An object in this image is a connected group of grid cells (see Fig. 3.17). In this setup we will instantiate contours by selecting a set of segment boundaries to be real image contours. In this way we can introduce and control the following two image artifacts:

Image clutter: Each segment boundary which is not an object boundary can be a potential background or interior clutter contour. We denote by η_g the percentage of the clutter as the portion of non-object boundaries which are image contours.

Missing object boundaries: Some of the object boundaries will not be instantiated as image contours. This will simulate segmentation leakage. We denote by μ_g the portion of the object boundaries which are not image contours.



(a) Segmentation.



(b) Object in presence of clutter and missing object boundaries.

Figure 3.17: Grid World Setup.

By controlling the above two parameters we can obtain input images with varying clutter and missing object boundaries (see Fig. 3.18). In the following experiments we will use grid of size $g = 6$, clutter levels $\eta_g \in \{0\%, 20\%, 40\%, 60\%, 80\%\}$ and missing object boundary levels $\mu_g \in \{0\%, 20\%, 40\%\}$. We will generate 20 objects randomly and for each object we will create $5 \times 3 = 15$ images, one for each of the possible values of (η_g, μ_g).

In order to evaluate the quality of a segmentation, we use two metrics. The first one is the overlap error is defined in terms of missegmented pixels – pixels which are either segmented incorrectly as foreground or background:

$$\text{overlap error} = \frac{\text{\# of missegmented pixels}}{\text{\# of image pixels}}$$

This measure gives the quality of the image segmentation. It has, however, lower values for smaller objects. In order, to evaluate the segmentation of an object, we use a metric introduced in conjunction with the Pascal Visual Object Challenge [Everingham and et. al, 2005]:

$$\text{pascal overlap score} = \frac{\text{area of object and segmentation intersection}}{\text{area of object and segmentation union}}$$

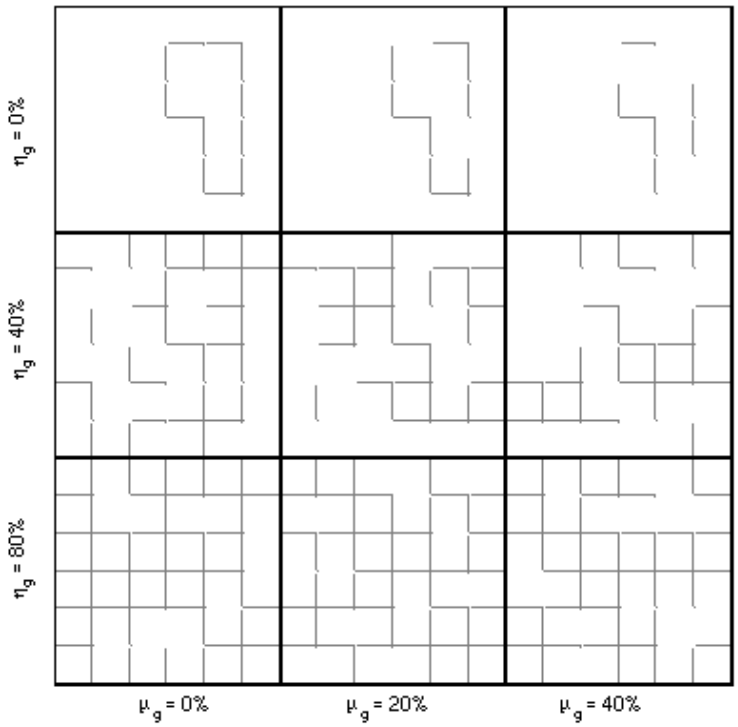


Figure 3.18: Input image contours for the object from Fig. 3.17 with different combinations of image clutter η_g and missing contours μ_g .

3.8.2 Importance of Segmentation for Shape-based Detection

Most of the applications of segmentation in computer vision serve as coarsening of the input space. In the case of general object recognition, one often computes texture-based descriptors for each segment [Shotton et al., 2009], groups of segments [Malisiewicz and Efros, 2008] or bag-of word descriptors of segments
[Russell et al., 2006]. In such approaches, a pre-segmentation is considered useful if a segment or groups of segments overlap sufficiently well with the object of interest. Therefore, using small groups of segments or multiple segmentations is often enough

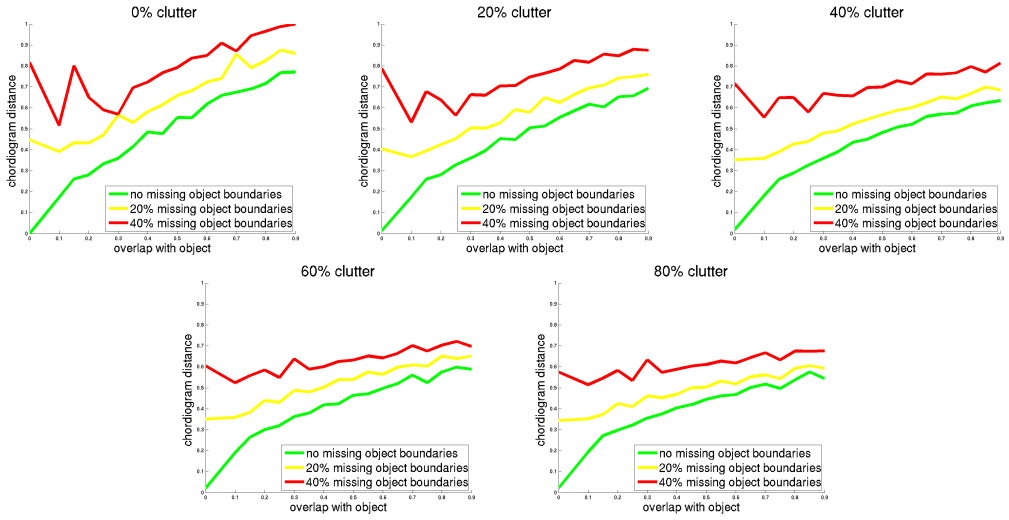


Figure 3.19: chordiogram distance versus overlap with the object. For each clutter level we plot three curves, each corresponding to one of the missing object boundary levels.

to capture an object.

In the case of shape-based object detection, it is important to capture the correct object boundaries in a segment selection. Therefore, even if the overlap of a segment or a group of segments with an object of interest is large, these segments may not capture the shape of the object at all.

To analyze this, we use the Grid World Setup. For all possible groups of segments we plot the chordiogram distance to the model versus the overlap error in Fig. 3.19. The chordiogram is normalized by the largest distance to show the relative degradation with growing overlap error. We present this analysis for all clutter and missing object boundary levels.

Even in the case when there are no missing object boundaries, we can see that the chordiogram distance degrades pretty quickly with increasing overlap error. On average, at overlap error 0.1 we have chordiogram distance 0.26, and at 0.2 the distance is 0.41. In other words, if a group of segments misses only 20% of the object, then the chordiogram distance degrades to 0.41 of the worst possible distance of any

segmentation	10	20	30	all
# of groups	289	1705	3343	5337

Table 3.4: Average number of groups of segments per image for each segmentation as well as the total number.

group of segments. This shows that even small overlap errors may affect the shape-based distance. This can be explained by the fact that even such a small overlap error may lead to missing important object boundaries.

As we can see in Fig. 3.19, the chordiogram distance degrades drastically as we have missing object boundaries. When 40% of the boundaries are missing, the object becomes unrecognizable since the chordiogram distance would not be sufficient to indicate how well a group of segments overlaps with the object.

This motivates the usage of an inference for the BoSS model which potentially can select any group of segments as an object segmentation. Thus, we do not exclude any potential grouping and do not limit ourselves to a set of segments obtained using a purely bottom-up process.

3.8.3 BoSS vs Multiple Segmentations

To see the importance of being able to select all possible groups of segments for real images, we compare the BoSS model to shape-based detection over segments computed via multiple segmentations.

More precisely, we use three different segmentations per image – using Normalized Cuts [Cour et al., 2005] with 10, 20, and 30 segments. For each segmentation, we compute groups of connected segments of up to 5 segments. This results in 5337 groups of segments per image on average. We consider each group of segments as a hypothesis for an object segmentation. To evaluate how likely a hypothesis is an object of a particular class, we compute the chordiogram distance between the hypothesis and the object model.

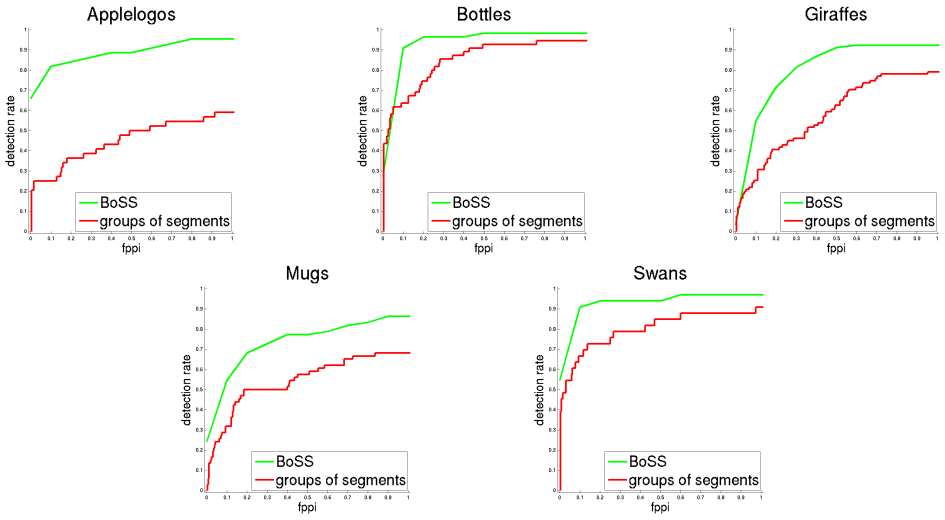


Figure 3.20: Detection rate vs false positives per image for all five classes of the ETHZ Shape Dataset computed using groups of segments.

The detection rates for the five classes of the ETHZ Shape Dataset are presented in Fig. 3.20 and Table 3.5. We can see that using only groups of segments, the detection rate drops, even drastically for some classes such as Applelogos and Mugs. The reason is that there are objects which cannot be segmented using groups of at most 5 segments. Of course, one can increase the size of the groups, however their number groups exponentially with their size. Therefore, it would become less feasible to compute the chordiogram for all groups of larger sizes. A different reason for the poor performance is that groups of segments approach causes more false positives.

3.8.4 Analysis of the BoSS Inference in Presence of Noise

The above grid world setup allows us to evaluate the quality of the segmentation obtained from the BoSS model. For each image and pair of clutter and missing object boundary levels, we try to segment the image using as a model the groundtruth object for this image. We use four different algorithms, which are based on the BoSS model:

1. BoSS model without any perceptual terms (see the shape matching model from

	Applelogos	Bottles	Giraffes
groups of segm.	38.6%/43.2%	85.5%/87.3%	46.2%/52.8%
BoSS	86.4%/88.6%	96.4%/96.4%	81.3%/86.8%
	Mugs	Swans	average
groups of segm.	50%/50%	78.8%/78.8%	59.1%/62.4%
BoSS	72.7%/77.3%	93.9%/93.9%	86.1%/88.6%

Table 3.5: Detection rates of group of segments and BoSS at 0.3 and 0.4 fppi for the five classes of the ETHZ Shape Datatset.

Sec. 3.3).

2. BoSS model with a boundary term only (see second principle from Sec. 3.4).

3. BoSS model (see Sec. 3.5). We add a segmentation term with affinity matrix W defined as $W_{e,g} = 1$ if both segments e and g belong to either foreground or background; $W_{e,g} = 0.73$ otherwise. Note that this term does not contain noise. However, it is relative weak since segments across the object boundary are considered relatively similar (with similarity 0.73).

4. Groups of segments. Since the grid world contains only g^2 cells, where g is relatively small, we can compute the chordiograms of all possible groups of segments. The segmentation under this model is the group of segments whose chordiogram is closest to the model chordiogram under the L_1 distance. We use only cells which do not touch the image boundary. As a result we have to generate $2^{(g-2)^2}$ groups of segments and compute their chordiograms. In our setup we use $g = 6$ which leads to 65536 groups of segments.

The first three algorithms are variations of the BoSS model. The last algorithm can be considered an exact solution of the chordiogram matching which is tractable for this small problem.

We evaluate the output of each algorithm by computing the overlap error of the obtained segmentation and the groundtruth object. The results of the algorithms

are presented in Fig. 3.21. In the case of no missing object boundaries, the exact algorithm performs almost perfectly, while all variations of the BoSS model achieve an error of less than 5%, even in the case when we have large clutter. This shows that our model is capable of dealing with clutter.

In case when we have missing object boundaries, all algorithms degrade slowly as the clutter increases. This is due to the fact that the clutter provides for accidental segmentations which may have a lower chordiogram distance than the true segmentation which is partially damaged.

In addition, we present the exact overlap errors averaged over all missing levels in Table 3.6 and over all clutter levels in Table 3.7. On average, the exact chordiogram distance computation through the last algorithm gives best segmentation. The worst performing algorithm is the one without perceptual terms.

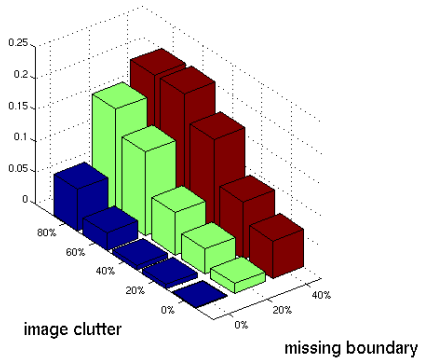
algorithm	0%	20%	40%	60%	80%	average
BoSS - no perceptual terms	2.6	4.6	7.8	12.1	14.6	8.4
BoSS - boundary term	1.3	3.3	8.4	11.5	17	8.3
BoSS - all perceptual terms	2.7	5.1	7.2	9.6	12.4	7.4
groups of segm.	1.7	3.6	7.4	8.3	10.5	6.3

Table 3.6: Missclassified pixels in the grid world under varying levels of clutter η_g.

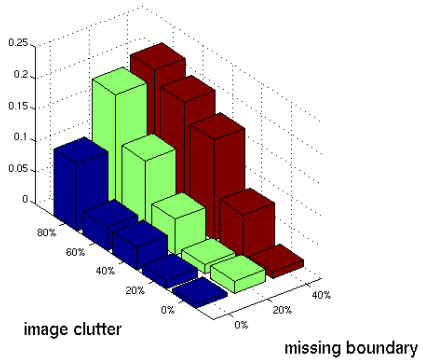
algorithm	0%	20%	40%	average
BoSS - no perceptual terms	2.2	8.5	14.3	8.4
BoSS - boundary term	4	8.1	12.8	8.3
BoSS - all perceptual terms	4.2	6.8	11.2	7.4
groups of segm.	0	5.6	12.7	6.3

Table 3.7: Missclassified pixels in the grid world under varying levels of missing object boundaries μ_g.

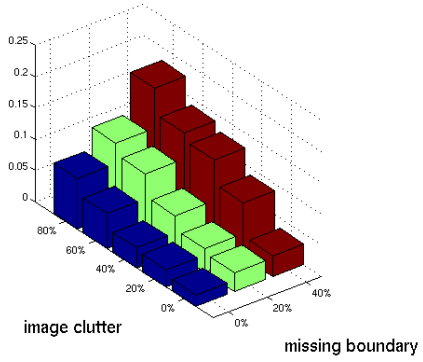
Adding boundary and segmentation terms helps to deal with the clutter and missing object boundaries. To see how exactly the perceptual terms help, consider



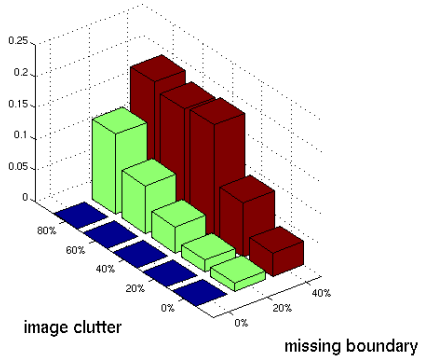
(a) BoSS w/o perceptual terms.



(b) BoSS w/ boundary term only.



(c) BoSS.

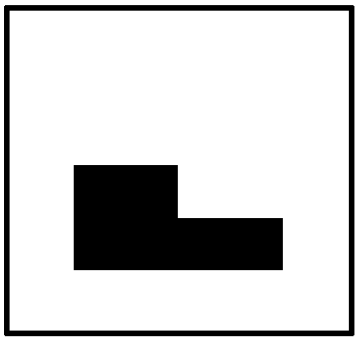


(d) Groups of segments.

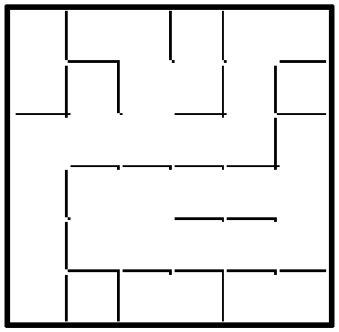
Figure 3.21: For each of the four algorithms (see text) we present the overlap error at all combinations of clutter level and missing object boundary level.

the examples in Fig. 3.22 and Fig. 3.23. In the first figure, we investigate the boundary term only. In this example, the object has two missing vertical boundaries on its right side. The relaxed segmentation tries to compensate by leaking in the upper right side of the image. Note that this leakage is not affected by many other non-vertical contours since most of the boundaries of this leakage are not supported by contours. As a result, the discretization cannot find the correct segmentation. By using the boundary term, we penalize this leakage since most of its boundaries are hallucinated. The resulting relaxed solution allows for the correct discretization.

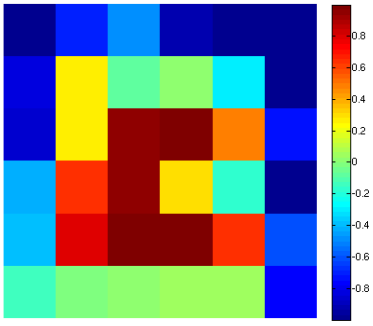
Similarly in Fig. 3.23, we show the benefit of the region grouping perceptual term. Since the contour image is very cluttered, two of the object boundaries are missing and the object is very simple, it is very easy to hallucinate this object anywhere in this image. For example, in the middle of the image one can see the same object with same number of missing contours. If you include region grouping cues, however, the correct object gets segmented better as observed in the relaxed solution. As a result, the discretization obtains the correct segmentation.



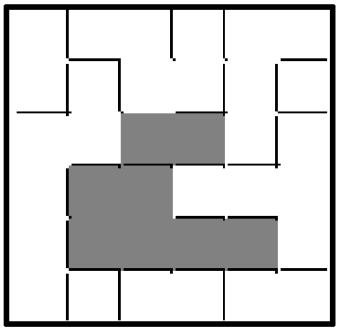
(a) Object.



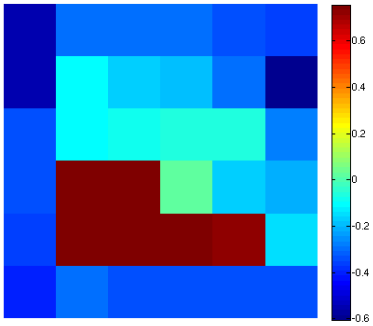
(b) Contour map.



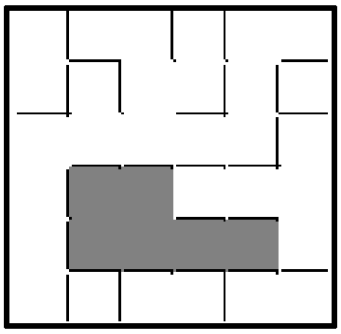
(c) BoSS w/o perceptual terms – relaxed solution



(d) BoSS w/o perceptual terms – discretized solution.

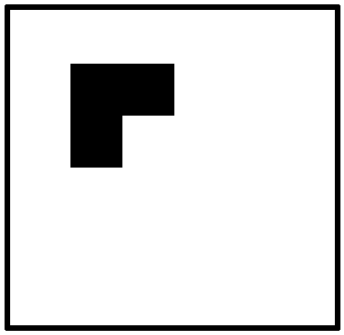


(e) BoSS w/ boundary term – relaxed solution

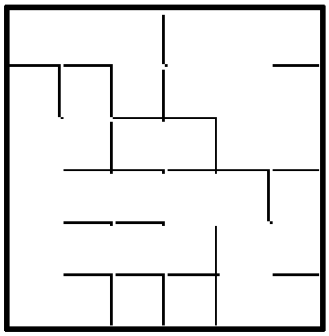


(f) BoSS w/ boundary term – discretized solution.

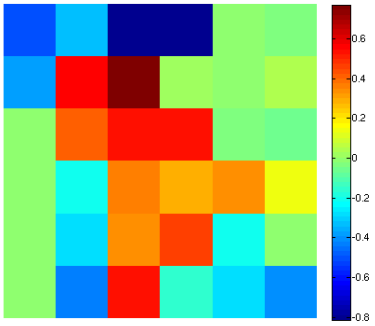
Figure 3.22: For one particular object (a) and a contour image (b) we segment the image using BoSS w/o any perceptual terms (c)-(d) and using BoSS with only boundary term (e)-(f).



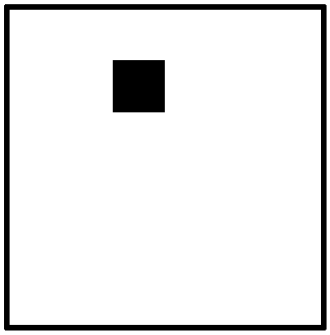
(a) Object.



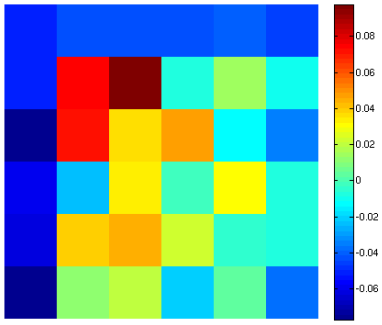
(b) Contour map.



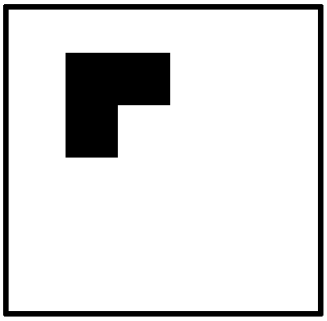
(c) BoSS w/o perceptual terms – relaxed solution



(d) BoSS w/o perceptual terms – discretized solution.



(e) BoSS w/ boundary term – relaxed solution



(f) BoSS w/ boundary term – discretized solution.

Figure 3.23: For one particular object (a) and a contour image (b) we segment the image using BoSS w/o any perceptual terms (c)-(d) and using BoSS with all perceptual terms (e)-(f).

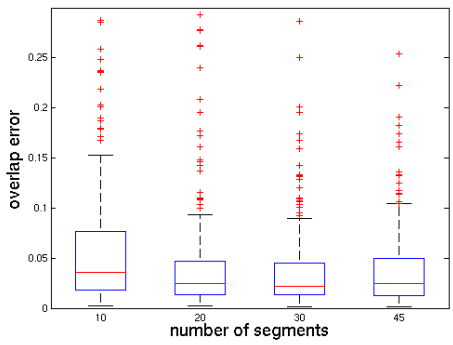
3.8.5 Influence of the Number of Segments

As justified in Sec. 3.8.2, being able to select any possible combination of segments as a figure segmentation is of paramount importance when it comes to shape-based object detection. Using more segments could potentially result in better object segmentation since one should be able to model finer details of an object shape. However, having more segments comes at a higher cost since the optimization problem in Sec. 3.6 will be carried over a larger number of variables.

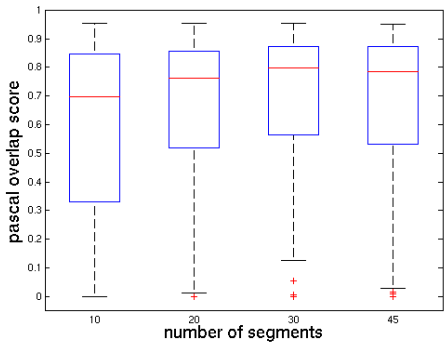
To evaluate the importance of the number of segments in the final object segmentation, we run BoSS with a pre-segmentation on the ETZ Shape Dataset (see Sec. 3.7.1) with 10, 20, 30 and 45 segments obtain using Normalized Cuts [Cour et al., 2005]. For every level of input pre-segmentation, we evaluate the obtained object segmentation using the ground truth model and scale for each image. We use the the overlap error and the pascal overlap score, as introduced in Sec. 3.8.1. To better evaluate the quality of the boundaries of the segmentation, we also compute boundary precision/recall, as used in the evaluation of the segmentation in Sec. 3.7.1.

The results for those four measures over the whole dataset for the four setups are summarized in Fig. 3.24. We can see that the overlap error and the pascal overlap scores improve with increasing number of segments. Moreover, the values become closer to the median, which indicates that with increasing number of segments the quality of the segmentation improves for more images. Similar behavior can be observed for boundary precision/recall. The biggest improvement is in the recall – as we have more segments, we obtain larger portions of the object boundaries better.

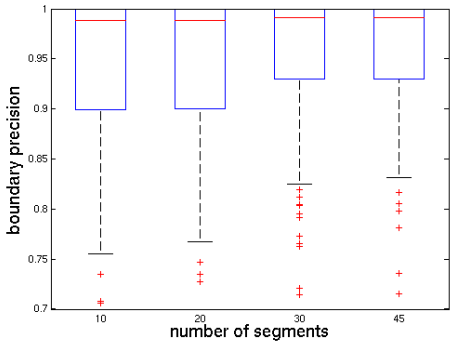
Also, we can see that there is a clear improvement from 10 to 20 and from 20 to 30 segments. However, the observed improvement beyond 30 segments is small.



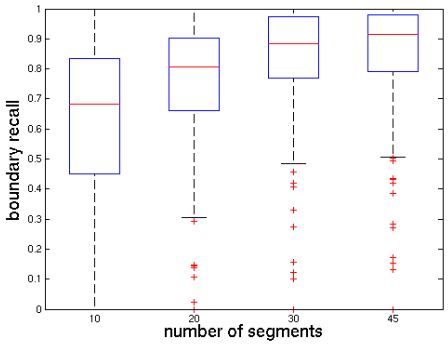
(a) Overlap error.



(b) Pascal overlap error.



(c) Boundary precision.



(d) Boundary recall.

Figure 3.24: We present four different measures for the quality of the segmentation. For each measure, we use all images from the ETHZ Shape Dataset and pre-segmentations with 10, 20, 30, and 45 segments. We display for each measure and pre-segmentation, the median in red, the 25% and 75% quantile as blue boxes, and the range of the values as black lines.

To understand, when one obtains improvement in the segmentation quality via more segments, consider the example in Fig. 3.25. In this case, using 10 segments is not sufficient to capture the mug. Hence, one needs more segments, and when using 45 the object can be segmented. For this reason in around 75% of the images using 45 segments leads to a better segmentation.

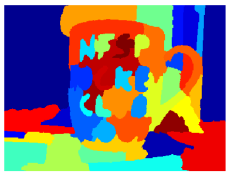
Using 10 segments can be beneficial when the pure bottom-up segmentation can capture the object well. In such situations, using more segments may give unnecessary freedom to the algorithm to make mistakes. For example, in Fig. 3.26, using 45 segments allows the algorithm to lose the cap of the bottle and thus we obtain a worse segmentation.

In general, using more segments is beneficial when the object is either too small or too large. In the former case, using too few segments may lead to large segments none of which captures the object. In the latter case, using too few segments may miss important object boundaries. This analysis can be see by displaying the pascal overlap score versus the object area for segmentation obtained using 10 and 45 segments (see Fig. 3.27).

Although using more segments may lead to better object segmentation, this comes at a cost. The reason is that the optimization from Sec. 3.6 has complexity polynomial in the number of boundary and segment variables. The number of boundaries for the four levels of segmentation is presented in Fig. 3.28 together with the running time on a 3.00 Ghz Intel Xeon Processor using the SeDuMi package [Sturm, 1999] for the optimization.

(a) Object segmentation using 45 segments.

(b) Object segmentation using 10 segments.



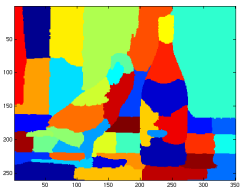
(c) Pre-segmentation with 45 segments.

(d) Pre-segmentation with 10 segments.

Figure 3.25: Using only 10 segments may not capture the object. In this case one should use more segments.

(a) Object segmentation using 45 segments.

(b) Object segmentation using 10 segments.



(c) Pre-segmentation with 45 segments.

(d) Pre-segmentation with 10 segments.

Figure 3.26: Using 10 segments leads sometimes to better results provided the segmentation captures the object of interest.

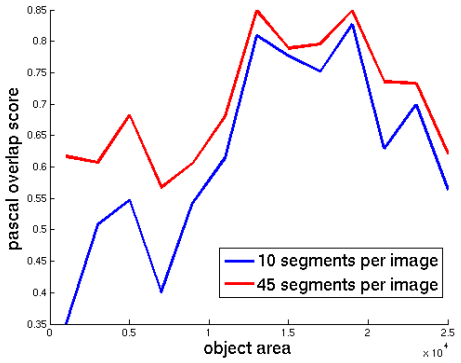
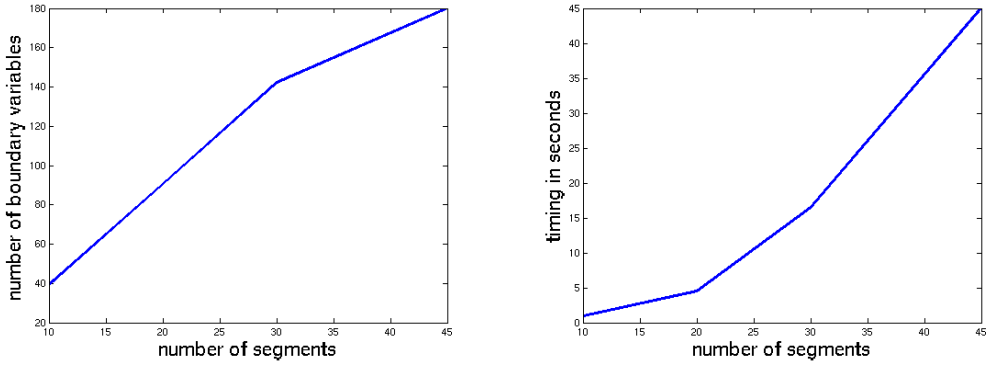


Figure 3.27: Pascal overlap score versus object area. Using too few segments leads to worse segmentation mainly for small and some of the very large objects.



(a) Number of boundary variables. (b) Running time.

Figure 3.28: Empirical computational complexity for different levels of pre-segmentation.

3.8.6 Importance of Representation

In this subsection we analyze the importance of using chordiogram in the BoSS model instead of other representations. Indeed, there are other possible representations which can be incorporated in the BoSS model. More precisely, we replace the chordiogram with Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005]. We will show that the chordiogram-based model performs better for two major reasons:

1. **Translation Invariance:** The HOG is not translation invariant. As a result one needs to run the BoSS model at each possible location and scale. Since each application of the BoSS model requires inference, we need to pick a subset of locations and scales which leads to a loss in precision.

2. **Representational Power:** Although the HOG has proven to be an excellent representation for recognition tasks such as human detection
   [Dalal and Triggs, 2005] and general object recognition
   [Felzenszwalb et al., 2008], it is not so powerful when it comes to capturing shape as defined in this thesis. As a result, we observe a loss in the segmentation quality and this recognition accuracy.

HOG-BoSS Model. In order to replace the chordiogram with HOG in the BoSS model as defined in Eq. 3.16, we need to introduce a new matching cost. For this purpose, similar to Sec. 3.2, we need a parameterization of the HOG in terms of the boundary variables t.

To do this, we define HOG in terms of the notation from Sec. 3.2. Using the binning scheme from [Dalal and Triggs, 2005], the value in the l^{th} bin of the histogram is given by:

$$\text{hog}_l = \#\{p | f_p \in bin(l)\}$$

where $f_p = (p_x - c_x, p_y - c_y, p_e)$ are the features of each boundary point: (p_x, p_y) are

the position of the boundary point p and p_e is its orientation. The position features are taken into account relative to the offset $c = (c_x, c_y)$ of the descriptor.

Suppose that we have a set of boundaries $\mathcal{B}$, as defined in Sec. 3.2. Then, for a given boundary $b^k \in \mathcal{B}$ and a HOG offset c we can define a HOG which captures only this boundary:

$$(\mathrm{hog}_b^k)_l = \#\{p | f_p \in bin(l), p \in b^k\}$$

For the sake of brevity we omit the HOG offset c from the above notation.

Then, for a given parameterization of the image boundaries $\mathcal{B}$ with a boundary indicator vector t, as introduced in Sec. 3.2, an HOG can be expressed as a linear function of the selected image boundaries:

$$\mathrm{hog}(t) = \sum_{b^k \in \mathcal{B}} \mathrm{hog}_b^k t_b^k$$

We can use this parameterization in the shape matching model from Sec. 3.9 and thus in the BoSS model. The resulting model is called HOG-BoSS.

Inference in the HOG-BoSS Model. Since the HOG-BoSS model is applied always with respect to an offset, we need to run it for every location and scale in the image which is not tractable. As a result, we use simple HOG matching in a preprocessing step to obtain a small candidate of potential object locations. The cost of the matching is the L_1 distance between the model HOG and the HOG computed at a particular location by overlaying the model bounding box at this location and using only the boundaries inside this box. We use models over several scales. In a second step, we apply the model for each location and scale as described in Sec. 3.6.

Implementation Details. The grid size of the HOG for each model is defined by finding a rectangular grid with square cells which covers the model and has approximately 32 cells. For each cell, we bin the boundary points according to their normals into 4 bins.

In the preprocessing step, we run HOG matching for each scale and retain the top 5 best matches after non-maximum suppression. We use 7 scales. All 35 hypotheses are passed to the HOG-BoSS model.

Analysis. We run the HOG-BoSS model on the ETHZ Shape Dataset. The results are summarized in Table 3.8 and Fig. 3.29, where we present detection rates for all five classes of the dataset. In particular, we analyze and compare the performance of BoSS with HOG and chordiogram by presenting the following detection rates:

HOG: This is the preprocessing step which generates hypothesis for the HOG-BoSS model.

HOG-BoSS: The BoSS model with HOG representation.

HOG-BoSS with chordiogram reranking: We use the segmentations obtained via HOG-BoSS and rerank them using the chordiogram distance.

HOG-BoSS – maximal possible rate: We present the maximally possible detection rate using all hypotheses generated by HOG-BoSS.

BoSS: Detection rate of the original BoSS model with chordiogram representation.

We can see that pure HOG matching performs poorly and this is result of the clutter and the simple distance function we use. Integrating HOG into the BoSS results in almost 30% boost of the performance. This gain is due to the removal of the clutter. However, HOG performs worse than the chordiogram and this can be seen by reranking the segmentations using the chordiogram distance, which results in 12% improvement.

Even after the reranking, however, the performance of the HOG-based model is almost 30% lower than the original chordiogram-based model. There are two reasons for this. First, the preprocessing step loses hypotheses. This is a result of the fact that HOG-BoSS is not translation invariant which does not allow us to use it

	Applelogos	Bottles	Giraffes
HOG	18.2%	20%	29.7%
HOG-BoSS	63.6%	67.3%	25.3%
HOG-BoSS – chordiogram reranking	68.2%	80%	41.2%
max det. rate	86.4%	94.6%	79.1%
BoSS	88.6%	96.4%	86.8%
	Mugs	Swans	average
HOG	6.1%	12.2%	17.2%
HOG-BoSS	33.3%	42.4%	46.4%
HOG-BoSS – chordiogram reranking	47%	54.6%	58.3%
max det. rate	78.8%	93.9%	86.6%
BoSS	77.3%	93.9%	88.6%

Table 3.8: Detection rates of BoSS with different representations at 0.4 fppi for the five classes of the ETHZ Shape Datatset.

exhaustively over the whole image. Second, the shape expressiveness of HOG is not sufficient enough to segment the objects with the desired precision. As a result, even the maximally possible detection rate is lower that the detection rates we achieve with BoSS at 0.4 fppi.

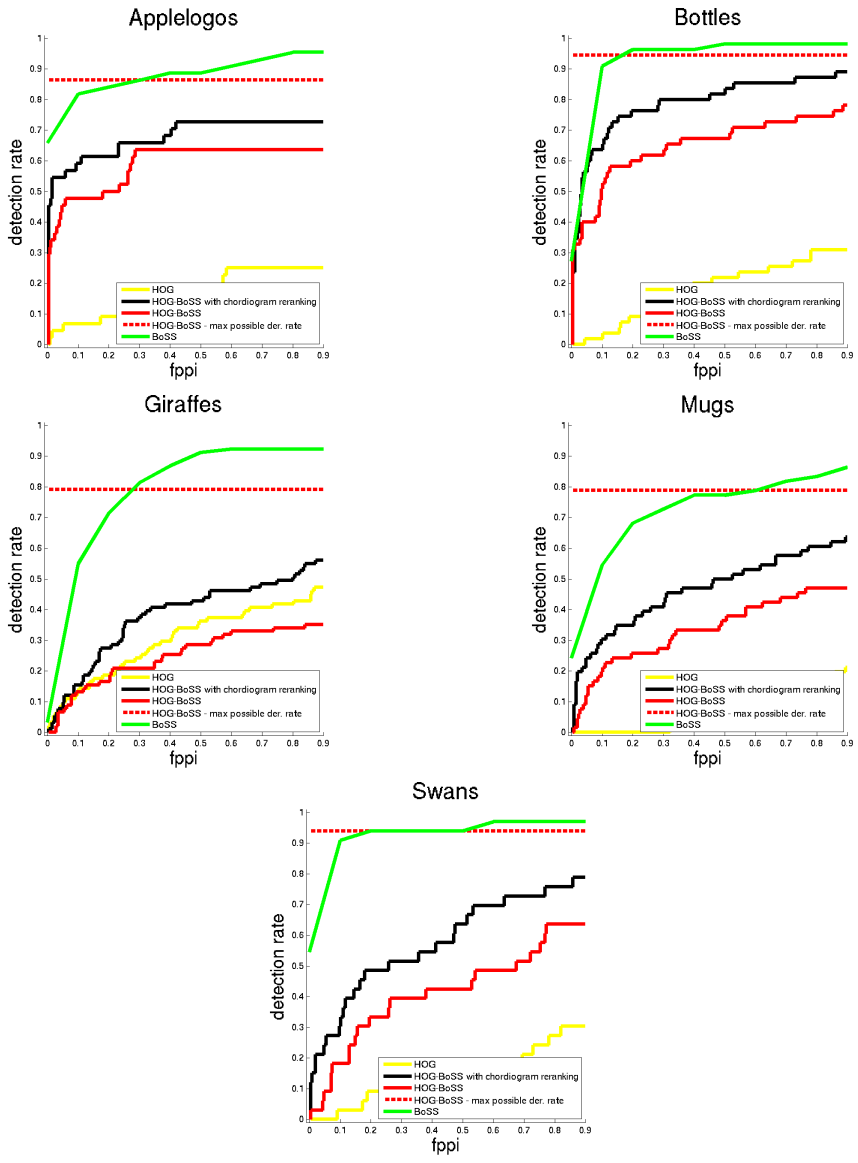


Figure 3.29: Detection rate vs false positives per image for all five classes of the ETHZ Shape Dataset computed HOG-BoSS and comparison to other variations of BoSS (see text for explanation).

3.9 Related Work

Due to the large volume of literature on recognition and segmentation, we review approaches closest to our work. Global shape descriptors, such as Fourier contour descriptors, Zernicke moments, Curvature Scale Space, etc. [Zhang and Lu, 2003] have a long tradition in shape retrieval. However, they are applicable only for already segmented objects and cannot deal robustly with clutter. Semi-local shape descriptors have been proposed to address this limitation. [Belongie et al., 2002] introduce shape context as a histogram of contour edges, capturing parts of an object. To perform recognition with shape context one needs to integrate it in a global matching framework such as thin plate spline or voting, for example. To alleviate further the issues arising from clutter, [Zhu et al., 2008] select relevant object contours while matching shape contexts. Later [Srinivasan et al., 2010] combine the previous work with discriminative learning to leverage salient object contours. Boundary fragments combined with a classifier and subsequent voting for object centers have been explored as well [Opelt et al., 2006, Shotton et al., 2005]. These approaches are part-based and do not use global descriptors. Moreover, all of the above methods recover a set of object contours, but not the figure/ground organization of the image.

A different approach to shape-based recognition is to search for a set of image contours which best matches to a model. [Ferrari et al., 2006] search in a contour network for contour chains which resemble the model. In a subsequent work [Ferrari et al., 2008] define a descriptor for groups of adjacent contour segments and use it in conjuction with an SVM classifier. [Lu et al., 2009] explore particle filtering to search for a set of object contours. [Felzenszwalb and Schwartz, 2007] propose a hierarchical representation by decomposing a contour into a tree of subcontours and using dynamic programming to perform matching. Dynamic programming has been applied also by [Ravishankar et al., 2008] in a mutli-stage framework to search for a chain of object contours. All of the above approaches have to deal with a combinatorial search among image contours and have to decompose their inference into

tractable subproblems, thus losing some of the global relationships between contours. On the contrary, we retain in our descriptor all relations between object boundaries to achieve a holistic representation. Although the above approaches recover some object contours, none of them recover full figure/ground organization.

Close interplay between segmentation and recognition has been studied by [Yu and Shi, 2003] who guide segmentation using part detections, but do not use global shape descriptors. Segment shape descriptors have been used by [Gorelick and Basri, 2009] for detection and segmentation. [Leibe et al., 2008] combine recognition and segmentation in a probabilistic framework. Recently, [Gu et al., 2009] use global shape features on image segments. However, segmentation is a preprocessing step, decoupled from the subsequent matching.

Object dependent segmentation has been addressed in prior work [Borenstein et al., 2004, Levin and Weiss, 2006]. Both methods combine bottom-up segmentation with top-down matching, using templates of object parts as a way to match shape. An explicit reasoning about figure/ground organization has been proposed by [Ren et al., 2005] who use shapemes for local shape matching. Although these approaches have segmentation and boundary priors they employ only local shape descriptors.

3.10 Conclusion

In this chapter we introduced a shape-based object segmentation and detection model, called BoSS. It is based on the shape representation introduced in the previous chapter. In addition, BoSS combines the chordiogram matching with perceptual grouping principles, expressed in terms of region and contours.

The resulting model is formulated as a quadratic integer program, which allows an approximate solution in a single step using off-the-shelf optimization techniques.

The BoSS model operationalizes our global and holistic shape representation,

the chordiogram, and applies it to real scenes, which contain background clutter and multiple objects. In addition, our detection algorithm provides a pixelwise object localization. We present state-of-the-art results on established benchmarks.

Chapter 4

Shape-based Detection in Videos

Many view-point invariant object recognition approaches rely on learning a 2D bag of features or feature constellations from a set of limited views as representation. This has been facilitated in the last decade through the plethora of images on the Internet as well as with the systematic annotation and construction of image benchmarks and corpora [Everingham and et. al, 2005]. In general, representations learnt from images are not useful for leveraging properties of 3D shape for recognition. However, recent advances in range sensor technology, as well as easy to use 3D design tools, have enabled significant collections of 3D models in the form of VRML descriptions[1] or even unorganized point clouds. Use of 3D models makes a recognition system immune to intra-class texture variations and it frees us from the burden to capture as many views as possible. However, it comes with the cost that we cannot make use of the discriminative properties of appearances.

In this chapter we aim at utilizing shape of 3D models and apply it for recognition in videos, which is the type of imagery where having a model of multiple views of an object is of paramount importance [Toshev et al., 2009]. In particular, we target moving objects in videos – as opposed to multiple views – which facilitate foreground-background segmentation as well as a temporal coherency of the object views. We do

[1] http://sketchup.google.com/3dwarehouse/

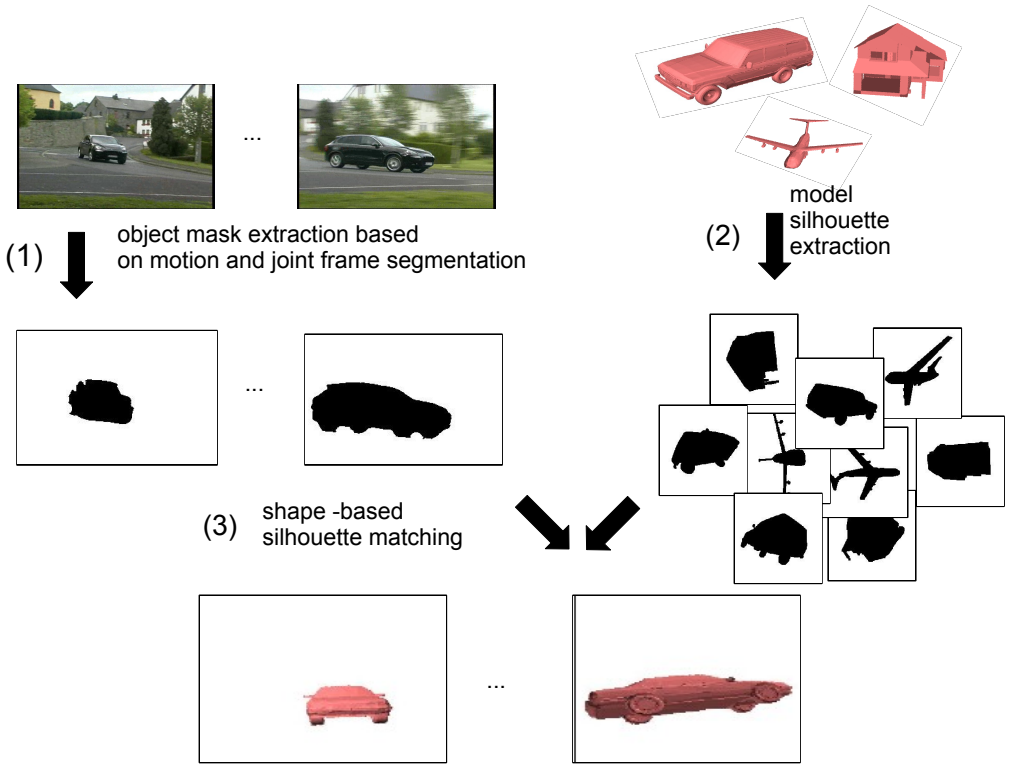


Figure 4.1: Recognition in videos by matching the shapes of object silhouettes obtained using motion segmentation with silhouettes obtained from 3D models.

not claim to propose a recognition system which should replace existing appearance or shape based video search engines [Sivic and Zisserman, 2008]. Instead, we want to show the potential of collections of 3D models to accomplish recognition of moving objects in video.

In a nutshell, we propose to detect objects in videos combining the following three steps (see fig. 4.1 for overview):

1. Extraction of the *silhouette of the moving object* in each video frame (see sec. 4.1). This is accomplished by building upon the video segmentation presented in Chapter 5 and subsequently segmenting the moving object.

2. Extraction of a set of representative model views organized in a *view graph*

(sec. 4.2). The view graph is an intermediate representation of a 3D object which is a compact representation of the set of all silhouettes when captured from a sphere of viewpoints. The use of silhouette projections gives us a stable shape representation that avoids the complexities and variability of a model's internal representation.

3. *Shape-based* matching of the object silhouette with the model silhouette while maintaining motion coherence over time as explained in sec. 4.3.

Our approach provides the following contributions to the state of the art:

1. A unified framework for *detection* of moving objects as well as their *tracking* and *rough pose* estimation in videos. We show that pose estimation is possible even with similar but not exact object models and without use of explicit motion models.

2. The approach is purely *shape-based*, which frees us from the need to model highly variable appearance. Since we use videos as input, we accumulate shape information from several object views, which makes shape information discriminative.

3. Using the proposed method, 3D model datasets, which contain a large number of object classes, can be successfully applied for recognition. This is done in a plug-and-play fashion without the need of manual interaction. Additionally, 3D models give us the ability to match to any model view and thus we do not need to learn recognizers for each object view.

4. The approach relies on good motion segmentation, which in our case is achieved by jointly segmenting the video frames.

4.1 Silhouette Extraction from Video

To recognize moving 3D objects within videos, we need to extract the necessary object shape information from the video sequences. As we described earlier, the shape information we use is a silhouette representation of the moving object in the video. To extract object silhouettes we propose a system that fuses two processes (see fig. (4.2)):

1. Feature tracking and motion-based clustering of the resulting tracks as either object or background (see Sec. 4.1.1).

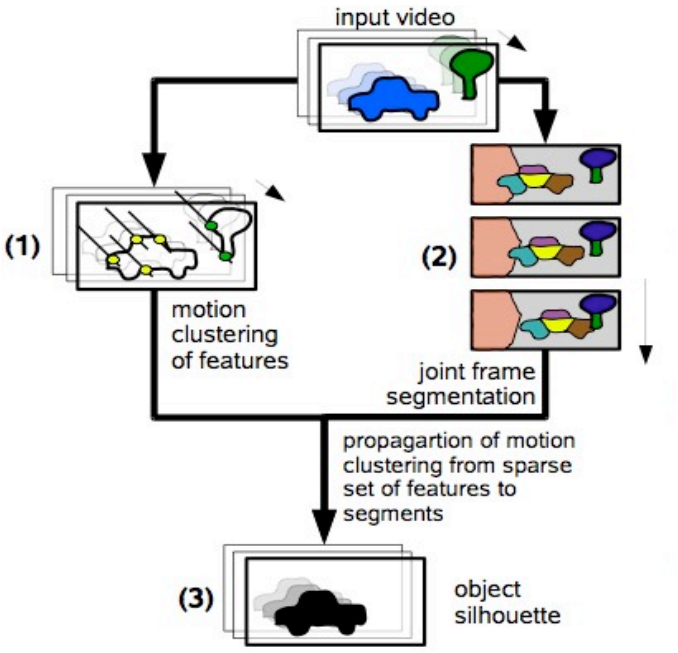
2. Video segmentation into region tracks which represent parts of the scene evolving over time. For this we use the approach presented in Chapter 5, in particular Algorithm 4 from Sec. 5.2.2. This segmentation results in co-salient regions, which we will call also segment tracks.

3. In a subsequent step, we combine the feature track labeling with the segment tracks to obtain masks for the object in each frame (see Sec. 4.1.2).

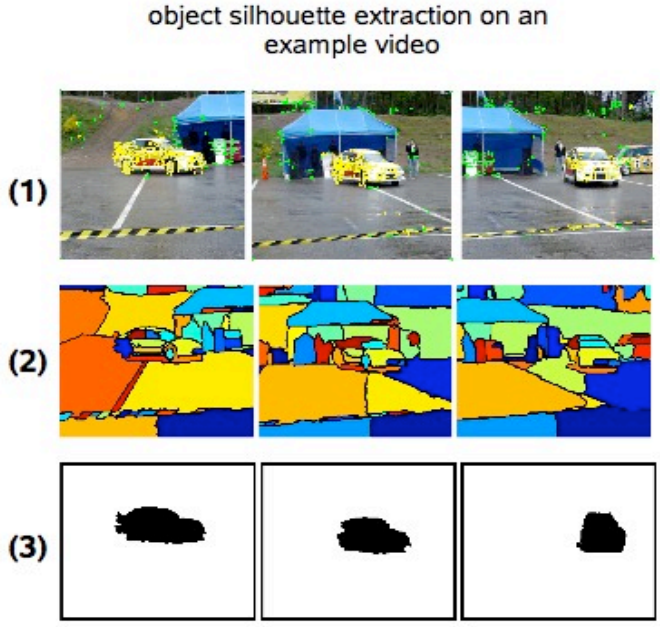
The motivation for this approach is that through feature tracking we can achieve robust sparse motion segmentation, while region tracks will propagate this motion segmentation to the whole image, and thus the object silhouette can be extracted. The approach is similar to [Wills et al., 2003], where the authors use a different segmentation algorithm.

4.1.1 Sparse Figure-ground Labeling

We assume that each video contains at most two different motions (the object and the background; the algorithm can be extended to handle multiple motions [Wills et al., 2003]) and that these two motions are well approximated by an affine motion model, motivated by large distance of the outdoor objects from the camera relative to the object depth variations.



(a) Schematic view of the object silhouette extraction.



(b) Example of the object silhouette extraction.

Figure 4.2: Steps of the object silhouette extraction: (1) feature clustering based on common motion; (2) segment tracks; (3) object silhouette.

More precisely, we seek to compute two motions $M_l = \{A_l^{(2)}, \ldots, A_l^{(T)}\}$, $l \in \{object, background\}$. Here $A_l^{(t)}$ is the affine motion which transforms features labeled as l from their locations in frame $t-1$ to their locations in frame t, and T is the total number of frames in the video. As we are assuming affine motion, we extract and track features using the KLT tracker [Shi and Tomasi, 1994]. The output is a collection of tracks $\bar{x}_1, \ldots, \bar{x}_n$. Each track has a start frame $start(\bar{x}_i)$, an end frame $end(\bar{x}_i)$, and a sequence of image locations $\bar{x}_i = \{x_i^{(t)} | start(\bar{x}_i) \leq t \leq end(\bar{x}_i)\}$. We enforce the feature labels to be consistent over the entire track and denote this by l_i for track $\bar{x}_i$, with $L = \{l_1 \ldots l_n\}$ being the set of all track labels.

Our goal is to recover the two motions $M = \{M_{\text{obj}}, M_{\text{bckg}}\}$ as well as an assignment from the feature tracks to the motions. This can be achieved by minimizing the fitting error of all tracks to their motion models:

$$E_{\text{motion}}(M, L) = E(M_{\text{obj}}, L) + E(M_{\text{bckg}}, L) \tag{4.1}$$

where $E(M_l, L)$ is the fitting error of a particular motion model M_l defined as the combined fitting error of all tracks with label l w. r. t. the affine motions in M_l:

$$E(M_l, L) = \sum_{i=1}^{n} \sum_{t=start(\bar{x}_i)+1}^{end(\bar{x}_i)} \delta(l, l_i) \epsilon(A_l^{(t)}, x_i^{(t)}) \tag{4.2}$$

where $\epsilon(A_l^{(t)}, x_i^{(t)}) = \|A_l^{(t)} x_i^{(t-1)} - x_i^{(t)}\|_2^2$ is the fitting error of a feature and $\delta(l, l_i) = 1$ if $l = l_i$ and 0 otherwise.

The energy in eq. (4.1) can be minimized by employing the EM algorithm [Dempster et al., 1977]. In the E-step we assign a label l to a feature track by identifying the motion M_l with the smallest fitting error. In the M-step, we update the affine motions of M_l using the tracked features with label l. Instead of estimating the affine transformations from all the feature tracks with label l (in closed form by minimizing the least-squares error), we use RANSAC [Fischler and Bolles, 1981] to stay robust to possible outlier tracks. After the final iteration, we discard all tracks that are counted as outliers from the final RANSAC estimation. After estimating two

motion models we assign label *object* to the one whose features have larger scatter in the image defined as the variance of the feature locations.

4.1.2 Object Silhouette Detection

Now we show how to use the track labels and the segments to obtain object silhouettes for the foreground object in each frame. The main idea is to propagate the motion labeling of the tracks to labels of the segments, while maintaing spatial and temporal smoothness of the segment labeling. We model the interplay between tracks and segments using an MRF over the segments $S = \{s_1 \dots s_K\}$ from all frames, $s_i \in \{object, background\}$ denoting the label of the i^{th} segment. The energy function

$$E_{\text{silhouette}}(S) = \sum_{i=1}^{K} E_{\text{propg}}(s_i) + \sum_{i,j=1}^{K} E_{\text{smooth}}(s_i, s_j) \tag{4.3}$$

consists of unitary energy term E_{propg} which propagates motion labels of features to segments, and a term E_{smooth} assuring that the resulting labels do not violate spatial and temporal smoothness. A detailed definition of both terms follows below.

Propagating from feature tracks to segments Although the segment tracks provide good segmentation, the segment boundaries are not precise enough to measure directly their compatibility with the estimated affine motion models. Therefore, we propose to label the segments by propagating labels from features to the segments. A simple way could be through assigning features to segments based on their locations. However, many of the tracked features tend to lie close to the boundary of a segment which makes feature-to-segment assignment ambiguous (see feature A in fig. (4.3)). To resolve this problem, we propose to use the Delaunay triangulation of the features. We define a motion energy term for each of the resulting triangles $\{d_1, \dots, d_m\}$ (we denote by d_i the i^{th} triangle as well as its label) based on the average fitting error of its vertices: $E_{\text{tri}}(d_i = l) = \frac{1}{3} \sum_{x \in d_i} \epsilon(A_l, x)$ where ϵ is the fitting error as defined in eq. (4.2) applied for the motion model A_l of the frame of d_i. The

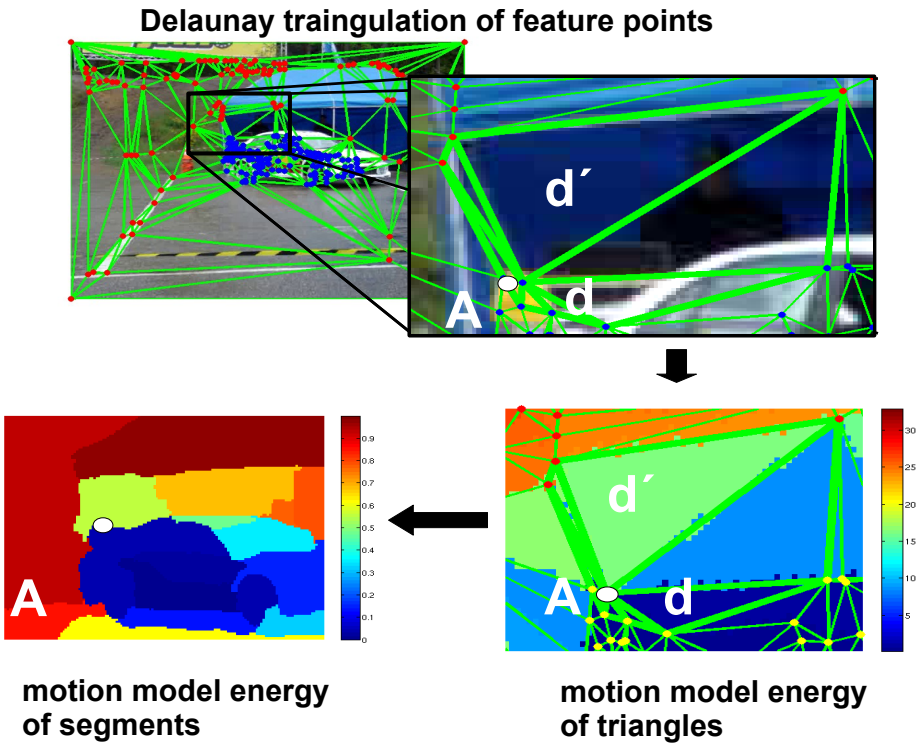


Figure 4.3: Upper left: all features with their motion (blue denotes object, red - background) and their Delaunay triangulation; upper right - a zoom in of feature A and its triangles; lower right: motion model energy of each triangle (dark blue means object); lower left: propagation of the triangle energy to the segments. (For further explanation see text.)

motivation for this definition is that we can assign a motion label robustly to the triangles since a triangle accumulates information from several vertices. For example, in fig. (4.3) the boundary point A, moving with the object, is the only vertex labeled as *object* on the background triangle d', while it correctly supports triangle d to be labeled as object. We propagate the motion label energy of the triangles to entire segments as the average of the triangle energies weighted by the area overlap o_{ij} between the i^{th} segment and j^{th} triangle:

$$E_{\text{seg}}(s_i) = \frac{1}{O_i} \sum_{j=1}^{m} o_{ij} E_{\text{tri}}(d_j = s_i) \tag{4.4}$$

where $O_i = \sum_{j=1}^{m} o_{i,j}$.

Spatial and temporal smoothness The smoothness constraint is imposed among neighboring segments using a Potts smoothness term and is based on the normalized color histogram h_i of the segments:

$$E_{\text{smooth}}(s_i, s_j) = \alpha + (1 - \alpha) \exp\left(-\frac{\|h_i - h_j\|^2}{2\sigma_c^2}\right)$$

if the j^{th} segment is in the neighborhood of the i^{th} one and $s_i \neq s_j$, 0 otherwise. The neighborhood of a segment i is defined as the set of segments which are either in the same frame and share a common boundary, or are in the consecutive frame and share a common boundary with a segment from the segment track of segment i. In this way we incorporate spatial as well as temporal relationships. In our experiments, we use 6-dim. RGB color histogram and set $\alpha = 0.2$, $\sigma_c = 0.3$, and $b = 6$.

Finally, the labeling of the segments S is obtained by minimizing the energy in eq. (4.3). This is a submodular binary labeling problem, hence it can be solved exactly using Graph Cuts [Kolmogorov and Zabih, 2002].

4.2 Model View Graph

The matching of video frames to 3D model silhouettes requires a compact yet complete model representation. We propose to use a small set of model silhouettes, called *view graph*[2] (see fig. 4.4). Each silhouette is a compact representation of a subset of all possible model views, while the whole graph covers the entire viewing sphere. The edges connect neighboring silhouettes represent view transitions which can be induced by ego-motion of the model.

To create such a view graph, we orthographically render a large number (500 in our experiments) of silhouettes from approximately uniformly distributed viewing

[2]We deviate from [Cyr and Kimia, 2004] and do not use the term *aspect graph* because we do not follow its mathematical definition.

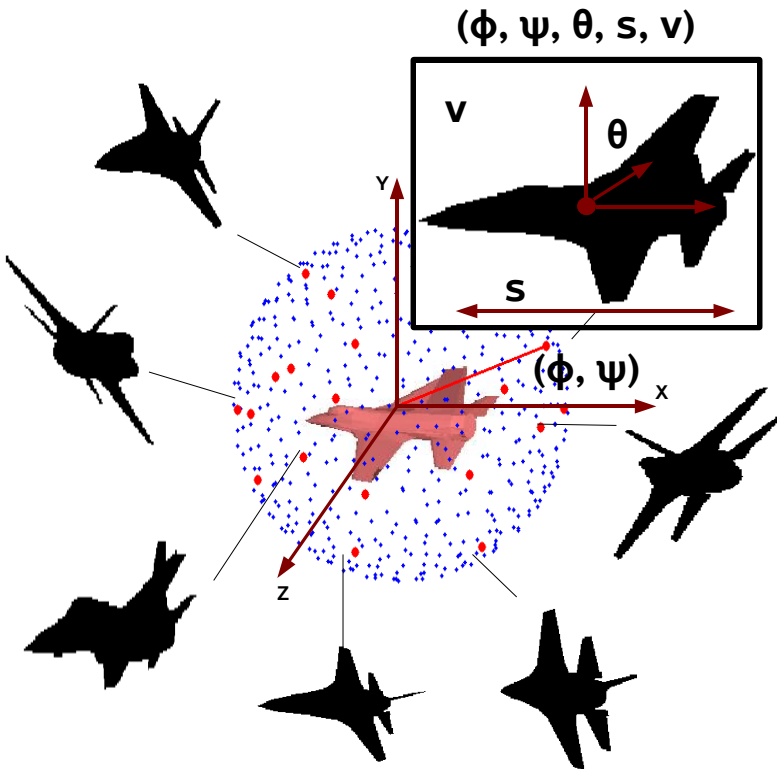


Figure 4.4: View graph: 500 viewing points (blue) extracted initially and the view graph (red) after clustering. Some of the silhouettes are displayed. In addition, we show the parameterization of one of the views.

angles and cluster them into a few representative views. However, for 260 models, which is the dataset size we use, this results in 130000 silhouettes, which not only contain redundant information, but also pose a computational challenge for matching. To obtain the graph, we perform k-medoids clustering of the 500 silhouettes with 20 modes [Cyr and Kimia, 2004].

The clustering for view graph creation requires a feature vector for each silhouette representing its shape. We compute shape contexts [Belongie et al., 2002] centered at silhouette boundary points (we uniformly sample 20% of the boundary). Since a shape context is not rotation invariant, each silhouette is pre-rotated to a canonical orientation, given by the rotation that brings the offset vector q (which connects

the offset of the shape context to the centroid of the silhouette) to the X-axis. Hence, the descriptor $\mathrm{sd}_k = (\mathrm{sc}_k, q_k)$ assigned to boundary point k contains both the shape context sc_k and the offset vector q_k. We use the extracted shape contexts from all silhouettes to compute a codebook of size 200, using k-means clustering. Using nearest neighbor binning we can build a histogram over codewords for each silhouette image. The resulting 200-dimensional histogram vector is used for the view clustering.

4.3 Matching of Object Silhouette Sequences to Models

After we have extracted a sequence of silhouettes from the video and a set of model silhouettes organized in a view graph, we cast the matching problem as matching of the object silhouettes with the view graph. A simple approach would be to let each frame vote for a particular model and thus to detect the object class as the class of the model with most votes.

In addition, one can try to incorporate motion information and require that the matched model views represent a smooth transition in the view graph. This second approach can be considered as an alignment of the video silhouettes with a view graph of a model (see fig. 4.5). The benefit of the second approach is two-fold – the best aligned model gives the class of the observed object in the video, while the path on the view graph gives a rough pose estimate of the object motion.

In the following exposition, we will denote the video silhouette sequence by $F = \{f_1 \dots f_T\}$ and a model view graph by $G = \{m_1 \dots m_K\}$ of K views. Each model view $m_i = (v_i, \theta_i, s_i, p_i)$ is parametrized by the its silhouette v_i, its orientation θ_i around the silhouette centroid, its scale s_i, and the viewpoint $p_i = p_i(\varphi_i, \psi_i)^T$ on the model view sphere from which it was extracted (see Fig. 4.4).

4.3.1 Shape Matching

We consider two shape matching techniques: shape context [Belongie et al., 2002] and the rotation invariant chordiogram from Sec. 2.2.

Shape Context. We sample the silhouette boundary and compute shape context descriptor (SC) [Belongie et al., 2002] at each sampled point. Since SC is not rotation invariant, before extracting it at boundary point q we rotate the silhouette to a canonical orientation for q such that the offset vector q (the vector connecting the offset with the object center) is aligned with the X axis. The resulting SC sc is combined together with the offset vector q in a descriptor $\mathrm{sd} = (\mathrm{sc}, q)$.

We denote a correspondence between boundary point $\mathrm{sd}_k = (\mathrm{sc}_k, q_k)$ of the object silhouette and a boundary point $\mathrm{sd}_n = (\mathrm{sc}_n, q_n)$ of the model silhouette by $c_{k,n}$. The L_2 distance between the shape contexts sc_k and sc_n is a measure of the semi-local shape similarity of the silhouettes and can be used to define a probability for observing model silhouette shape v and the point correspondence $c_{k,n}$:

$$P(c_{k,n}, v|f_i) = \exp(-||\mathrm{sc}_k - \mathrm{sc}_n||^2/2\sigma_{\mathrm{sc}}^2) \tag{4.5}$$

Besides local similarity, we also evaluate the global shape similarity by measuring how well individual point matches agree with each other. For this we need a parameterization of the alignment of the model with the object. Since we know the centroids of the silhouettes, the alignment can be parameterized by a similarity transformation with a zero translational component. It is defined by a rotation $R(\theta) \in SO(2)$ around the mask centroid and a scale s. Hence this is the transformation which aligns the offset vectors q_k and q_n of the shape contexts. The probability of this alignment given the correspondence $c_{k,n}$ is:

$$P(\theta, s|c_{k,n}, v, f_i) = \exp(-||sR(\theta)q_k - q_n||^2/2\sigma_{\mathrm{sp}}^2) \tag{4.6}$$

By combining all shape context similarities and the global alignment the probability

of a matching model silhouette $m = (v, \theta, s)$ can be written as:

$$P(m|f_i) = \frac{1}{A} \sum_{k,n} P(\theta, s|c_{k,n}, v, f_i) P(c_{k,n}, v|f_i) \tag{4.7}$$

where A is a normalization factor guaranteeing that the above quantity is a probability distribution.

In our implementation we used the following parameter values: $\sigma_{\text{sc}} = 0.25$ in eq. (4.5), $\sigma_{\text{sp}} = 4$ in eq. (4.6).

Chordiogram. For each silhouette v we compute the rotation invariant chordiogram $\text{ch}(v)$ (see Sec. 2.2).

In this application of the chordiogram, we use the following parameter values: $b_l = 5$, $b_d = 5$, $b_n = 8$. The maximal bin boundary of the chords' length bins is equal to the length of silhouette major axis, while the maximal bin boundary of distance to center bins is set to be half of the silhouette major axis length. These parameters ensure that the chordiogram covers the whole silhouette.

4.3.2 Frame Voting for a Model

Suppose that for each frame f_i and each class label l we have a score $w(l, f_i)$ for this frame containing an object of class l. Then we can aggregate those scores as votes over the whole video and determine the class of the video as the one with highest cumulative vote:

$$\mathit{label}(F) = \arg\max_l \sum_{i=1}^{T} w(l, f_i) \tag{4.8}$$

Shape Context. To compute a score in the case of the SC matching, we use the shape probability from eq. (4.7) to compute a score of the best alignment between each video frame and each model view:

$$s(v, f_i) = \max_{\theta, s} P(\theta, s, v|f_i)$$

The maximum is computed using Hough transform – each match $c_{k,n}$ casts a vote for a rotation and scale and we pick the ones with largest vote accumulation. Having already computed a matching score we can use it to define a voting score

$$w_{\text{sc}}(l, f_i) = \sum_v \delta(label(v), l) s(v, f_i) \tag{4.9}$$

where we sum over all model silhouettes in the model dataset which have label l. This score can be used in the voting scheme from Eq. (4.8).

Chordiogram. For each model view v and a frame f_i we can compute the L_1 distance between their corresponding chordiograms and interpret this as a shape distance:

$$s(v, f_i) = ||\text{ch}(v) - \text{ch}(f_i)||_1 \tag{4.10}$$

Then we can define a binary voting score such that a frame f_i votes for a class l exactly if there is a model of this class which is among the k closest models to the frame.

$$w_{\text{ch}}(l, f_i) = \begin{cases} 1 & \text{if there is a model } v \text{ among the closest } k \text{ models with label}(v) = l \\ 0 & \text{otherwise} \end{cases}$$

4.3.3 Alignment of the Video to a Model View Graph

To define an alignment score for a given video $F = \{f_1 \dots f_T\}$ and a model view graph $V = \{m_1 \dots m_T\}$, where m_i are views from the view graph, we use a conditional random field [Lafferty et al., 2003]. It defines a joint distribution over the aligned model views V (see fig. 4.5 as well):

$$P(V|F) = \frac{1}{Z(F)} \prod_i^T P(m_i|F) P(m_i, m_{i-1}|F) \tag{4.11}$$

where $Z(F)$ is the partition function. Estimating the model silhouette sequence V for which the above distribution is maximized amounts to finding the alignment

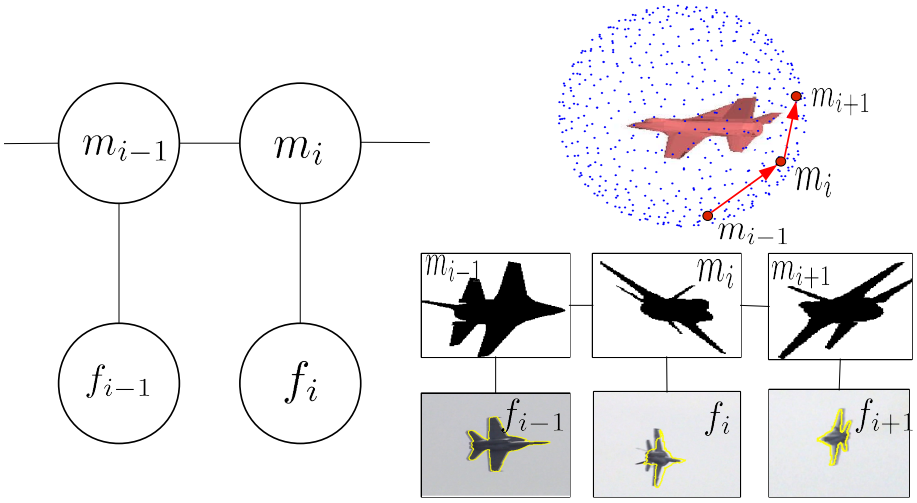


Figure 4.5: Left: CRF model of the video-to-model alignment. Right: alignment shown for 3 frames of a video and a model view graph.

with the highest score. This inference can be solved exactly by backwards-forwards algorithm for CRFs [Lafferty et al., 2003]. Below we define the different terms. To define the unitary potentials in the CRF in Eq. (4.11), we use eq. (4.7).

Transition smoothness in the view graph. The second term $P(m_i, m_{i-1}|F)$ represents the transition of the model silhouette at time $i-1$ to the one at time i. We require smooth transitions – the viewing points p as well as the alignment defined as rotation θ to corresponding frames (see previous section) should be similar:

$$P(m_i, m_{i-1}|F) = \exp\left(-\frac{(\theta_i - \theta_{i-1})^2}{2\sigma_r^2} - \frac{\text{acos}(p_i^T p_{i-1})^2}{2\sigma_p^2}\right)$$

Voting for initial object detection. To perform full recognition we need to solve the problem in eq. (4.11) for each model. Since this can be computationally challenging, in a first step we use shape information from each video frame independently to detect a few matching models for the whole video – each individual frame votes for the best matching model class and we retain the class with largest votes. We use the voting scheme described in Sec. 4.3.2.The full model from eq. (4.11) is

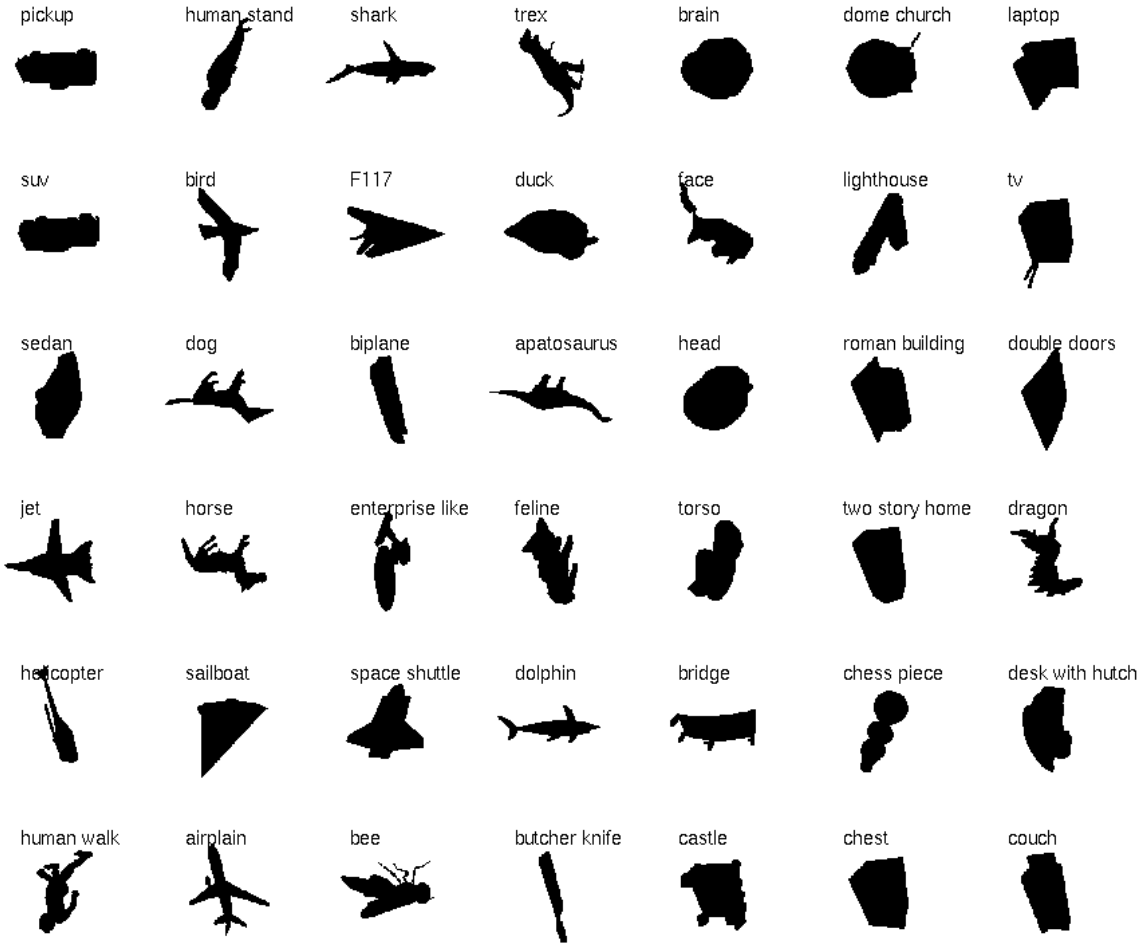


Figure 4.6: A sample of the 43 classes we use from the Princeton Shape Benchmark [Shilane et al., 2004].

solved in a second step only for models from the best matching class.

We used the following parameter values: $\sigma_p = 60°$, $\sigma_r = 20°$. We rescale model silhouettes to have variance 70 pixels.

4.4 Experiments

Videos. We perform experimental evaluation of our approach using 42 videos representing 3 different classes: *car* (15 videos), *airplane* (12 videos), and *helicopter* (15

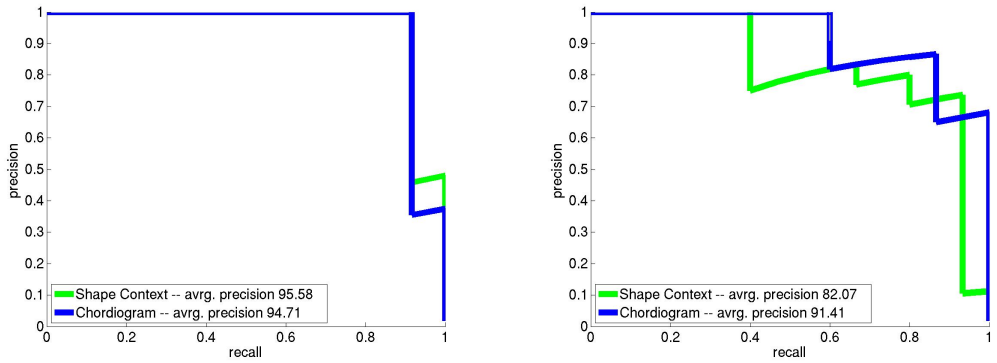
videos). These videos are between 41 and 1159 frames long, in total 10029 frames, and were collected manually or from the web. During the video segmentation process we do not use every frame, but depending on the length of the video we use every third or every tenth frame. This is motivated by computational limitations. As a result, the number of frames which were used in the recognition stage is 2954, ranging from 21 to 116 frames per video.

3D Models. We obtain 3D models from the Princeton Shape Benchmark [Shilane et al., 2004], which contains a variety of different objects. We use 43 classes, 5 models per class (see Fig. 4.6). The classes represent a rich variety of object types – vehicles, animals, furniture, architectural elements, etc. In particular, the dataset contains classes of type *car* and *helicopter* and 5 classes of type *airplane* (*passenger plane*, *jet*, *F117*, *biplane*, and *space_shuttle*). Note that we initially extract 500 views per model, resulting in a total 107500 silhouettes for the 215 models we use, many of which contain unrealistic views, e. g. bottom of a sailboat, which make the matching problem even harder. Moreover, although we test on 3 video classes, we use all 43 model classes in order to test the robustness of our system to shape variation. The goal is to utilize the whole shape dataset without any manual selection.

Recognition and alignment results. We match the input videos to the models and determine a model class using the voting procedure of sec. 4.3. In Table. 4.1 we present the percentage of the correctly classified videos. We achieve accuracy of 88.3% over all videos using the chordiogram and Eq. (4.10), while the score from Eq. (4.9) based on shape context result in 83.9% detection rate.

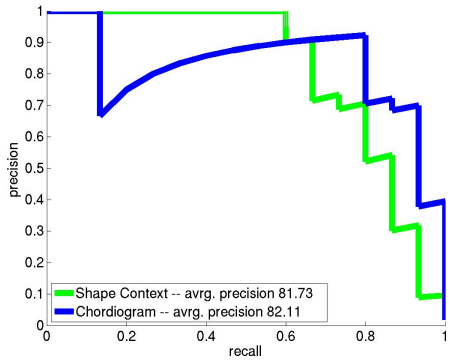
In addition, we present the precision–recall curves for the two voting schemes in Fig. 4.7. The average precision for the chordiogram-based method is 89.4%, while for shape context it is 86.5%.

The results show that we can robustly detect the correct object class using a textureless dataset of models without being confused by the large variety of object



(a) airplain

(b) helicopter



(c) car

Figure 4.7: Precision–recall curves using the voting scheme and scores based on shape context and the chordiogram. In addtion, we present the average precision for each class.

types. The few mistakes are result of shape ambiguities. For example, in fig. 4.8 we can see incorrect matches due to very similar model object outlines. However, due to object motion we always see one or several discriminative object shapes, which decrease the effect of ambiguous shapes.

Further, we apply the the alignment procedure of Eq. (4.11). We can achieve good alignment of the model with the video, and thus estimate the rough pose of the moving object (see Fig. 4.9–4.9).

class	detection rate	
	Shape Context	Chordiogram
car	80.0%	86.7%
airplane	91.7%	91.7%
helicopter	80.0%	86.7%
average	83.9%	88.3%

Table 4.1: We show the detection rates for the voting scheme and scores based on shape context and the chordiogram.

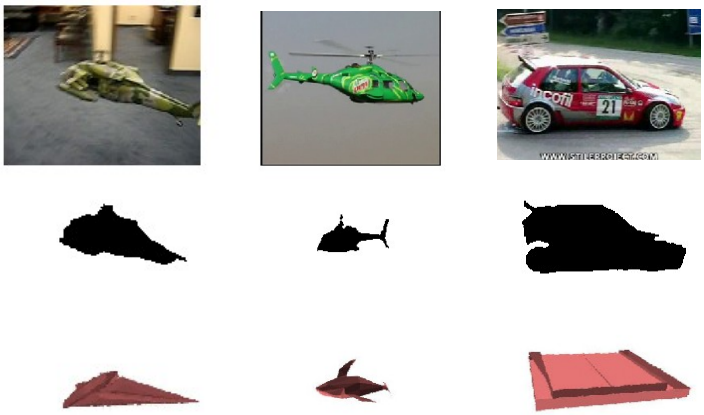


Figure 4.8: Failure cases for the matching (first row – frame, second row – object mask, third row - best model match).

4.5 Related work

Much of the early work in 3D model recognition was performed by matching wire-frame representations of simple 3D polyhedral objects to detected edges in an image with no background clutter and no missing parts (a nice summary can be found in Grimson's book [Grimson, 1990]). An exception was aspect graphs, which first appeared in [Koenderink and Doorn, 1979] and emerged again as a new representation of 3D objects (see for example [Kriegman and Ponce, 1990]). Aspect graphs in their strict mathematical definition (each node sees the same set of singularities) were not considered practical enough for recognition tasks. However, the notion of sampling in the view-space for the purpose of recognition was introduced again in

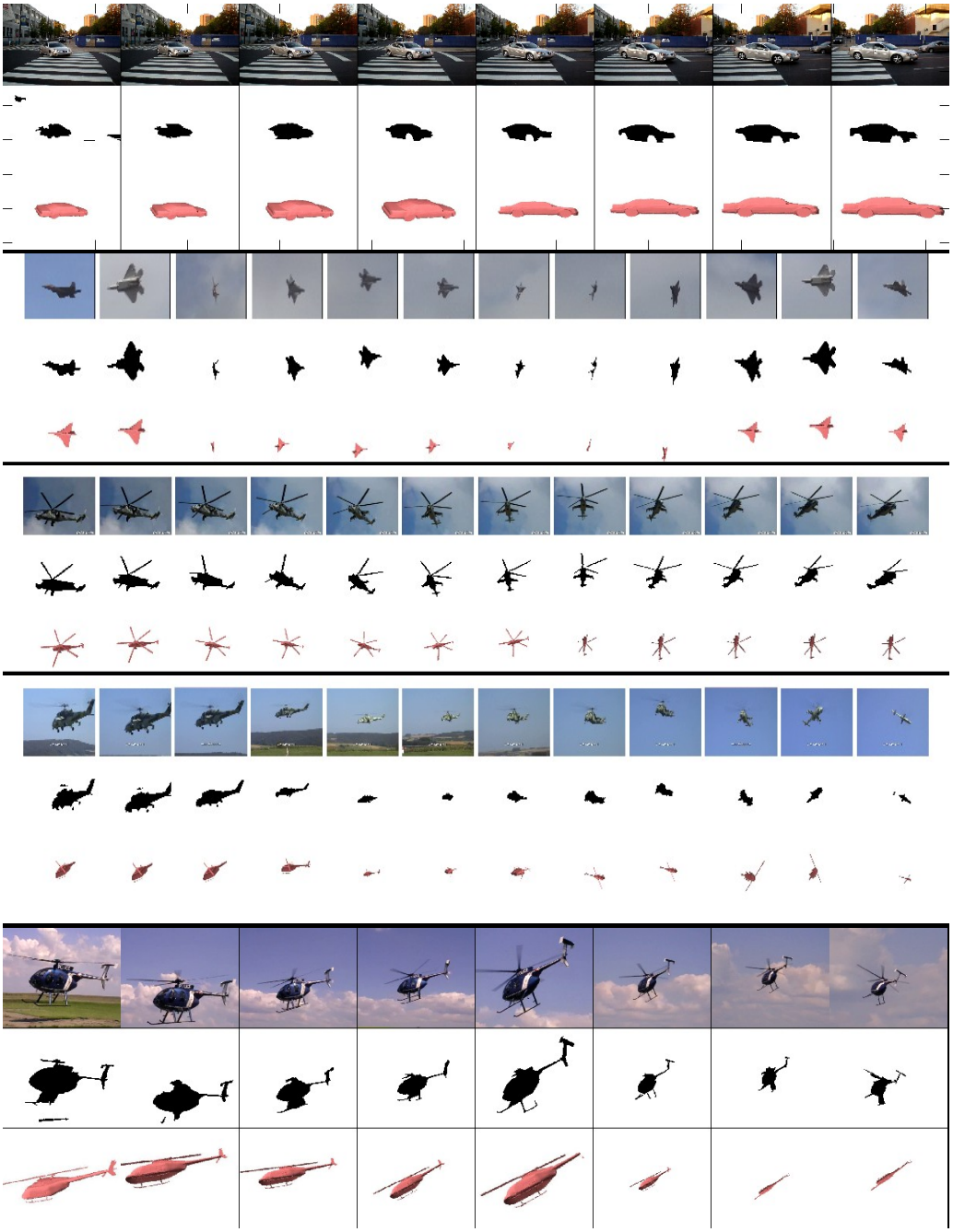


Figure 4.9: Matching results and model alignment for 5 videos. For each example, we show in three rows the input video, the detected silhouette sequence, and the aligned matched model (we sample 8 equally spaced in time frames from each of the displayed videos).

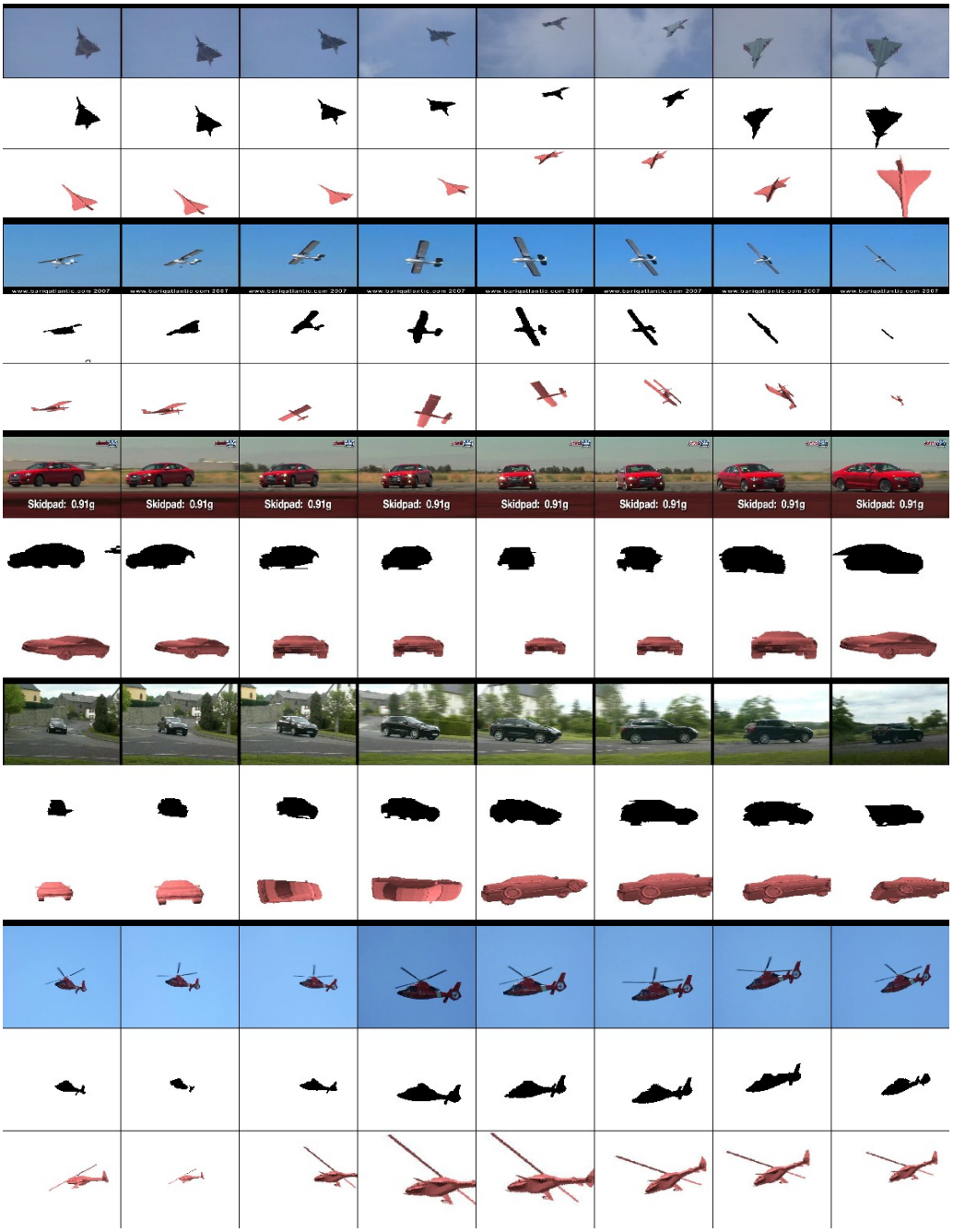


Figure 4.10: Matching results and model alignment for 5 videos. For each example, we show in three rows the input video, the detected silhouette sequence, and the aligned matched model (we sample 8 equally spaced in time frames from each of the displayed videos).

[Cyr and Kimia, 2004], and is the closest to our use of an aspect graph but is applied for matching synthetic 3D models.

Regarding recognition from still images, Ullman introduced the representation of 3D objects based on view exemplars [Ullman and Basri, 1991] and several recent approaches use a sample of appearance views deliberately taken to build a model [Rothganger et al., 2003, Ferrari et al., 2004, Savarese and Fei-Fei, 2007]. The authors of [Savarese and Fei-Fei, 2007] propose to learn object category models encoding shape and appearance from multiple images of the same object category by relating homographies between the same plane in multiple views. [Rothganger et al., 2003] extract 3D object representations based on local affine-invariant descriptors of their images and the spatial relationships between surface patches. To match them to test images, they apply appearance feature correspondences and a RANSAC procedure for selecting the inliers subject to the geometric constraints between candidate matching surface patches. In terms of models, Liebelt's approach [Liebelt et al., 2008] is very close to ours since it works with a view space of rendered views of 3D models. Appearance features are selected based on their discriminativity regarding aspect as well as object category and they are matched to single images in the standard benchmark datasets.

[Sivic et al., 2006] use matching based on affine co-variant regions across multiple video frames to create models of all seen parts of 3D objects. It is the closest approach to ours regarding the nature of the data while we use an aspect representation which is the closest to [Cyr and Kimia, 2004].

4.6 Conclusion

In this chapter we have addressed the problem of object recognition in videos. The proposed approach is based on the ideas and methods presented earlier in this thesis.

First, we explore shape as a major representation for recognition in videos.

We use and compare the chordiogram, presented in Chapter 2, and shape context [Belongie et al., 2002] for the purpose of recognition in videos. Using shape allows us to utilize large datasets of 3D models, which are freely available on the web. Our experimental evaluation shows that shape is discriminative among many classes when we deal with videos.

Second, we show that co-salient perceptual grouping as described in Chapter 5 combined with motion information can be successfully applied to segment the moving object. This is sufficient in many frames to obtain an object mask precise enough for shape matching. Note that this is a different approach than the one advocated in Chapter 3, where the model-based shape matching was tightly integrated with the perceptual grouping – while in a single image such tight coupling is of paramount important, in the case of videos we can exploit motion information.

Chapter 5

Co-salient Perceptual Grouping and Matching

Perceptual region grouping has been widely used as a preprocessing step in recognition [Mori, 2005, Russell et al., 2006]. Although unsupervised segmentation provides meaningful groups, it is inherently unstable – segments rarely capture the same parts in different images of the same object or scene. For example, in Fig. 5.1, upper image pair, segmenting two images of the same scene with the same algorithm results in different sets of regions. In the previous chapters we have addressed this problem by precomputing an oversegmentation of an image and piecing an object out of this set of precomputed segments.

Correspondence estimation is one of the fundamental challenges in computer vision lying at the core of many problems, from stereo and motion analysis to object recognition. The predominant paradigm in such cases has been the correspondence of object parts or scene components. Such parts are usually represented using local features [Lowe, 2004, Matas, 2004, Mikolajczyk and Schmid, 2004, Dalal and Triggs, 2005] such as interest points, whose power is in the ability to robustly capture discriminative image structures in a repeatable manner – interest point detectors tend to fire at similar structures in different images. Feature-based

approaches, however, suffer from the ambiguity of local feature descriptors and therefore are often augmented with global models which are in many cases domain dependent (see upper image pair in Fig. 5.1).

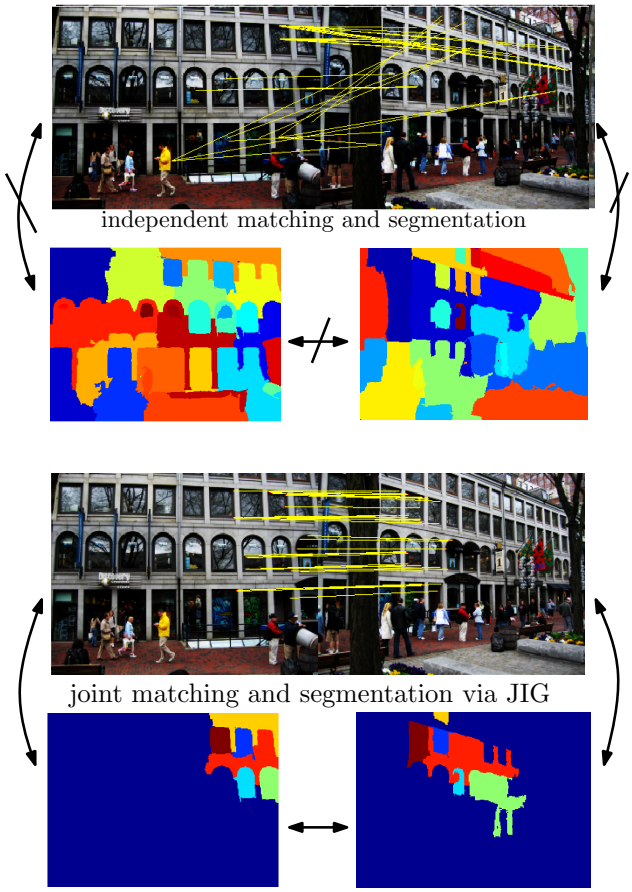


Figure 5.1: Independently computed correspondences and segments (upper diagram) for a pair of images can be made consistent with each other via the joint image graph and thus improved (lower diagram).

In this chapter we address the above problems by combining the complementary strengths of local features and segmentation. We attempt to resolve the matching ambiguities related to local features by providing grouping constraints via segmentation. In addition, the feature correspondences can be used to synchronize both segmentations and obtain consistent segments. In this way we attempt to obtain

image segmentation such that the segments can be put in correspondence across images.

We introduce a perceptual framework to matching and segmentation by modeling in one score function both the coherence of regions within images as well as similarities of features across images. We will refer to such a pair of corresponding regions as *co-salient* and define them as follows:

1. Each region in the pair should exhibit strong internal coherence with respect to the background in the image;

2. The correspondence between the regions from the two images should be supported by high similarity of features extracted from these regions (see fig. 5.1).

To formalize the above model we introduce the *joint-image graph* (JIG) which contains as vertices the pixels of both images and has edges representing intra-image similarities and inter-image feature matches. The matching problem is cast as a spectral segmentation problem in the JIG. A good cluster in the JIG consists of a pair of coherent segments describing corresponding scene parts from the two images. The eigenvectors of the JIG weight matrix represent 'soft' joint segmentation modes and capture the co-salient regions.

The resulting score function can be optimized with respect to both the joint segmentation and feature correspondences. In fact we employ a two step iteration with optimization of the joint segmentation eigenvectors in the first step. In the second step we improve the feature correspondences by identifying those correspondences which support the region matches indicated by the joint eigenvectors and suppressing the ones which disagree with it. Furthermore, we can use the co-salient regions to induce new feature correspondences by extracting additional features not used by the initial estimation and checking their compatibility with the region matches.

In the next section we proceed with the introduction of the model. The solution to the problem is presented in sec. 5.2 and sec. 5.3. We conclude with experimental

results in sec. 5.4.

5.1 Joint-Image Graph (JIG) Matching Model

The JIG is a representation of N images, which incorporates both intra- and inter-image information. It is constructed as a weighted graph $G = (I_1 \cup \ldots \cup I_N, E, W)$, whose vertex set consists of the pixels of all images I_i, $i \in \{1, \ldots, N\}$. Denote the number of pixels in I_i by n_i. The weights W of the edges represent similarities between pixels:

$$W = \begin{pmatrix} W_1 & C_{1,2} & 0 & \ldots \\ C_{1,2}^T & W_2 & C_{2,3} & \ldots \\ \vdots & \vdots & \ddots & \end{pmatrix} \tag{5.1}$$

$W_i \in [0,1]^{n_i \times n_i}$ is a weight matrix of the edges connecting vertices in I_i with entries measuring how well pixels group together in a single image, called also an *affinity matrix*. The other component $C_{i,j} \in [0,1]^{n_i \times n_j}$ is a *correspondence matrix*, which contains weights of the edges connecting vertices from I_i and I_j, i. e. the similarities between local features across the two images. We can assume that $C_{i,j}^T = C_{j,i}$. Further, denote all the correspondence matrices by $\mathcal{C} = \{C_{i,j} | i, j \in \{1, \ldots, N\}, i \neq j\}$. To emphasize that W contains correspondences, we will occasionally write W as $W[\mathcal{C}]$.

In order to combine the robustness of matching via local features with the descriptive power of salient segments we detect clusters in JIG. Each such cluster S represents a set of regions $S = S_1 \cup \ldots \cup S_N$, $S_i \subseteq I_i$, $i \in \{1, \ldots, N\}$, and contains pixels from some of the N images. We can describe each region S_i with an *indicator vector* $v_i \in \{0,1\}^{n_i}$: $(v_i)_x = 1$ iff pixel x lies in the region S_i; all indicator vectors for S can be stacked in a vector $v = (v_1^Y \ldots v_N^T)^T \in \{0,1\}^n$ for $n = \sum_{i=1}^N n_i$.

Using the above notation we can introduce the idea of co-salient regions. A set S is called co-salient if each pair S_i, S_j, with $S_i, S_j \neq \emptyset$,

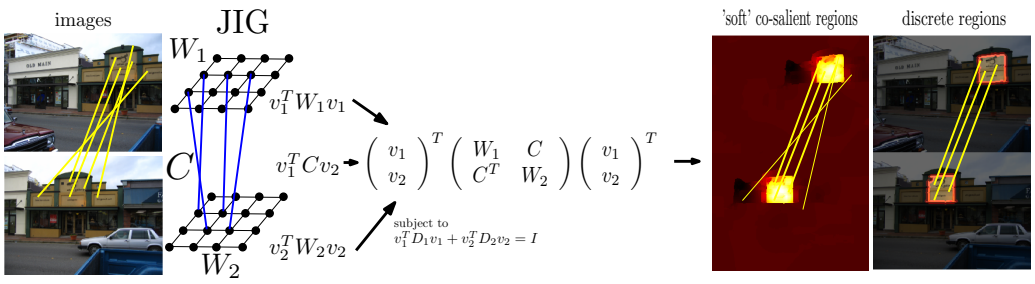


Figure 5.2: Diagram of the matching score function exemplified on two images. The final score function consists of the sum of two components from eq. (5.4) and eq. (5.6). The joint optimization results in 'soft' eigenvectors, which can be further discretized, and a correct set of feature matches. This example can be extended to more than two images in a straightforward manner.

Intra-image similarity criterion: Form coherent and perceptually salient regions in the i^{th} and j^{th} image respectively.

inter-image similarity criterion: Match well according to the feature descriptors.

We formalize the two criteria as follows (see also fig. 5.2):

Intra-image similarity

The image segmentation score is the Normalized Cut criterion applied to all co-salient regions

$$\text{IntraIS}(S) = \frac{\sum_{i\in\{1,\ldots,N\}} \sum_{x\in S_i, y\in S_i} (W_i)_{x,y}}{N(S)} \tag{5.2}$$

with normalization

$$N(S) = \sum_{i\in\{1,\ldots,N\}} \sum_{x\in S_i, y\in I_i} (W_i)_{x,y} \tag{5.3}$$

Using the notation of indicator vectors, the criterion can be written as

$$\text{IntraIS}(v) = \frac{\sum_{i\in\{1,\ldots,N\}} v_i^T W_i v_i}{v^T D v} \tag{5.4}$$

where $D_i = \text{diag}(W_i \mathbf{1}_{n_i})$ is the *degree matrix* of W_i; $\mathbf{1}_{n_i}$ is an n_i dimensional vector with all elements equal to one. Both the indicator vectors and degree matrices for all images can be written succinctly by stacking them in a single vector v and a matrix D:

$$D = \begin{pmatrix} D_1 & 0 & \dots \\ 0 & D_2 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \qquad v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \end{pmatrix}$$

Inter-image similarity

The matching score can be expressed as

$$\text{InterIS}(S) = \frac{\sum_{i,j \in \{1,\dots,N\}, i \neq j} \sum_{x \in S_i, y \in S_j} (C_{i,j})_{x,y}}{N(S)} \tag{5.5}$$

with the normalization defined in Eq. (5.3). This function measures the strength of the connections between the regions S_i and S_j.

The normalization favors correspondences between pixels which are weakly connected with their neighboring pixels – exactly at places where the above segmentation criterion is uncertain in one of the images. We do not use in the normalization any correspondences across images. The reason is that if we include the correspondence scores in the normalization then a keypoint, which is similar to several keypoints in another image, will decrease the above score. However, these ambiguous matches are indication of repetitive local structure and should not penalize the matching.

If we use the same indicator vector as above, then it can be shown that

$$\text{InterIS}(v, \mathcal{C}) = \frac{\sum_{i,j \in \{1,\dots,N\}, i \neq j} v_i^T C_{i,j} v_j}{v^T D v} \tag{5.6}$$

The correspondence matrix $C_{i,j}$ is defined in terms of feature correspondences encoded in a $n_i \times n_j$ matrix M (detailed definition of M is given in section 5.4). As $C_{i,j}$ we select a subset of all feature correspondences such that no feature is mapped

to more than one other feature. This can be encoded by a matrix P:

$$C_{i,j} = P_{i,j} \circ M_{i,j} \quad \text{for all pairs} \quad i,j \in \{1,\ldots,N\}, i \neq j \tag{5.7}$$
$$\text{with} \quad P_{i,j}\mathbf{1}_{n_j} \leq \mathbf{1}_{n_i}, \quad \mathbf{1}_{n_j}^T P_{i,j} \leq \mathbf{1}_{n_i} \quad P_{i,j} \in \{0,1\}^{n_i \times n_j}$$

where $\circ$ is the elementwise matrix multiplication.

Co-saliency score function

Because we want to match co-salient regions, we should maximize the sum of the scores in eq. (5.4) and eq. (5.6) simultaneously. In the case of k pairs of co-salient regions we can introduce k indicator vectors packed in $n \times k$ matrix $V = (v^{(1)}, \ldots, v^{(k)})$, $n = \sum_i^N n_i$. Then we need to maximize

$$\begin{aligned} \text{CoSaliency}(V, \mathcal{C}) &= \sum_{c=1}^{k} \text{IntraIS}(v^{(c)}) + \text{InterIS}(v^{(c)}, \mathcal{C}) \\ &= \sum_{c=1}^{k} \frac{(v^{(c)})^T W v^{(c)}}{(v^{(c)})^T D v^{(c)}} \end{aligned}$$

The score IntraIS is related closely to the Normalized Cuts image segmentation function [Yu and Shi, 2003] – its maximization amounts to obtaining 'soft' segmentation, represented by the eigenvectors of W with large eigenvalues. In our case, however, the estimation of v_i is related via the score function InterIS. Therefore, this process synchronizes the segmentations of all images and retrieves matches of segments, which are supported by the feature matches.

Co-saliency Matching

The CoSaliency score from Eq. (5.8) can be used in two ways to match a set of images.

Co-saliency region matching. The first application would be to estimate a set of co-salient regions, which is also known as the problem of co-segmentation

[Rother et al., 2006]. More precisely, one would estimate segments V across images for a given set of correspondences C.

$$(CSRM): \quad \max_{V} \quad \text{CoSaliency}(V) = \sum_{c=1}^{k} \frac{(v^{(c)})^T W v^{(c)}}{(v^{(c)})^T D v^{(c)}} \tag{5.8}$$

$$\text{subject to} \quad V\mathbf{1}_k = \mathbf{1}_n, \quad V \in \{0,1\}^{n \times k} \tag{5.9}$$

The last two constraints (5.9) enforce V to be a valid indicator vector – the elements of V are integral and each pixel is assigned to exactly one region. Since we do not estimate any feature correspondence, the above problem can be also understood as a synchronization of the segmentation of the images.

Co-saliency region and feature matching. Besides synchronizing the segmentation of two or more images, one may want to improve feature matches across the images. This leads us to the second more general application of the CoSaliency score in which we attempt not only to co-segment the images but also to detect feature matches consistent with the co-salient region matches:

$$(CSRFM): \quad \max_{V,\mathcal{C}} \quad \text{CoSaliency}(V,\mathcal{C}) = \sum_{c=1}^{k} \frac{(v^{(c)})^T W[\mathcal{C}] v^{(c)}}{(v^{(c)})^T D v^{(c)}} \tag{5.10}$$

$$\begin{aligned} \text{subject to} \quad & V\mathbf{1}_k = \mathbf{1}_n, \quad V \in \{0,1\}^{n \times k} \\ & C_{i,j} = P_{i,j} \circ M_{i,j} \\ & P_{i,j}\mathbf{1}_{n_j} \leq \mathbf{1}_{n_i}, \quad \mathbf{1}_{n_j}^T P_{i,j} \leq \mathbf{1}_{n_i} \quad P_{i,j} \in \{0,1\}^{n_i \times n_j} \end{aligned}$$

In addition to $(CSRM)$, we select correspondences supported by the region matches by optimizing over P as well. The last two constraints are take directly from the parameterization of the correspondence matrices in Eq. (5.7).

5.2 Optimization in the JIG

Both problems ($CSRM$) in Eq. (5.8) and ($CSRFM$) Eq. (5.10) are in general non-convex integer quadratic programs and are NP-hard. Therefore, we seek an optimization procedure which gives an approximate solution. We will use the techniques presented in [Yu and Shi, 2003].

Problem ($CSRM$) in Eq. (5.8) can be considered as a subproblem of ($CSRFM$) from Eq. (5.10). Therefore, we first show how to approximately optimize ($CSRM$). This step amounts to synchronization of the 'soft' segmentations of two images based on C as shown in the next section. Then we use ($CSRM$) to optimize ($CSRFM$). This involves an additional step, in which we find an optimal correspondence matrix C given the joint segmentation V.

5.2.1 Co-saliency Region Matching

For fixed C, the optimization problem from eq. (5.8) can be relaxed to by dropping the constraints (see [Yu and Shi, 2003]):

$$\max_{Z} \quad \operatorname{tr}\left(Z^T W Z\right) \tag{5.11}$$

$$\text{subject to} \quad Z^T D Z = \mathbf{1}_k \tag{5.12}$$

The above program is formulated in terms of the *scaled indicator matrix* $Z = V(V^T DV)^{-1/2}$. After one has estimated an approximate scaled indicator matrix, the corresponding original indicator can be recovered via

$$V = \operatorname{Diag}(\operatorname{diag}^{-1/2}(ZZ^T))Z \tag{5.13}$$

The above problem is maximization of the Rayleigh quotient can be solved by computing the eigenvectors corresponding to the largest k eigenvalues of the generalized eigenvalue problem (W, D) [Golub and Loan, 1989].

Although one could apply this procedure directly to ($CSRM$), there are two obstacles. First, C may contain many erroneous matches which will lead to noise

in the matrix W. And since we attempt to discover clusters in JIG, this noise may be detrimental. Second, the bottleneck of the above relaxation is the computation of the eigenvectors of W, whose size is $Nn \times Nn$ for N images each containing n_i pixels each, $n = \sum_i n_i$. Using Power Iteration, the top k eigenvectors can be computed in $O(N^2n^2k)$ [Golub and Loan, 1989]. If we exploit the sparsity of W, then its top eigenvectors can be computed in $O(N^{3/2}n^{3/2}k)$ using the Lanczos method [Golub and Loan, 1989]. This is an intensive computation which is challenging even for a single image and therefore does not scale up for more than one image ($N > 1$).

As a remedy to the above problems we assume that the joint 'soft' segmentation V of all images lies in the subspace spanned by the the 'soft' segmentations of the individual images. In the following, we will present the optimization in the case of two images ($N = 2$), although it can be generalized to many images in a straightforward manner. Suppose that $S_i \in$ are the top s eigenvectors of the corresponding generalized eigenvalue problems for all images $W_iS_i = D_iS_i\Lambda_i$. Then, the notion that V should lie in the subspace of the 'soft' segmentations of the individual images translates to the constraint

$$V = SV_{\text{sub}}, \quad \text{where} \quad S = \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix}$$

is the joint image segmentation subspace basis and V_{sub} are the coordinates of the joint 'soft' segmentation in this subspace.

With this subspace restriction for V the relaxed program from Eq. (5.11) can be written as

$$\max_{V_{\text{sub}}} \quad \text{CoSaliency}_{\text{sub}}(V_{\text{sub}}) = \text{tr}\left(V_{\text{sub}}^T S^T W S V_{\text{sub}}\right) \tag{5.14}$$

$$\text{subject to} \quad V_{\text{sub}}^T V_{\text{sub}} = I_k \tag{5.15}$$

The matrix S^TWS is the original JIG weight matrix restricted to the segmentation subspaces. If we write $V_{\text{sub}} = \begin{pmatrix} V_1^{(s)} \\ V_2^{(s)} \end{pmatrix}$ in terms of the subspace basis coordinates

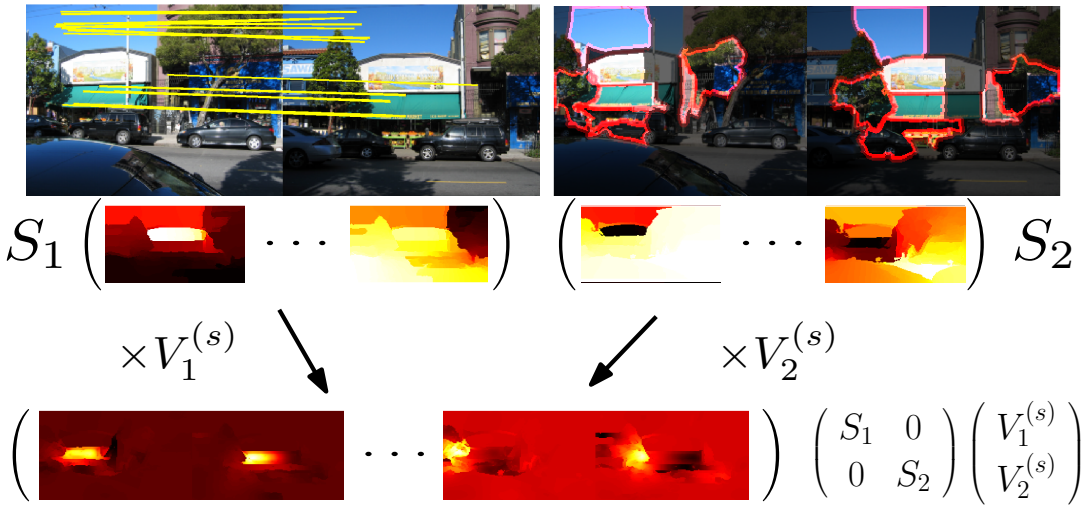


Figure 5.3: Image view of segmentation synchronization. Top left: an image pair with outlined matches. Below: the image segmentation subspaces S_1 and S_2 (each eigenvector is reshaped and displayed as an image) can be linearly combined to obtain clear corresponding regions (awning, front wall), which can be discretized, as displayed in the upper right corner of this figure.

$V_1^{(s)}$ and $V_2^{(s)}$ for both images, then the above score function can be decomposed as follows:

$$\begin{aligned} \text{tr}\left(V_{\text{sub}}^T S^T W S V_{\text{sub}}\right) &= \text{tr}\left((V_1^{(s)})^T \Lambda_1 V_1^{(s)} + (V_2^{(s)})^T \Lambda_2 V_2^{(s)}\right) \\ &\quad +2\text{tr}\left((V_1^{(s)})^T S_1^T C S_2 V_2^{(s)}\right) \end{aligned} \tag{5.16}$$

The second term is a correlation between the segmentations of both images weighted by the correspondences in C and, thus, it measures the quality of the match. The first term serves as a regularizer, which emphasizes eigenvectors in the subspaces with larger eigenvalues and, therefore, describing clearer segments.

The optimal V_{sub} in eq. (5.14) for k co-salient regions is attained for the top k eigenvectors of $S^T W S V_{\text{sub}} = V_{\text{sub}} \Lambda_s$, corresponding to the largest eigenvalues written as a diagonal matrix Λ_s. Note that $S^T W S$ is a $Ns \times Ns$ matrix, for $s \leq 100$ and N images, while the eigenvalue problem necessary to solve program eq. (5.11) has much higher dimension n^2. Therefore, the subspace restriction speeds up the

problem and makes it tractable for a set of images. The resulting SV_{sub} represents a linear combination of the original 'soft' segmentation such that matching regions are enhanced. The initial and synchronized segmentation spaces for an image pair are shown in Fig. 5.3.

Alignment in the embedding space. A different view of the above process can be obtained by representing the eigenvectors by their rows: denote by b_x^i the x^{th} row of $S_i V_i^{(s)}$. Then we can assign to each pixel x in the i^{th} image a k-dimensional vector b_x^i which we will call the embedding vector of this pixel. This vector is obtain by projecting the s-dimensional rows of S_i into a k-dimensional space ($k < s$) via $(V_i^{(s)})^T$, where for each image we compute a different projection. Then the segmentation synchronization can be viewed as an alignment of the segmentation embeddings of both images such that corresponding pixels are close in the resulting lower dimensional embedding (see Fig. 5.4).

Discretization of co-salient regions. From the synchronized segmentation eigenvectors we can extract regions. Suppose $b_x^i = (b_{x,1}^i \dots b_{x,k}^i)^T \in \mathbb{R}^k$ is the embedding vector of a particular pixel x in the i^{th} image (also the x^{th} row of V_i). Then, we label this pixel with the eigenvector, for which the corresponding element in the embedding vector has its highest value:

$$\text{label}(x) = \arg\max_l \{b_{x,l}^i | l \in \{1, \dots, k\}\} \tag{5.17}$$

A more principled way to obtain a discrete solution is presented in [Yu and Shi, 2003]. It is based on the fact the value of the score $\text{CoSaliency}_{\text{sub}}$ from Eq. (5.14) is invariant with respect to a rotation:

$$\begin{aligned} \text{CoSaliency}_{\text{sub}}(V_{\text{sub}} R) &= \text{tr}\left(R^T V_{\text{sub}}^T S^T W S V_{\text{sub}} R\right) \\ = \text{tr}\left(V_{\text{sub}}^T S^T W S V_{\text{sub}}\right) &= \text{CoSaliency}_{\text{sub}}(V_{\text{sub}}) \quad \text{for any } R \in O(k) \end{aligned}$$

This inspires the authors to apply the discretization from Eq. (5.17) to a relaxed indicator $SV_{\text{sub}}R$ which after applying a rotation R is closest to an integral solution.

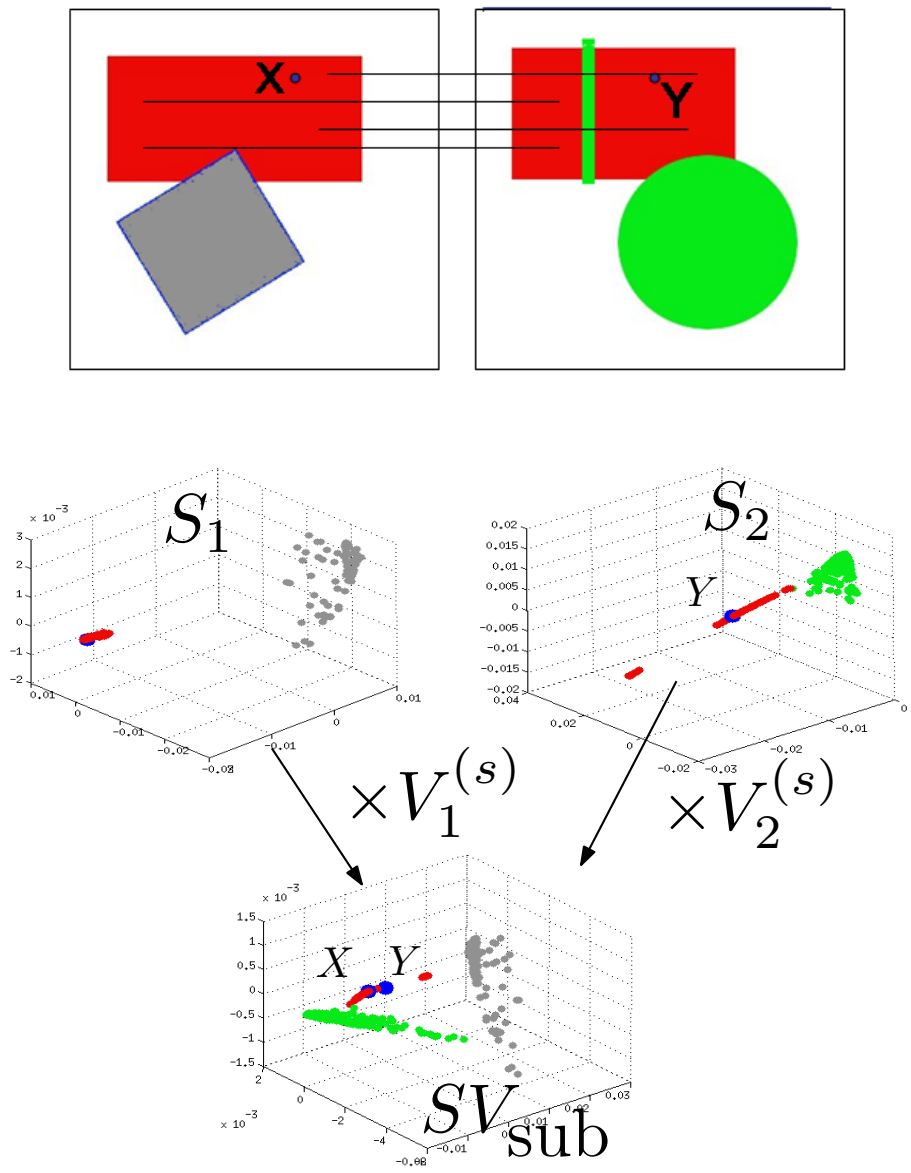


Figure 5.4: Subspace view of the segmentation synchronization. Below each of the images in the first row, the embedding of the pixels of the image in the segmentation space spanned by the top 3 eigenvectors is displayed. The pixels coming from different objects in the image are encoded with the same color. In the third row, both embeddings transformed by the optimal V_{sub} (eq. (5.16)) are presented, given the matches selected as shown in the first row. Both embeddings were synchronized such that all pixels from both rectangles form a well grouped cluster (the red points). In this way the matches were correctly extended over the whole object, even in presence of an occlusion (green vertical line in right image).

More precisely, one solves the program (see Theorem 1 from [Yu and Shi, 2003]):

$$\begin{aligned} \min_{V,R} \quad & \text{Discret}(V,R) = ||V - \tilde{S}V_{\text{sub}}R||_F \\ \text{subject to} \quad & V \in \{0,1\}^{n\times k}, \quad V\mathbf{1}_k = \mathbf{1}_n \\ & R \in O(k) \end{aligned} \tag{5.18}$$

for $\tilde{S} = \text{Diag}(\text{diag}^{-1/2}(SS^T))S$ (see program in Eq. (5.11)).

The resulting indicator vector mask $\widehat{V} = [\widehat{v}_1 \cdots \widehat{v}_k]$ describes k segments, where the l^{th} column is the indicator for the l^{th} co-salient region. Note that $\widehat{v}_l$ describes a segment in the JIG and therefore represents a set of corresponding regions in the images. The matching score between segments can be defined as CoSaliency($\widehat{v}_l$).

The final algorithm for co-saliency region matching is outlined in Algorithm 3.

Algorithm 3 Co-saliency region matching.

1: Initialize W_i and $C_{i,j}$ as in section 5.1. Compute W.
2: Compute segmentation subspaces S_i as the eigenvectors to the s largest eigenvalues of W_i.
3: Find optimal segmentation subspace alignment by computing the top k eigenvectors of S^TWSV_{sub}: $S^TWSV_{\text{sub}} = V_{\text{sub}}\Lambda_s$, where Λ_s are the eigenvalues.
4: Discretize using program 5.18.

Discussion. The computation of the segmentation subspaces can be done in $O(n_i^{3/2}s)$ for the i^{th} image. The complexity of the third step is the computation of the top k eigenvectors of a dense matrix of size $Ns \times Ns$, which can be accomplished in $O((Ns)^2k)$. The discretization has running time $O(\sum_i n_i k^2)$ [Yu and Shi, 2003]. The total complexity is $O(\sum_i n_i^{3/2}k + (Ns)^2k + \sum_i n_i k^2)$.

Clearly the dominant term in the above complexity is the first one. Notice that in case we need to match an image several times, the segmentation subspaces can be precomputed once. Then, each co-salient region matching will take $O((Ns)^2k + \sum_i n_i k^2)$.

5.2.2 Co-saliency Region Matching during Discretization

The segmentation synchronization procedure outlined in sec. 5.2.1 obtains a relaxed synchronized solution, which is discretized in a second step. This can be applied for two or more images. It involves the computation of the top k eigenvectors of a the matrix $S^T W S$ of size $Ns \times Ns$. For $N = 2$ images and a subspace $s = 100$, this results in a 200×200 matrix. If we want to segment a video, however, whose length is $N = 150$ frames, then we need to compute eigenvectors of a matrix of size 30000×30000. Note that, contrary to W, the matrix $S^T W S$ is in general not sparse and thus the efficient Lanczos method cannot be applied.

A different approach would be to obtain independent relaxed solutions for each image, which are synchronized during discretization using the correspondence matrices $C_{i,j}$. The feature matches can be utilized during the discretization by requiring that the segment indicator matrices of the individual images are not only close to the relaxed indicator vectors but also their inter-image similarity is strong. More precisely, we adapt the discretization program from Eq. (5.18) as follows:

$$\begin{aligned}
\min_{\mathcal{V},\mathcal{R}} \quad & \text{CoDiscret}(\mathcal{V},\mathcal{R}) = \sum_{i=1}^{N} ||V_i - \tilde{S}_i R_i||_F - \gamma \sum_{i,j=1, i\neq j}^{N} V_i^T C_{i,j} V_j \qquad (5.19) \\
\text{subject to} \quad & V_i \in \{0,1\}^{n_i \times k}, \quad V_i \mathbf{1}_k = \mathbf{1}_{n_i} \quad \text{for all } i \in \{1,\ldots,N\} \\
& R_i \in O(k) \quad \text{for all } i \in \{1,\ldots,N\} \\
& \mathcal{V} = \{V_1,\ldots,V_N\}, \mathcal{R} = \{R_1,\ldots,R_N\}
\end{aligned}$$

for $\tilde{S}_i$ obtained using Eq. (5.13) and S_i obtained as explained in sec. 5.2.1.

The above optimization problem is optimized iteratively with respect to R and V:

Optimization w. r. t. R for fixed V: Rewriting the objective function yields:

$$\text{CoDiscret}(\mathcal{V},\mathcal{R}) = 2\sum_{i=1}^{N} n_i - \text{tr}(V_i^T \tilde{S}_i R_i) - \gamma \sum_{i,j=1, i\neq j}^{N} V_i^T C_{i,j} V_j$$

Using Lagrange multipliers, as explained in Theorem 1 in [Yu and Shi, 2003], the maximum is obtained from the SVD decomposition of

$$V_i^T \tilde{S}_i = U_i \Omega \tilde{U_i}^T \quad \text{via} \quad R_i = U_i \tilde{U_i}^T \tag{5.20}$$

Optimization w. r. t. V for fixed R: Rewriting the objective function yields:

$$\text{CoDiscret}(\mathcal{V}, \mathcal{R}) = 2\sum_{i=1}^{N} n_i - \sum_{i=1}^{N} \text{tr}(V_i^T(\tilde{S}_i R_i + \gamma \sum_{j=1, j\neq i}^{N} C_{i,j} V_j)$$

Suppose that V_j above are fixed from the previous iteration. Then the maximum is obtained for

$$(V_i)_{l,m} = \begin{cases} 1 & \text{if } m = \arg\max_{m'}\{(A_i)_{l,m'}\} \\ 0 & \text{otherwise} \end{cases} \tag{5.21}$$

for $A_i = \tilde{S}_i R_i + \gamma \sum_{j=1, j\neq i}^{N} C_{i,j} V_j$.

The final algorithm is outlined in Algorithm 4.

Algorithm 4 Segmentation Synchronization during Discretization

1: Initialize W_i and C as in section 5.1.
2: Compute segmentation subspaces S_i as the eigenvectors to the k largest eigenvalues of W_i; obtain $\tilde{S}_i$ from S_i via Eq. (5.13).
3: Initialize $R_i^{(0)} \leftarrow I_k$; $t \leftarrow 1$;
4: Initialize $V_i^{(0)}$ using Eq. (5.21) with $A_i^{(0)} = \tilde{S}_i$.
5: **repeat**
6: Set $R_i^{(t)}$ from $V_i^{(t-1)}$ and $\tilde{S}_i$ using Eq. (5.20).
7: Set $V_i^{(t)}$ from $R_i^{(t)}$ and $V_i^{(t-1)}$ using Eq. (5.21) with $A_i = \tilde{S}_i R_i^{(t)} + \gamma \sum_{j=1, j\neq i}^{N} C_{i,j} V_j^{(t-1)}$
8: **until** $V_i^{(t)}$ has not changed from previous iteration for all $i \in \{1, \ldots, N\}$.

Discussion. The computation of the segmentation subspaces can be done in $O(n_i^{3/2}k)$ for all N images. The complexity of the first step of the iteration is $O(k^3)$, while the complexity of the second step is $O(n_i k^2)$ steps. Assuming that the number of the eigenvectors is negligible compared to the pixel count in the images ($k \ll n_i$), we arrive at complexity $O(\sum_i n_i^{3/2}k + \sum_i n_i k^2)$.

5.2.3 Co-saliency Region and Feature Matching

Similarly to the relaxation of the program (CSRM) in Eq. (5.11), we can relax the program (CSRFM) from Eq. (5.10) as follows:

$$\begin{aligned} \max_{Z,P} \quad & \operatorname{tr}\left(Z^T W Z\right) && (5.22)\\ \text{subject to} \quad & Z^T D Z = \mathbf{1}_k && (5.23)\\ & C_{i,j} = P_{i,j} \circ M_{i,j} \quad \text{for all} \quad i,j \in \{1,\ldots,N\}, i \neq j \\ & P_{i,j}\mathbf{1}_{n_j} \leq \mathbf{1}_{n_i}, \quad \mathbf{1}_{n_j}^T P_{i,j} \leq \mathbf{1}_{n_i}, \quad P_{i,j} \in [0,1]^{n_i \times n_j} \end{aligned}$$

where Z is related to the original variable via Eq. (5.13). In addition, we relax the elements of P to lie be real numbers in $[0, 1]$.

Then we can iteratively optimize the above function w. r. t. Z and P:

Optimize w. r. t. Z for fixed P: See sec. 5.2.1.

Optimize w. r. t. P for fixed Z: For two images (for $N > 2$ images the derivations are analogous) we can rewrite the objective function as follows:

$$\begin{aligned} \operatorname{tr}\left(Z^T W Z\right) &= \operatorname{tr}(Z_1^T W_1 Z_1 + Z_2^T W_2 Z_2) + 2\operatorname{tr}(Z_1^T C_{12} Z_2) \\ &= \operatorname{tr}(Z_1^T W_1 Z_1 + Z_2^T W_2 Z_2) + 2\operatorname{tr}(Z_2 Z_1^T (M_{1,2} \circ P_{1,2})) \end{aligned}$$

If we drop the terms in the objective which do not depend on P, then the optimization problem, which should be solved in this iteration step has the form:

$$\begin{aligned} \max_{P} \quad & \operatorname{tr}(Z_2 Z_1^T (M_{1,2} \circ P_{1,2})) && (5.24)\\ \text{subject to} \quad & P_{i,j}\mathbf{1}_{n_j} \leq \mathbf{1}_{n_i}, \quad \mathbf{1}_{n_j}^T P_{i,j} \leq \mathbf{1}_{n_i}, \quad P_{i,j} \in [0,1]^{n_i \times n_j} \end{aligned}$$

The latter program is a linear program and can be solved exactly.

The product $Z_2 Z_1^T$ is not sparse and of size $n_2 \times n_1$. However, we do not need to compute the complete matrix $Z_2 Z_1^T$, but only those elements for which $M_{1,2}$ is non zero. Since the number of feature matches F is very small (F is usually in

the order of several hundred), then the above objective has only F variables. Thus this program can be formulated efficiently and solved quickly using any standard LP solver.

The final optimization is outlined in Algorithm 5.

Algorithm 5 Co-saliency region and feature matching.

1: Initialize W_i, M, and C as in section 5.1. Compute W.
2: **repeat**
3: Solve for relaxed co-saliency regions $Z = SV_{\text{sub}}$ using Algorithm 3.
4: Solve for correspondence selection P using program in Eq. (5.24).
5: **until** P does not differ from previous step.

5.3 Estimation of Dense Correspondences

Initially we choose a sparse set of feature matches M extracted using a feature detector. In order to obtain denser set of correspondences we use a larger set M' of matches between features extracted everywhere in the image. Since this set can potentially contain many more wrong matches than M, running algorithm 1 directly on M' does not give always satisfactory results. Therefore, we prune M' based on the solution Z^* of the relaxed (CSRM) from Eq. (5.22) by combining

- Similarity between co-salient regions obtained for old feature set M. Using the embedding view of the segmentation synchronization from Fig. 5.4 this translates to euclidean distances in the joint segmentation space;
- Feature similarity from new M'.

Suppose, two pixels $x \in I_1$ and $y \in I_2$ have embedding coordinates $b_x^* \in \mathbb{R}^k$ and $b_y^* \in \mathbb{R}^k$ obtained as rows of Z^*. Then following feature similarities embody both requirements from above:

$$M''_{x,y} = M'_{x,y}((b_x^*)^T b_y^*)$$

The new set M is obtained by thresholding the values of M''. The final matching algorithm is outlined in algorithm 6.

Algorithm 6 Dense feature matching.

1: Construct M using a sparse feature detector (see sec. 5.4.1).
2: Obtain $Z^* = SV_{\text{sub}}$ as outlined in sec. 5.2.1 and Algorithm 3.
3: Construct M' using a dense feature detector (see sec. 5.4.1).
4: Compute M'': $M''_{x,y} = M'_{x,y}((b^*_x)^T b^*_y)$
5: Construct M by thresholding M'':

$$M_{x,y} = \begin{cases} M'_{x,y} & \text{iff } M''_{x,y} \geq t_c \\ 0 & \text{otherwise.} \end{cases}$$

Scale M'' such that maximal element in M'' is 1.
6: Solve $(V_{\text{dense}}, \mathcal{C}_{\text{dense}}) = \max_{V,\mathcal{C}} \text{CoSaliency}(V, \mathcal{C})$ using Algorithm 5.

5.4 Experiments

In this section we evaluate the performance of the presented algorithms. We apply them on two problems: wide-baseline stereo and video segmentation.

The first application presents the challenge of erroneous feature matches with a large outlier portion. In addition, independent segmentation of scenes viewed from widely different viewpoints tend to be quite different. Thus obtaining region and dense feature correspondences are of practical importance.

The second application presents the challenge of processing large videos and obtaining a small set of precise segments of the video volume.

5.4.1 Wide-baseline Stereo

For the following experiments we use Algorithm 5 to obtain co-salient region and feature matches. Further, we apply Algorithm 6 to obtain a dense set of feature correspondences.

Inter-image similarities. The feature correspondence matrix $M \in [0,1]^{n_1 \times n_2}$ is based on affine covariant region detector. Each detected point p has an elliptical region R_p associated with it and is characterized by an affine transformation $H_p(x) = A_p x + T_p$, which maps R_p onto the unit disk $D(1)$. For comparison, each feature is represented by a descriptor d_p extracted from $H_p(R_p)$. These descriptors can be used to evaluate the appearance similarity between two interest points p and q, and thus, to define a similarity between pixels $x \in R_p$ and and $y \in R_q$ lying in the interest point regions:

$$m_{x,y}(p,q) = e^{-\|d_p - d_q\|^2/\sigma_i^2} e^{-\|H_p(x) - H_q(y)\|^2/\sigma_p^2}$$

The first term measures the appearance similarity between the regions in which x and y lie, while the second term measures their geometric compatibility with respect to the affine transformation of R_p to R_q. Provided, we have extracted two feature sets P from I_1 and Q from I_2 as described above, the final match score $M_{x,y}$ for a pair of pixels equals the largest match score supported by a pair of feature points:

$$M_{x,y} = \max\{m_{x,y}(p,q) | p \in P, q \in Q, x \in R_p, y \in R_q\}$$

In this way, pixels on different sides of corresponding image contours in both images get connected and thus shape information is encoded in M (see fig. 5.5). The final M is obtained by pruning: retain $M_{x,y}$ for $M_{x,y} \geq t_c$, otherwise 0, where t_c is a threshold. For feature extraction we use the MSER detector [Tuytelaars and Gool, 2004] combined with SIFT descriptor [Lowe, 2004]. The choice of the detector is motivated

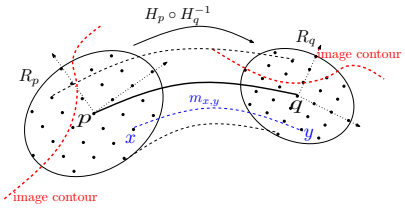


Figure 5.5: For a match between features p and q their similarity gets extended to pixel pairs, e. g. x and y.

by MSER's large support. For the computation of the dense correspondences M' in sec. 5.3 we use features extracted on a dense grid in the image and use the same descriptor.

Intra-image similarities. The matrices $W_i \in [0,1]^{n_i \times n_i}$, for each image are based on intervening contours. Two pixels x and y from the same image are considered to belong to the same segment, if there are no edges with large magnitude, which spatially separate them:

$$(W_i)_{x,y} = e^{-\max\{\|\text{edge}(z)\|^2 | z \in \text{line}(x,y)\}/\sigma_e^2}, i \in \{1,2\}$$

Algorithmic settings. The optimal dimension of the segmentation subspaces in second step of Algorithm 5 depends on the area of the segments in the images – to capture small detailed regions we need more eigenvectors. For the experiments we used $s = 50$, $k = 10$. The threshold t_c from is determined so that initially we obtain approx. $200 - 400$ matches and for our experiments it is $t_c = 3.2$.

Dataset. We conduct two experiments: (i) detection of matching regions and (ii) place recognition. For both experiments we use two datasets from the ICCV2005 Computer Vision Contest[Szeliski, 2005]: *Test4* and *Final5*, containing each 38 and 29 images of buildings. Each building is shown in several images under different viewpoints.

Experiment I – Detection of Matching Regions

In this experiment we detect matching regions, enhance the feature matches, and segment common objects in manually selected image pairs (see Fig. 5.7 – 5.9). The 30 matches with highest score in C_{dense} of the output of the dense feature matching Algorithm 6 and the top 6 matching regions according to Algorithm 5 are displayed in Fig. 5.7 – 5.9.

Finding the correct match for a given point may fail usually because (i) the appearance similarity to the matching point is not as high as the score of the best matches and therefore it is not ranked high in the initial C; or (ii) there are several matches with high scores due to similar or repeating structure. The segment-based reranking in step 4 of the matching algorithm helps on one side to boost the match score of similar features lying in corresponding segments and thus to find more correct matches (darker regions in row 1 in Fig. 5.7 – 5.9). On the other side the reranking eliminates matches connecting points in different segments and in this way resolves ambiguous correspondences (repeating structures in row 3).

To compare quantitatively the difference between the initial and the improved set of feature matches we count how many of the top 30, 60, and 90 best matches are correct. We rank them using the score from the initial and improved C respectively and show the table (5.1). The number of the correct matches in all sets is around 4 times higher than the number of the correct matches in the initial feature set.

Experiment II – Place Recognition

As in ICCV2005 Computer Vision Contest each of the two datasets *Test4* and *Final5* has been split into two subsets: exemplar set and query set. The query set contains for *Test4* 19 and for *Final5* 22 images, while the exemplar set contains 9 and 16 images respectively. Each query image is compared with all exemplars images and the matches are ranked according to the value of the co-saliency region and feature match score from Eq. (5.10) at the estimated approximate optimum. For each query there are usually several (2 up to 5) exemplars, which display the same scene viewed from different viewpoint. For all queries, which have at least k similar exemplars in the dataset, we compute how many of them are among the top k matches. Accuracy rates are presented in fig. 5.6 for *Final5* ($k = 1 \ldots 4$) and *Test4* ($k = 1 \ldots 4$). With a few exceptions the match score function ranks most of the similar exemplars as top matches.

matches	initial C	improved C
1 - 30	19%	75%
31 - 60	12%	52%
60 - 90	15%	44%

Table 5.1: Percentage of correct matches among the first 90 matches ranked with the initial and improved C. The top 90 matches are separated into 3 groups: top 30 matches, top 60 matches without the top 30, and top 90 matches without the top 60.

5.4.2 Video Segmentation

In this experiment we apply Algorithm 4 to extract co-salient regions from the frames of a video. In this way we obtain a segmentation of the video which incorporates appearance-based grouping cues in the individual frames as well as temporal information.

Inter-image similarity. We extract and track features in the video using the KLT tracker [Shi and Tomasi, 1994]. For any two frames i and j, denote by $T_{i,j}$ the set of feature tracks which involve both frames i and j. Then, the correspondence matrix $C_{i,j}$ between those two frames puts those pixels in correspondence which share a track:

$$(C_{i,j})_{x,y} = \begin{cases} 1 & \text{there is a track } t \in T_{i,j} \text{ with } x, y \in t, x \in I_i, y \in I_j \\ 0 & \text{otherwise.} \end{cases}$$

Contrary to the setup in sec. 5.4, we do not model affine deformation of the patches around the pixels in correspondence since the deformation of the frames is not that strong. Additionally, most of the correspondences relate frames which are nearby – frames which are far in the video usually share very few or no tracks at all.

The intra-image similarities are set as in sec. 5.4.

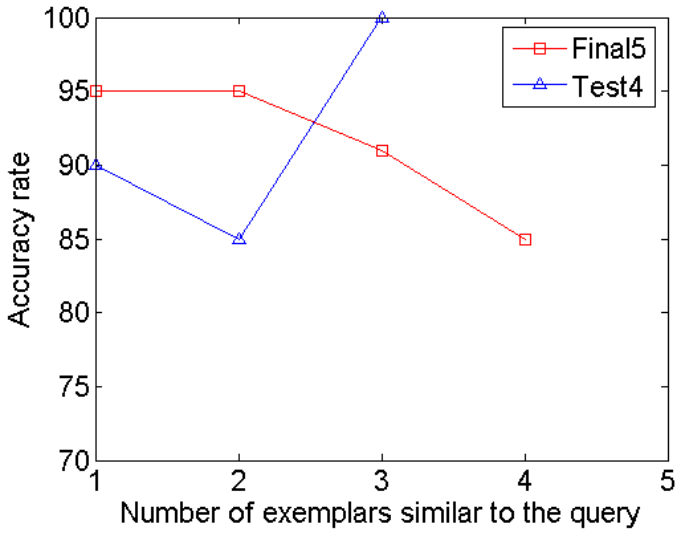


Figure 5.6: Accuracy rate in percentage for datasets *Test4* and *Final5*.

Video Segmentation. We test the proposed approach on videos obtained from YouTube [you,]. Those videos are usually low resolution and suffer from compression artifacts. In the conducted experiments we set the number of segmentation subspaces (see second step in Algorithm 4) and thus the number of resulting segments per frame to $k = 50$.

The resulting co-salient regions are displayed in Fig. 5.10 – 5.11. We obtain segments which are correctly tracked across frames, even when the shape or the size of the segment changes due to scene motion. Note that in some videos the object undergoes a motion, while in other videos the motion of the background is larger. The resulting co-salient regions should be treated as an oversegmentation of the video and as such do not always represent object parts. Also, due to the motion in the scene, new co-salient regions may emerge or disappear. For example, in the right video in Fig. 5.11 in the first two frames the side windows are represented as two regions (dark red and blue). As the vehicle undergoes rotation, the side window becomes too small to be split into two segments. As a result, after the second frame

Figure 5.7: Matching results for manually selected pairs of images from [Szeliski, 2005]. For each pair, the top 30 matches are displayed in the left column, while the top 6 matched segments according to the match score function are presented in the right column.

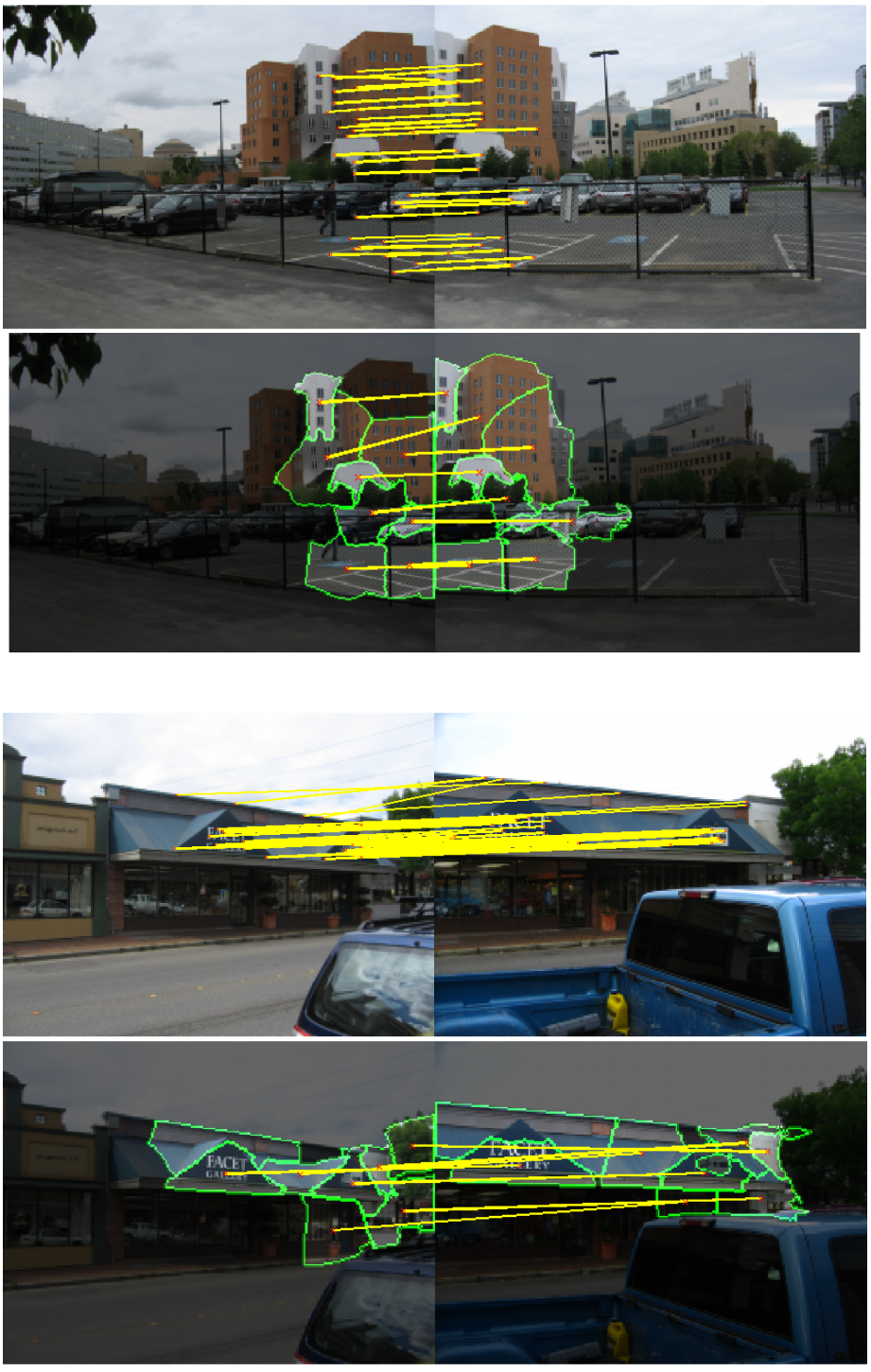


Figure 5.8: Matching results for manually selected pairs of images from [Szeliski, 2005]. For each pair, the top 30 matches are displayed in the left column, while the top 6 matched segments according to the match score function are presented in the right column.

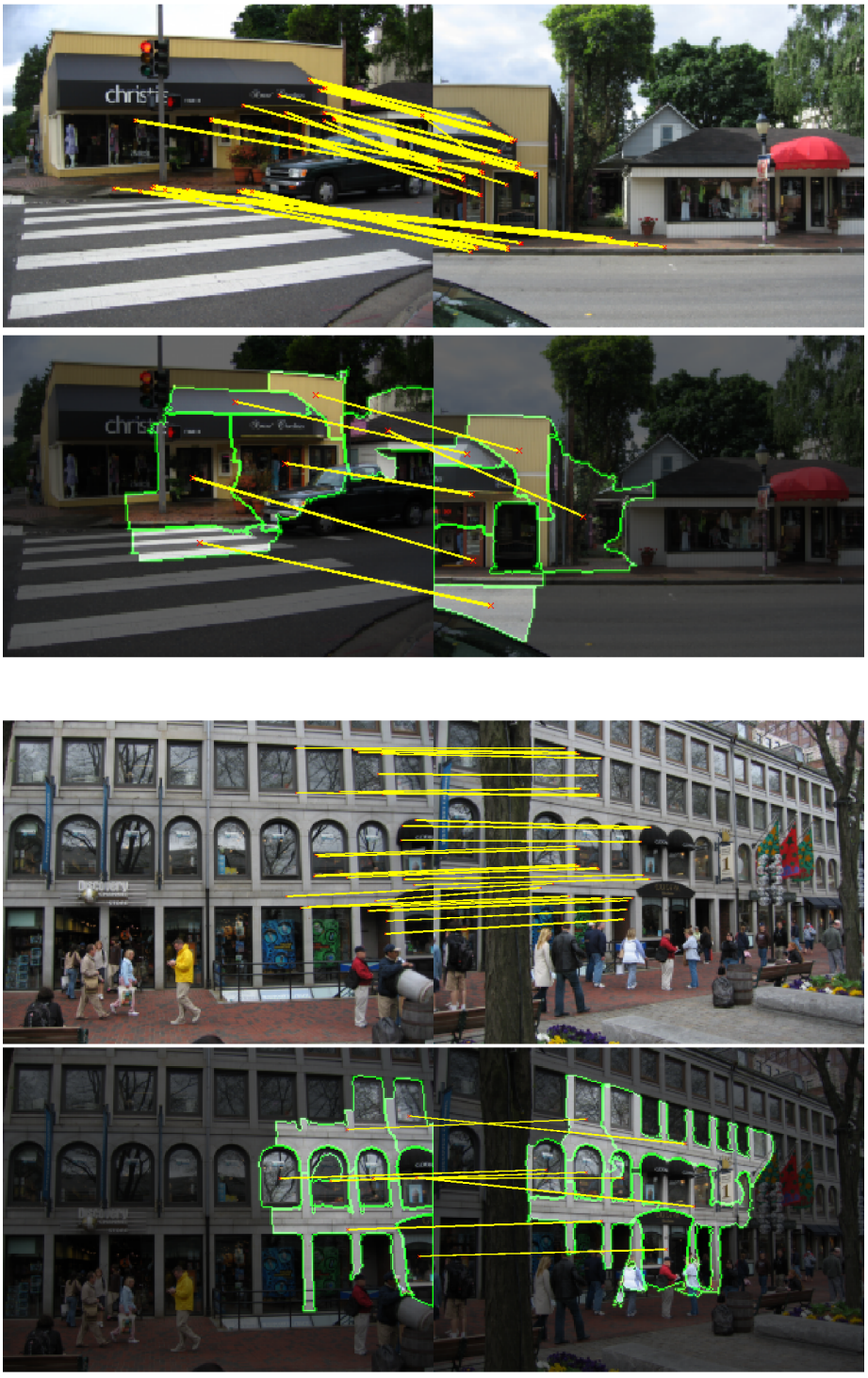


Figure 5.9: Matching results for manually selected pairs of images from [Szeliski, 2005]. For each pair, the top 30 matches are displayed in the left column, while the top 6 matched segments according to the match score function are presented in the right column.

the window becomes a single co-salient region region (dark red).

To see the advantages of the co-salient region extraction, we compare it to independent frame segmentation. In Fig. 5.12 – 5.13 we show for each of the three displayed videos one co-salient region. In addition, we show the segment from the independent frame segmentation, which has highest overlap with the co-salient region. We can see that the co-salient region consistently tracks the same scene structure and has a more stable shape.

5.5 Related Work

The presented framework touches on several research streams in computer vision.

Spectral graph matching and its applications to object recognition. Spectral approaches for weighted graph matching have been extensively studied, some of the notable works being [Umeyama, 1988b, Shapiro and Brady, 1992]. Such approaches characterize the graphs by their dominant eigenvectors. However, these eigenvectors are computed independently for each graph and thus often do not capture *co*-salient structures as the eigenvectors of the JIG. Reasoning in the JIG helps extracting representations from two images which contain relevant information for the matching of the particular pair of images.

Our approach has also been inspired by the work on simultaneous object recognition and segmentation [Yu et al., 2002], which uses spectral clustering in a graph capturing the relationship between image pixels and object parts. Our work has parallels in machine learning [Ham et al., 2004], where based on correct partial correspondences between manifolds the goal is to infer their complete alignment using regularization based on similarities between points on the manifolds.

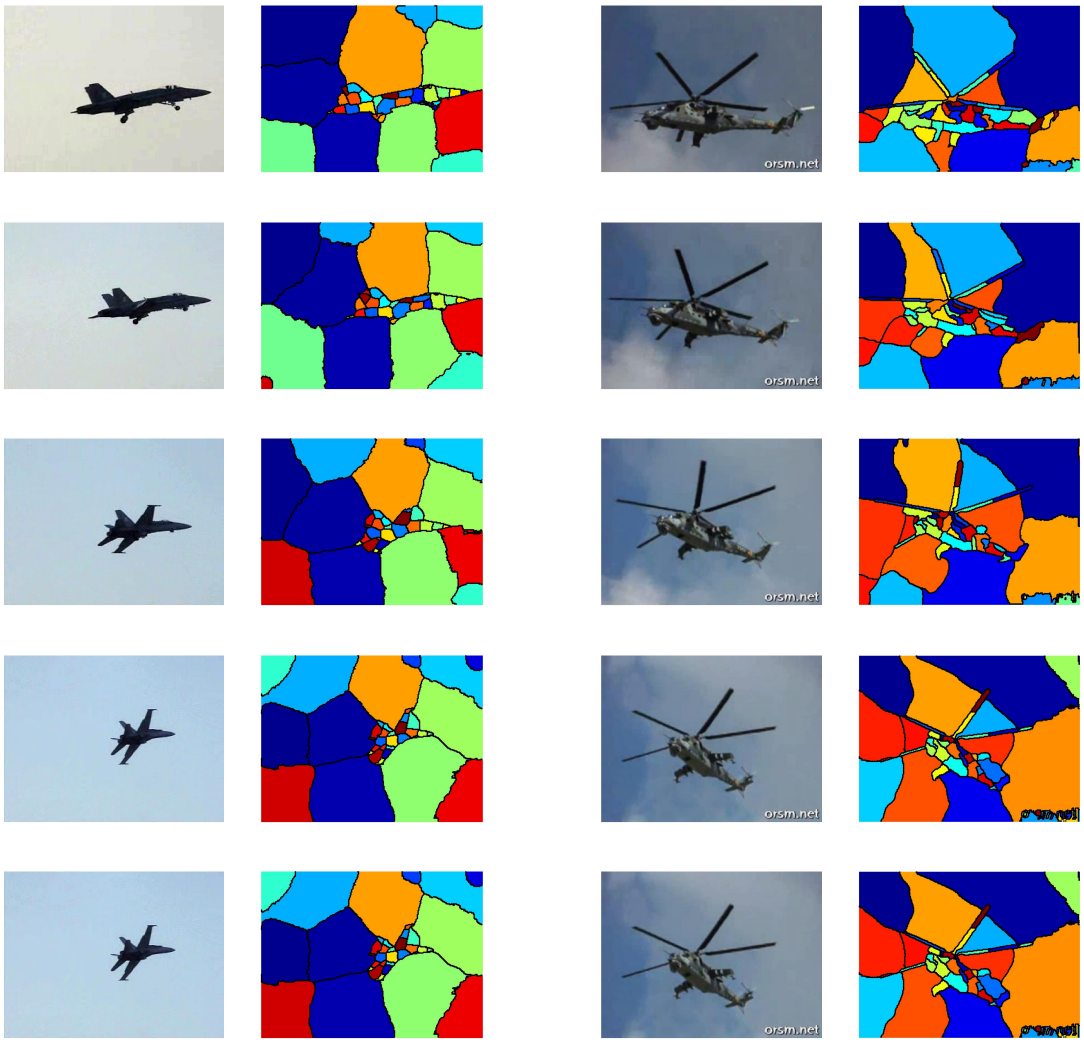


Figure 5.10: For selected frames of a video we show the original image on the left and the output of the co-saliency region matching on the right. Each set of co-salient regions has a unique color in the shown video.

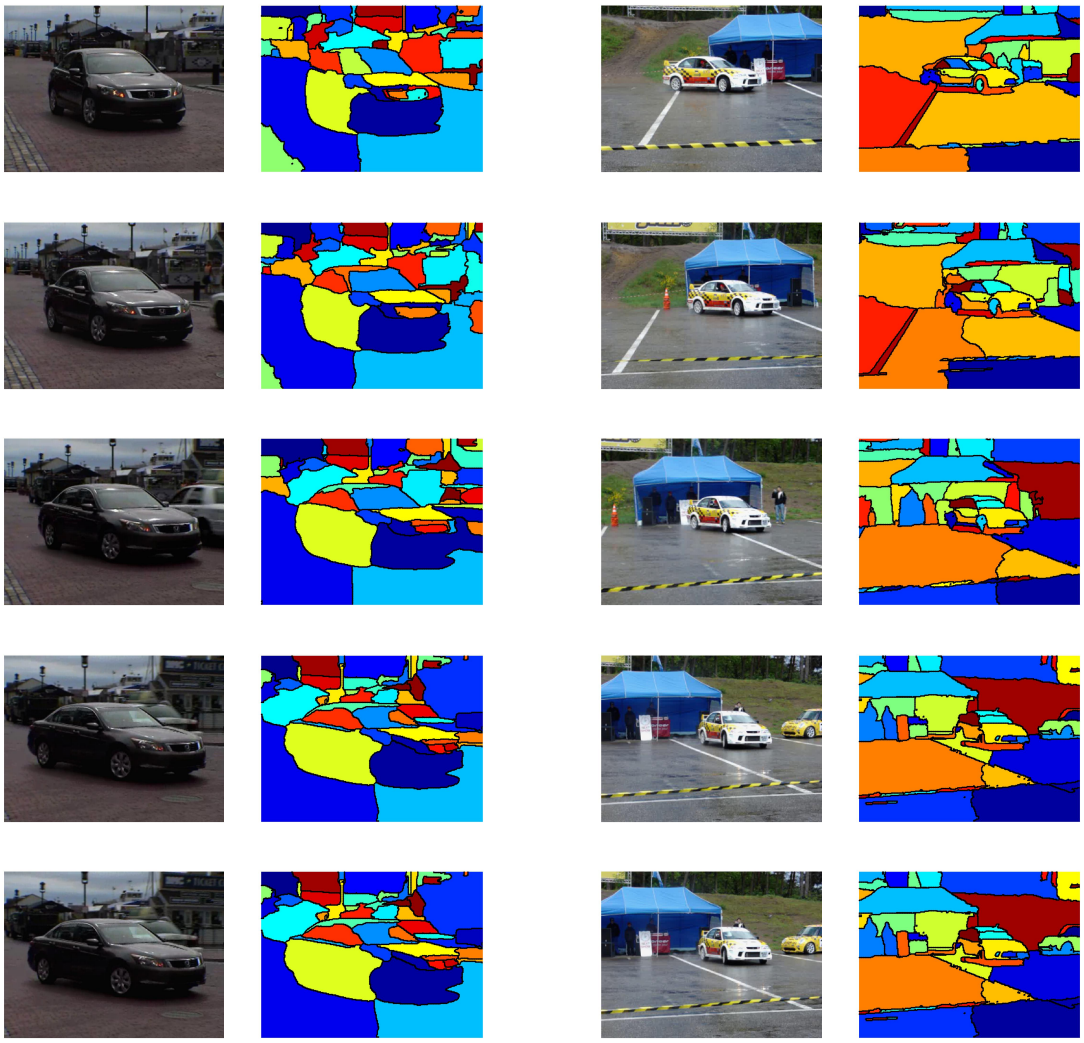


Figure 5.11: For selected frames of a video we show the original image on the left and the output of the co-saliency region matching on the right. Each set of co-salient regions has a unique color in the shown video.

Figure 5.12: For selected consecutive frames for a video we show a single set of co-salient regions on the left and the region with the highest overlap from independent frame segmentation on the right.

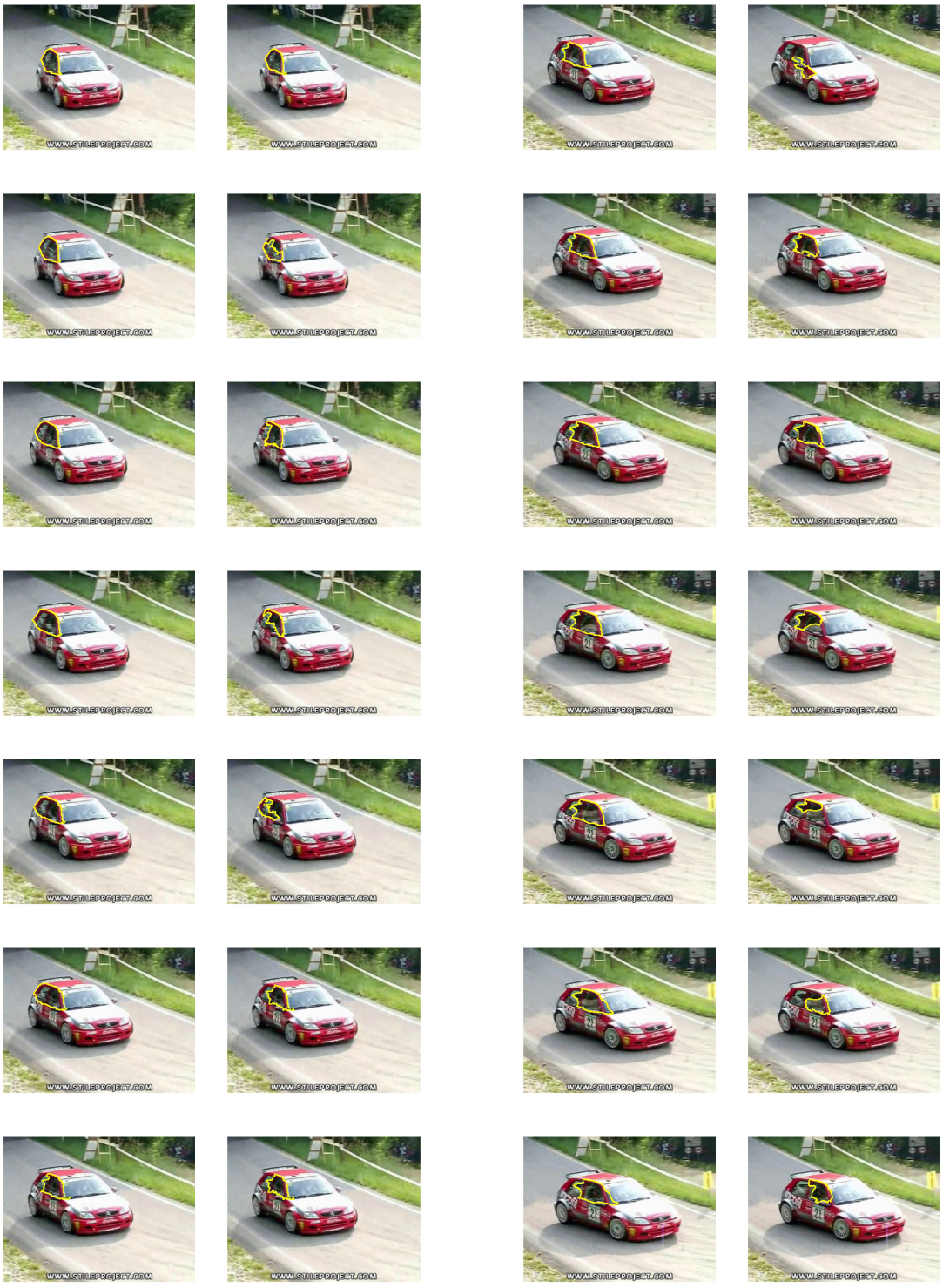


Figure 5.13: For selected consecutive frames for a video we show a single set of co-salient regions on the left and the region with the highest overlap from independent frame segmentation on the right.

Co-segmentation. The term co-segmentation was coined by [Rother et al., 2006] and refers to simultaneous segmentation of two images and extraction of the common objects. The authors use a generative graphical model, which consists of a smoothness prior for segmentation and appearance-based model for the common object. This model is usually described as histogram. Depending, on how the models of the objects in the two images are compared, several approaches are introduced which lead to different optimization problems [Mukherjee et al., 2009, Hochbaum and Singh, 2009, Vicente et al., 2010].

Joint image representation is also used by [Boiman and Irani, 2006], who define a similarity between images as the composability of one of the images from large segments of the other image. Independently extracted regions have been used already for wide-baseline stereo [Schaffalitzky and Zisserman, 2001] and object recognition [Russell et al., 2006]. In the latter work the authors deal with the variability in the segmentation by using multiple segmentations of each image.

5.6 Conclusion

In this chapter we have defined co-salient regions in a set of images and presented algorithms for their extraction. Co-salient regions are segments in images which are coherent and distinct from their surroundings, while at the same time they are similar to each other where the similarity is quantified in terms of feature similarities. The presented algorithms are based on the Normalized Cuts framework [Shi and Malik, 2000, Yu and Shi, 2003].

We apply the presented approach to two problems. First, we use co-salient regions to obtain region correspondences and dense point correspondences in the context of wide-baseline stereo. We show that in this way we can improve local feature matching.

Second, we segment video incorporating motion information. The resulting co-salient regions can be thought of as segment tracks which represent a coarsening of the video. We will use these segment tracks for object silhouette extraction and object recognition in Chapter 4.

Appendices

Appendix A

Proofs of Theorem 1 and 2

Theorem 1. *Define the set*

$$\mathcal{P}^*_{CM} = \{X \in \mathcal{P}_{CM} | \sum_{\substack{(i,j)\in bin_b(m)\\(k,l)\in bin_b(m)}} X_{ijkl} = \min\{ch^1_m, ch^2_m\} \text{ for all bins } m \text{ and schemes } b\} \tag{A.1}$$

as a subset of $\mathcal{P}_{CM}$ for which the chord correspondence variable X is constraint through the chordiograms.

Then we can show that each $X^* \in \mathcal{P}^*_{CM}$ is a minimizer of the problem (CM) with data terms W^{mbins} and the minimum of this problem is analytically computable using the chordiogram:

$$W^{mbins} \cdot X^* = \sum_{b=-1}^{B} \alpha_b ||ch^{b,1} - ch^{b,2}||_1$$

for weights $\alpha_b = 2^b$.

Proof. First we will show that the chordiogram matching lower bounds the problem (CM). In a second step, we will show that for $X^* \in \mathcal{P}^*_{\mathrm{CM}}$ the bound turns into an equality.

Lower bound for (CM). Suppose that $X \in \mathcal{P}_{\mathrm{CM}}$, in particular it satisfies constraints (2.21) and (2.22). Then, one can show that

$$
\begin{aligned}
& \sum_{b=-1}^{B} \alpha_b ||\mathrm{ch}^{b,1} - \mathrm{ch}^{b,2}|| && \text{(A.2)} \\
= & \sum_{b=-1}^{B} \alpha_b || \sum_{i,j} \mathrm{ch}_{ij}^{b,1} - \sum_{k,l} \mathrm{ch}_{kl}^{b,2} ||_1 \quad \text{(definition of chordiogram)} \\
= & \sum_{b=-1}^{B} \alpha_b || \sum_{i,j} (\sum_{k,l} X_{ijkl}) \mathrm{ch}_{ij}^{b,1} - \sum_{k,l} (\sum_{i,j} X_{ijkl}) \mathrm{ch}_{kl}^{b,2} ||_1 \text{ (by Eq. (2.21))} \\
= & \sum_{b=-1}^{B} \alpha_b || \sum_{i,j,k,l} (\mathrm{ch}_{ij}^{b,1} - \mathrm{ch}_{kl}^{b,2}) X_{ijkl} ||_1 \\
\leq & \sum_{i,j,k,l} \sum_{b=-1}^{B} \alpha_b ||\mathrm{ch}_{ij}^{b,1} - \mathrm{ch}_{kl}^{b,2}||_1 X_{ijkl} \quad \text{(norm inequality and positivity of } X) \\
= & \sum_{i,j,k,l} W_{ij;kl}^{\mathrm{mbins}} X_{ijkl} \quad \text{(by Eq. (2.6))} \\
= & \ W^{\mathrm{mbins}} \cdot X
\end{aligned}
$$

Minimizers for (CM). As a second step, we will show that for each $X^* \in \mathcal{P}_{\mathrm{CM}}^*$ the above inequality turns into an equality.

Consider for a moment a concrete bin m using finest binning scheme $b = -1$. We can use the bin indices of the chords to define a matching between them. More precisely, we put chords in correspondence if they lie in the same bin. After this procedure there will remain chords which are not in any correspondence. The correspondence assignment for such chords is deferred for a coarser binning scheme.

Now we turn to the description of the correspondence assignment for a particular binning scheme b. For the sake of brevity we will skip the binning scheme index b. Suppose that X gives a chord mapping for which d_m denotes the number of chords from shape 1 from bin m mapped to chords from shape 2 which are also in bin m; a_m chords from shape 1 from bin m mapped to chords not in bin m; and c_m chords from shape 1 not in bin m mapped to chords from shape 2 in bin m. From the definition

of d_m we have

$$d_m = \sum_{\substack{(i,j)\in \mathrm{bin}(m) \\ (k,l)\in \mathrm{bin}(m)}} X_{ijkl} \tag{A.3}$$

Then, it is clear that $\mathrm{ch}^1_m = a_m + d_m$ and $\mathrm{ch}^2_m = d_m + c_m$, hence $|\mathrm{ch}^1_m - \mathrm{ch}^2_m| = |a_m - c_m|$. Also, it is clear that $\sum_{i,j,k,l} |(\mathrm{ch}^1_{ij})_m - (\mathrm{ch}^2_{kl})_m|_1 X_{ijkl} = a_m + c_m$. Thus, we can express the gap in the above inequality derivation for a single binning scheme as:

$$W^{\mathrm{bin}} \cdot X - ||\mathrm{ch}^1 - \mathrm{ch}^2||_1 = \sum_m (a_m + c_m - |a_m - c_m|)$$

X is a minimizer for (CM) exactly when the above gap equals zero, i. e. $a_m + c_m - |a_m - c_m| = 0$ for all m. This is equivalent to $\min\{a_m, c_m\} = 0$, which holds iff $d_m = \min\{\mathrm{ch}^1_m, \mathrm{ch}^2_m\}$. The latter identity together with Eq. (A.3) gives the desired characterization.

Now, suppose that $d^b_m = \min\{\mathrm{ch}^{1,b}_m, \mathrm{ch}^{2,b}_m\}$ holds for all binning schemes from the definition of multiple-bin distance between chords from Eq. (2.6). This means that all gaps disappear:

$$W^{\mathrm{b,\ bin}} \cdot X - ||\mathrm{ch}^{b,1} - \mathrm{ch}^{b,2}||_1 = 0 \quad \text{for all} \quad b \in \{-1, 0, \ldots, B\}$$

Combining the above inequalities together with weights α_b gives the equality relationship in the theorem. □

Theorem 2. *Suppose that $X^*_{cm,orig}$ is the minimizers of problem (CM) in Eq. (2.20) using the data terms W^{orig}. Further, X^*_{pm} is the minimizer of problem (PM) in Eq. (2.16) using the data terms W^{mbins}.*

Then, the following relationship holds:

$$C(W^{orig} \cdot X^*_{bm,orig}) \leq \sum_{b=-1}^{B} \alpha_b ||ch^{b,1} - ch^{b,2}||_1 \leq W^{mbins} \cdot X^*_{pm}$$

for a positive constant C.

Proof. We show both inequalities separately.

First inequality. The left inequality is result of a direct application of Lemma 1 from [Indyk and Thaper, 2003]. Note that the point sets, which are considered in [Indyk and Thaper, 2003], correspond to the chords sets in our setting. Then there is a constant C such that the chordiogram distance is lower bounded by the weighted bipartite matching among the chords, where the weights are defined in terms of the L_1 distance in the chord feature space:

$$C(W^{\text{orig}} \cdot X^*_{cm,\text{orig}}) \leq \sum_{b=-1}^{B} \alpha_b ||\text{ch}^{b,1} - \text{ch}^{b,2}||_1$$

Second inequality. From the previous theorem, we have that the middle term is the minimum of the (CM) problem with data terms W^{mbins}. It is known that the minimum of the (CM) problem interpreted as a bipartite matching is smaller that the minimum of the (PM) problem interpreted as linear programming relaxation of the graph matching. This gives us the second inequality. □

Bibliography

[you,] Youtube. Website. `http://www.youtube.com`.

[Basri et al., 1998] Basri, R., Costa, L., Geiger, D., and Jacobs, D. (1998). Determining the similarity of deformable shapes. *Vision Research*, 38(15-16):2365–2385.

[Belongie et al., 2002] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522.

[Biederman, 1987] Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147.

[Binford, 1971] Binford, T. O. (1971). Visual perception by computer. In *IEEE Conference on Systems and Control*.

[Blum, 1973] Blum, H. (1973). Biological shape and visual science. *Journal of Theoretical Biology*, 38(2):205–287.

[Boiman and Irani, 2006] Boiman, O. and Irani, M. (2006). Similarity by composition. In *Neural Information Processing Systems*.

[Borenstein et al., 2004] Borenstein, E., Sharon, E., and Ullman, S. (2004). Combining top-down and bottom-up segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Borgefors, 1986] Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, 34(3):344–371.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

[Carlsson, 1999] Carlsson, S. (1999). Order structure, correspondence and shape based categories. In *International Workshop on Shape, Contour and Grouping*.

[Chekuri et al., 2005] Chekuri, C., Khanna, S., Naor, J., and Zosin, L. (2005). A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM Journal on Discrete Mathematics*, 18(3):608–625.

[Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619.

[Cootes, 1995] Cootes, T. (1995). Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59.

[Cour et al., 2005] Cour, T., Benezit, F., and Shi, J. (2005). Spectral segmentation with multiscale graph decomposition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Cyr and Kimia, 2004] Cyr, C. and Kimia, B. (2004). A similarity-based aspect-graph approach to 3d object recognition. *International Journal of Computer Vision*, 57(1):5–22.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

[Evans et al., 1992] Evans, A., Thacker, N., and Mayhew, J. (1992). Pairwise representations of shape. In *Internatonal Conference on Pattern Recognition*, pages 133–133.

[Everingham and et. al, 2005] Everingham, M. and et. al (2005). The 2005 pascal visual object classes challenge. In *MLCW*, pages 117–176.

[Felzenszwalb and Huttenlocher, 2004] Felzenszwalb, P. and Huttenlocher, D. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181.

[Felzenszwalb et al., 2008] Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Felzenszwalb and Schwartz, 2007] Felzenszwalb, P. and Schwartz, J. (2007). Hierarchical matching of deformable shapes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Ferrari et al., 2008] Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Ferrari et al., 2009] Ferrari, V., Jurie, F., and Schmid, C. (2009). From images to shape models for object detection. *International Journal of Computer Vision*.

[Ferrari et al., 2006] Ferrari, V., Tuytelaars, T., and Gool, L. V. (2006). Object detection by contour segment networks. In *European Conference on Computer Vision*.

[Ferrari et al., 2004] Ferrari, V., Tuytelaars, T., and Van Gool, L. (2004). Integrating multiple model views for object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

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

[Fritz and Schiele, 2008] Fritz, M. and Schiele, B. (2008). Decomposition, discovery and detection of visual categories using topic models. In *CVPR*.

[Goemans and Williamson, 1995] Goemans, M. X. and Williamson, D. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6).

[Gold and Rangarajan, 1996] Gold, S. and Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388.

[Golub and Loan, 1989] Golub, G. and Loan, C. V. (1989). *Matrix Computation*. The Johns Hopkins University Press.

[Gorelick and Basri, 2009] Gorelick, L. and Basri, R. (2009). Shape based detection and top-down delineation using image segments. *IJCV*, 83(3).

[Grant and Boyd, 2010] Grant, M. and Boyd, S. (2010). CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`.

[Grauman and Darrell, 2007] Grauman, K. and Darrell, T. (2007). The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760.

[Grimson, 1990] Grimson, W. (1990). *Object recognition by computer: The role of geometric constraints*. The MIT Press, Cambridge, MA.

[Grimson and Lozano-Perez, 1987] Grimson, W. and Lozano-Perez, T. (1987). Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482.

[Gu et al., 2009] Gu, C., Lim, J., Arbelaez, P., and Malik, J. (2009). Recognition using regions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Ham et al., 2004] Ham, J., Lee, D., and Saul, L. (2004). Semisupervised alignment of manifolds. In *AISTATS*.

[Hochbaum and Singh, 2009] Hochbaum, D. and Singh, V. (2009). An efficient algorithm for co-segmentation. In *International Conference on Computer Vision*.

[Huttenlocher et al., 1993] Huttenlocher, D., Klanderman, D., and Rucklige, A. (1993). Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863.

[Indyk and Thaper, 2003] Indyk, P. and Thaper, N. (2003). Fast image retrieval via embeddings. In 3^{rd} *International Workshop on Statistical and Computational Theories of Visio*.

[Joachims, 1999] Joachims, T. (1999). Making large-scale svm learning practical. In *Advances in Kernel Methods - Support Vector Learning*.

[Kendall, 1989] Kendall, D. (1989). A survey of the statistical theory of shape. *Statistical Science*, 4(2):116–120.

[Kimia et al., 1995] Kimia, B., Tannenbaum, A., and Zucker, S. (1995). Shapes, shocks, and deformations i: the components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15(3).

[Koenderink and Doorn, 1979] Koenderink, J. and Doorn, A. (1979). The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216.

[Koffka, 1935] Koffka, K. (1935). *Principles of Gestalt Psychology*. Lund Humphries.

[Kolmogorov and Zabih, 2002] Kolmogorov, V. and Zabih, R. (2002). What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:65–81.

[Kriegman and Ponce, 1990] Kriegman, D. and Ponce, J. (1990). Computing exact aspect graphs of curved objects: Solids of revolution. *International Journal of Computer Vision*, 5:119–135.

[Lafferty et al., 2003] Lafferty, J., McCallum, A., and Pereira, F. (2003). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*.

[Lamdan et al., 1990] Lamdan, Y., Schwartz, J., and Wolfson, H. (1990). Affine invariant model-based object recognition. *Robotics and Automation, IEEE Transactions on*, 6(5):578–589.

[Latecki and Lakamper, 2000] Latecki, L. and Lakamper, R. (2000). Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190.

[Latecki et al., 2000] Latecki, L., Lakamper, R., and Eckhardt, U. (2000). Shape descriptors for non-rigid shapes with a single closed contour. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1.

[Leibe et al., 2008] Leibe, B., Leonardis, A., and Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3).

[Leordeanu et al., 2007] Leordeanu, M., Hebert, M., and Sukthankar, R. (2007). Beyond local appearance: Category recognition from pairwise interactions of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Leung and Malik, 2001] Leung, T. and Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44.

[Levin and Weiss, 2006] Levin, A. and Weiss, Y. (2006). Learning to combine bottom-up and top-down segmentation. In *European Conference on Computer Vision*.

[Liebelt et al., 2008] Liebelt, J., Schmid, C., and Schertler, K. (2008). Viewpoint-independent object class detection using 3D Feature Maps. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Ling and Jacobs, 2007] Ling, H. and Jacobs, D. (2007). Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:286–299.

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

[Lu et al., 2009] Lu, C., Latecki, L. J., Adluru, N., Yang, X., and Ling, H. (2009). Shape guided contour grouping with particle filters. In *International Conference on Computer Vision*.

[Maji and Malik, 2009] Maji, S. and Malik, J. (2009). Object detection using a max-margin hough transform. In *CVPR*.

[Malisiewicz and Efros, 2008] Malisiewicz, T. and Efros, A. A. (2008). Recognition by association via learning per-exemplar distances. In *CVPR*.

[Marr, 2010] Marr, D. (2010). *Vision: A computational investigation into the human representation and processing of visual information.* Henry Holt and Co., Inc.

[Martin et al., 2004] Martin, D., Fowlkes, C., and Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[Matas, 2004] Matas, J. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767.

[Mcneill and Vijayakumar, 2006] Mcneill, G. and Vijayakumar, S. (2006). Hierarchical procrustes matching for shape retrieval. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:885–894.

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

[Mokhtarian et al., 1997] Mokhtarian, F., Abbasi, S., and Kittler, J. (1997). Efficient and robust retrieval by shape content through curvature scale space. *Image Databases and Multi-Media Search*, pages 51 – 58.

[Mori, 2005] Mori, G. (2005). Guiding model search using segmentation. In *Proceedings of 10th International Conference on Computer Vision*, volume 2, pages 1417–1423.

[Mukherjee et al., 2009] Mukherjee, L., Singh, V., and Dyer, C. (2009). Half-integrality based algorithms for cosegmentation of images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*

[Opelt et al., 2006] Opelt, A., Pinz, A., and Zisserman, A. (2006). A boundary-fragment-model for object detection. In *European Conference on Computer Vision.*

[Osada et al., 2002] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. (2002). Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832.

[Palmer, 1999] Palmer, S. E. (1999). *Vision science: Photons to phenomenology*. The MIT Press.

[Pentland, 1986] Pentland, A. (1986). Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331.

[Pizer et al., 1999] Pizer, S., Fritsch, D., Yushkevich, P., Johnson, V., and Chaney, E. (1999). Segmentation, registration, and measurement of shape variation via image object shape. *IEEE Transactions on Medical Imaging*, 18(10):851–865.

[Ravishankar et al., 2008] Ravishankar, S., Jain, A., and Mittal, A. (2008). Multi-stage contour based detection of deformable objects. In *European Conference on Computer Vision*.

[Ren et al., 2005] Ren, X., Fowlkes, C., and Malik, J. (2005). Cue integration in figure/ground labeling. In *Neural Information Processing Systems*.

[Rosch et al., 1976] Rosch, E., Mervis, C., Gray, W., Johnson, D., and Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive psychology*, 8(3):382 – 439.

[Rother et al., 2006] Rother, C., Minka, T., Blake, A., and Kolmogorov, V. (2006). Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Rothganger et al., 2003] Rothganger, F., Lazebnik, S., Schmid, C., and Ponce, J. (2003). 3D Object Modeling and Recognition Using Affine-Invariant Patches and Multi-view Spatial Constraints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Rubin, 1915] Rubin, E. (1915). *Synsoplevede figurer*. Gyldendal.

[Russell et al., 2006] Russell, B., Efros, A. A., Sivic, J., Freeman, B., and Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Savarese and Fei-Fei, 2007] Savarese, S. and Fei-Fei, L. (2007). 3d generic object categorization, localization and pose estimation. In *International Conference on Computer Vision*.

[Schaffalitzky and Zisserman, 2001] Schaffalitzky, F. and Zisserman, A. (2001). Viewpoint invariant texture matching and wide baseline stereo. In *International Conference on Computer Vision*.

[Sebastian et al., 2003] Sebastian, T., Klein, P., and Kimia, B. (2003). On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125.

[Sebastian et al., 2004] Sebastian, T., Klein, P., and Kimia, B. (2004). Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571.

[Shapiro and Haralick, 1979] Shapiro, L. and Haralick, R. (1979). Structural descriptions and inexact matching. technical report CS79011-R, Computer Science, Virginia Tech.

[Shapiro and Brady, 1992] Shapiro, R. and Brady, M. (1992). Feature-based correspondence: an eigenvector approach. *Image Vision Comput.*, 10(5):283–288.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 442(8).

[Shi and Tomasi, 1994] Shi, J. and Tomasi, C. (1994). Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Shilane et al., 2004] Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T. (2004). The Princeton shape benchmark. In *Shape Modeling International*, Genova, Italy.

[Shotton et al., 2005] Shotton, J., Blake, A., and Chipolla, R. (2005). Contour-based learning for object detection. In *International Conference on Computer Vision*.

[Shotton et al., 2009] Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23.

[Siddiqi et al., 1999] Siddiqi, K., Shokoufandeh, A., Dickinson, S., and Zucker, S. (1999). Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13–32.

[Sivic et al., 2006] Sivic, J., Schaffalitzky, F., and Zisserman, A. (2006). Object Level Grouping for Video Shots. *International Journal of Computer Vision*, 67:189–210.

[Sivic and Zisserman, 2008] Sivic, J. and Zisserman, A. (2008). Efficient visual search for objects in videos. *Proceedings of the IEEE*, 96(4):548–566.

[Srinivasan et al., 2010] Srinivasan, P., Zhu, Q., and Shi, J. (2010). Many-to-one contour matching for describing and discriminating object shape. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Sturm, 1999] Sturm, J. F. (1999). Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*.

[Szeliski, 2005] Szeliski, R. (2005). Iccv2005 computer vision contest. `http://research.microsoft.com/iccv2005/Contest/`.

[Toshev et al., 2009] Toshev, A., Makadia, A., and Daniilidis, K. (2009). Shape-based object recognition in videos using 3d synthetic object models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Toshev et al., 2010] Toshev, A., Taskar, B., and Daniilidis, K. (2010). Object detection via boundary structure segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Tu and Yuille, 2004] Tu, Z. and Yuille, A. (2004). Shape matching and recognition–using generative models and informative features. In *Seventh Europen Conference on Computer Vision*, pages 195 – 209.

[Tuytelaars and Gool, 2004] Tuytelaars, T. and Gool, L. V. (2004). Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1).

[Ullman and Basri, 1991] Ullman, S. and Basri, R. (1991). Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:992–1006.

[Umeyama, 1988a] Umeyama, S. (1988a). An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703.

[Umeyama, 1988b] Umeyama, S. (1988b). An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703.

[Varma and Zisserman, 2005] Varma, M. and Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81.

[Vicente et al., 2010] Vicente, S., Kolmogorov, V., and Rother, C. (2010). Cosegmentation revisited: Models and optimization. In *European Conference on Computer Vision*.

[Wills et al., 2003] Wills, J., Agarwal, S., and Belongie, S. (2003). What went where. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[Yoshida and Sakoe, 1982] Yoshida, K. and Sakoe, H. (1982). Online handwritten character recognition for a personal computer system. *IEEE Transactions on Consumer Electronics*, pages 202–209.

[Yu et al., 2002] Yu, S. X., Gross, R., and Shi, J. (2002). Concurrent object recognition and segmentation by graph partitioning. In *Neural Information Processing Systems*.

[Yu and Shi, 2003] Yu, S. X. and Shi, J. (2003). Multiclass spectral clustering. In *International Conference on Computer Vision*.

[Zhang and Lu, 2003] Zhang, D. and Lu, G. (2003). Evaluation of mpeg-7 shape descriptors against other shape descriptors. *Multimedia Systems*, 9(1).

[Zhang et al., 2007a] Zhang, J., Marszalek, M., Lazebnik, S., and C., S. (2007a). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238.

[Zhang et al., 2007b] Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C. (2007b). Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2).

[Zhu et al., 2007] Zhu, Q., Song, G., and Shi, J. (2007). Untangling cycles for contour grouping. In *Proceedings of the International Conference on Computer Vision*.

[Zhu et al., 2008] Zhu, Q., Wang, L., Wu, Y., and Shi, J. (2008). Contour context selection for object detection: A set-to-set contour matching approach. In *European Conference on Computer Vision*.