Learning and Optimization for Shape-based Representations

Iasonas Kokkinos

HABILITATION À DIRIGER DES RECHERCHES
DE
L'UNIVERSITÉ PARIS-EST

HABILITATION À DIRIGER DES RECHERCHES

Spécialité: **Informatique**

présentée par

Iasonas Kokkinos

Learning and Optimization for Shape-based Representations

Soutenue le 25 Septembre 2013 devant le jury composé par

Rapporteurs:

M.	Daniel	Cremers	Professeur, Université Technique de Munich
M.	Ales	Leonardis	Professeur, Université de Birmingham
M.	Andrew	Zisserman	Professeur, Université de Oxford

Examinateurs:

M.	Rachid	Deriche	Directeur de Récherche, INRIA - Sophia Antipolis
M.	Frederic	Jurie	Professeur, Université Caen
M.	Jean-Christophe	Pesquet	Professeur, Université Paris-Est

Contents

1 Introduction **1**
- 1.1 Shape, grouping, and recognition . . . 1
- 1.2 Prior work: low-, high-, and mid-level vision . . . 3
- 1.3 Contributions . . . 11

2 Contour Detection in Natural Images **21**
- 2.1 Learning-based boundary detection . . . 21
- 2.2 Learning-based symmetry detection . . . 28
- 2.3 Contour grouping with linear-fractional programming . . . 34

3 Invariant Image and Surface Descriptors **43**
- 3.1 Dense scale-invariant image descriptors . . . 43
- 3.2 Segmentation-aware descriptors . . . 48
- 3.3 Scale-invariant surface descriptors . . . 53
- 3.4 Intrinsic Shape Context descriptors . . . 57

4 Learning Shape Models for Objects and Actions **65**
- 4.1 Learning object deformation models . . . 65
- 4.2 Learning hierarchical shape models . . . 70
- 4.3 Learning mid-level models of actions in videos . . . 77

5 Efficient Optimization of Shape-based Models **87**
- 5.1 Branch-and-Bound for Deformable Part Models . . . 87
- 5.2 A^* for hierarchical object models . . . 94
- 5.3 Reinforcement learning for facade parsing . . . 97

6 Future Research **109**

7 Curriculum Vitae **113**

Chapter 1

Introduction

Socrates: Let shape be this for us, what always happens to accompany color. Perhaps this is acceptable, or do you seek something else? ...
Meno: But this is so naive, Socrates ...
some discussion follows
Socrates: ... That which limits a solid, that is shape.
Meno: And what do you say of color, Socrates?
Socrates: You are overbearing Meno, pressing a senior for answers.

– Plato, *Meno*

1.1 Shape, grouping, and recognition

Shape is used in computer vision in a broad sense to describe whatever is not affected by changes in appearance, and in a more restricted manner to refer to geometric information, such as contours that correspond to surface boundaries. This distinction is reminiscent of the two alternative definitions provided in the introductory quote: the first defines shape as being the complement of color, while the second makes explicit reference to an object and its boundary.

Shape is currently used in vision mainly in its first, broader, sense through features such as Shape Context [1], Scale-Invariant Feature Transforms (SIFT) [2] or Histograms-of-Gradients (HOG) [3], which describe shape around a point as a distribution on invariant features, such as a histogram of gradients. By virtue of being probabilistic these features provide to subsequent tasks such as recognition a robust description of shape.

This however leaves a geometric way of representing shape, as in the second sense, to be desired: the use of geometric information coming in the form of contours has delivered results in computer vision tasks as diverse and challenging as 3D instance recognition [4–7] tracking [8, 9] and surface reconstruction [10–15]. In all these works exploiting the geometrical aspects of contours had played a key role in simplifying the underlying problems. One can therefore anticipate that similar approaches could simplify object recognition in its modern, statistical setting.

An example from [4] that illustrates the potential, but also the complex role of shape in recognition is the pair of images in Fig. 1.1 and Fig. 1.2. It is hard to recognize the object in Fig. 1.1 - as reported in [4], 9 out of 10 of people presented with the image failed at recognizing it after 60 seconds. In the second image however an object can easily be perceived - 3 out of 10 subjects

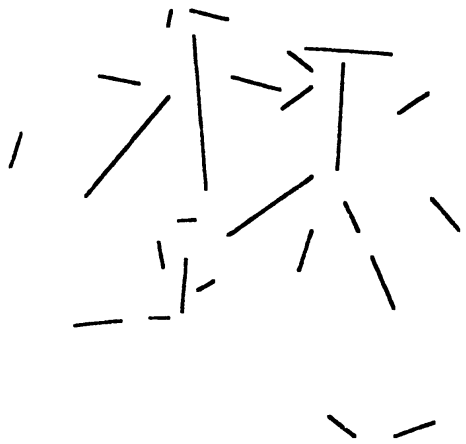


Figure 1.1: From [4]: All line segments in this image belong to a single object - so the segmentation problem can be seen as solved. Still, the object's identity is not obvious.

recognized it in less than 5 seconds and 7 out of 10 recognized it within 60 seconds. The only difference is the introduction of an additional line on the bottom-right, which on its own is not that different from the others. The different impression cannot be therefore attributed to the added line itself, but rather to the longer contour whose grouping was facilitated by the added line. One can imagine that finding this semi-cycle can 'trigger' a wheel hypothesis that in turn triggers the object hypothesis.

This example illustrates how shape grouping can help when dealing with a large set of object categories. When viewing these minimal sketches we have no color, shading, or other contextual cues to help us guess the object's identity. Therefore thousands of categories, and a continuum of pose parameters may have to be considered; this is the case for the first image. However, in the second image a more distinctive piece of evidence is available in the form of a long grouping. Its effect can be understood as hinting at a model that involves this grouping; once this model is suggested, in its turn it manages to explain the remaining image observations. This was the main theme of geometric 3D recognition [4–7] before the advent of statistical learning techniques.

Extending this scheme to deal with broader classes of models relates to the top-down/bottom-up computation problem: namely, 'bottom-up', image-based guidance is necessary for efficient detection, while 'top-down', object-based knowledge can disambiguate and help reliably interpret a given image. Organizing computation to combine both modes of operation has been motivated based on arguments of computational efficiency and accuracy, as well as evidence related to human vision and perception. But it remains elusive to construct practical computer vision algorithms based on this principle.

The starting point for the research presented in this thesis has been my earlier research on the combination of low- and high- level vision problems [16, 17]. As also demonstrated in several other works of the same period [16, 18–23], the main positive conclusion is that a joint treatment can yield better performance for both low- and high- level tasks; but the complexity of the sys-

tems that result from combining separate low- and high- level modules can become excruciating. Starting from [24] I have therefore been focused on shape-based approaches.

The main motivation for this direction, and the works presented in this thesis, is the understanding that shape, and more generally, grouping, can build a bridge between low- and high-level representations, by delivering 'mid-level' structures of intermediate complexity. There are two main advantages to this: first, having a single representation, that starts from the image measurements and finishes at the object hypotheses; this can be trained and optimized in an integrated, end-to-end manner. Second, developing a common, shared mid-level representation that can be used by multiple categories, thereby reducing the amount of necessary data during training and computation time during testing.

These ideas have some broad appeal in computer vision, but shape grouping is still not a part of the object recognition mainstream. A main reason for this is that finding reliable and repeatable groupings is challenging; for example occlusions and boundary detector failures can disrupt grouping and break subsequent stages down. Other open questions include what is the best way of representing shape mathematically, how to model its statistical variations, exploit it in discriminative training, or use it for efficient detection.

Still, the combination of shape, grouping, and recognition remains the object of intensive research, as documented e.g. in [25]. The following section aims at presenting in some more detail the major research threads developed around this problem, before outlining in Section 1.3 the relevant contributions presented in this thesis.

1.2 Prior work: low-, high-, and mid-level vision

This section aims at providing background regarding the interplay of shape, grouping and recognition in vision. We highlight in particular (i) that efficient computation dictates a joint low- and high- level processing, (ii) the 'mid-level' problems that emerge at the interface of low- and high-level vision, and (iii) how shape-based and grouping-based techniques fit in with this setting.

Sparse low-level image representations and grouping

Sparse, or feature-based, approaches to low-level vision represent an image in terms of a small set of structures extracted by a front-end mechanism. The underlying assumption is that some small set of features contains all of the task-relevant information. This is illustrated by Attneave's cat example [26]: as shown in Fig. 1.3(a), we can make a complex inference about the identity, pose and state of an object while only employing corners and their connections. This suggests that as far as recognition is concerned, we can compactly encode an image in terms of these few structures.

There are certain caveats to this. First, the sketch shown in (a) contains multiple interior edges, which correspond to surface creases or tips. These are typically hard to detect in natural images, in particular for smoothly textured objects. As shown in Fig. 1.3(b), removing such edges makes it impossible to recover surface information. This highlights that information can be irrecoverably lost through front-end processing, which should therefore operate well in the 'high-recall' regime. Second, as shown in Fig. 1.3(c), forming a new image with corners at the midpoints of the original

Figure 1.2: From [4], continued: The addition of one line (to the right of the picture) facilitates grouping - which in turns makes it easier to perceive the object that gave rise to the configuration.

lines gives the same impression. This suggests that the corners themselves are not really crucial, but rather it is their overall connection pattern that contains the pose information.

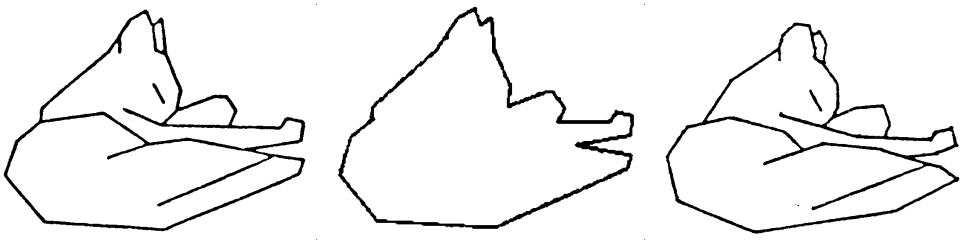
This brings us to the task of grouping, which can be understood as the identification of image structures that 'go together'. The solution space to such problems typically has a combinatorial structure - for instance in contour grouping the solution can indicate which edgels form a group and in what order they appear. The resulting optimization problems are therefore often hard, but in particular for contour grouping efficient algorithms can exploit the problem's mostly 1D structure. In specific, Dynamic Programming was applied to contour grouping in [28], A^* in [29] and coarse-to-fine DP in [30]. Moreover, if one normalizes the contour's energy function by the curve length this renders the grouping invariant to scale changes; again the resulting optimization problem can be globally optimized, as shown originally in [31], and then extended in [32–35].

Shape variability and high-level vision

In object category detection shape deformations are accommodated primarily with deformable part models (DPM), defined in terms of a set of parts that can deform with respect to each other. The state y of a DPM with P parts involves a set of position vectors, $\mathbf{x}_i, i = 1, \dots P$ and global transformation parameters $\mathcal{T}$ that encode scale changes or rotations:

$$y = \{\mathbf{x}_1, \dots, \mathbf{x}_P, \mathcal{T}\}. \tag{1.1}$$

Each of the position vectors can correspond to any of the N image pixels, meaning that in principle N^P part combinations should be considered. Furthermore the continuous transformation



(a) (b) (c)

Figure 1.3: Image (a) shows Attneave's sketch, from [26], and (b), (c) its perturbations from [27], and [4] respectively. As demonstrated in image (a), we can perceive a complex object configuration even when we are provided with only a reduced shape-based representation, coming in the form of corners, junctions and their connections. But as shown in image (b), object boundaries need to be complemented with internal contours to estimate pose; and as shown in image (c), by displacing the corners of image (a) to the line midpoints, the information is encoded in the corner arrangement, rather than the corners themselves.

parameters $\mathcal{T}$ need to be estimated, while for the goal of scene analysis N_O object categories may need to be simultaneously dealt with. This can result in a complexity in the order of

$$C(N_O, \mathcal{T}, N, P) = |\mathcal{T}| \cdot N_O \cdot N^P, \tag{1.2}$$

where $\mathcal{T}$ is understood as a set of quantized transformation parameters and $|\mathcal{T}|$ denotes its cardinality. Coming up with algorithms of a smaller complexity is thus crucial for fast object detection.

The Pictorial Structure works of Felzenszwalb and collaborators [36–38] express the object's scoring function as a Markov Random Field (MRF) with a tree-structured graph topology. For the case of a 'star-shaped' graph, i.e. a tree of depth one, the model's score function is:

$$s(y, x, w) = \sum_{i=0}^{P} u_i(x, y_i, w_i^u) + \sum_{i=1}^{P} v_i, (y_i, y_0, w_i^p) \tag{1.3}$$

where node 0 is the 'root' and the remaining P parts connect to the root; the score is formed as the sum of unary, appearance fidelity terms at the part positions $y_0, \ldots, y_P$, and pairwise, geometric consistency terms between the locations $y_1, \ldots, y_P$ and the root's location, y_0. Maximizing this score with Max-Product [39] has a complexity of $O(PN^2)$, but for special cases of the pairwise term the Generalized Distance Transform (GDT) technique [40] can bring the complexity down to $O(PN)$; furthermore, as detailed in Section 5.1 in [41] we have developed a branch-and-bound algorithm that maximizes Eq. 1.3 with a best-case complexity of $O(P \log N)$, making to a large extent the image size N the least important aspect of the complexity of optimization.

These acceleration techniques do not deal though with the $|\mathcal{T}|$ factor in Eq. 1.2: DPMs require exhaustive search over scale, and potentially in-plane rotations, while out-of-plane rotations are currently dealt with by using different, view-specific mixtures, which amounts to an increase in

(a) (b)

Figure 1.4: Examples of images where one cannot name any single object, yet there exists a 'mid-level' interpretation in terms of simple geometric primitives: (a) 'blue segment', by V. Kandinsky and (b) a chimeric object, from [42].

N_O. Furthermore, the linear complexity in N_O amounts to using a separate detector per category; this makes large-scale recognition infeasible, when working with hundreds, or thousands of categories. It is at this point that image-based (or, 'bottom-up') information can help detection.

Perceptual organization, shared parts and geometric constraints

A straightforward way to accelerate detection through bottom-up processing is to use sparse image representations; this allows us to cut the N term down to the number of interest point locations, and also to potentially suggest proper values of $\mathcal{T}$ by exploiting scale- and orientation- invariant front end processing. This approach was extensively explored in conjunction with the learning of part-based models from the responses of interest point detectors [43–50]. But treating an image as a set of corners/junctions misses out on the way in which these are organized - and one can expect that larger structures can convey more information and further increase efficiency.

This is one the main tenets of perceptual organization: before identifying objects in an image, we can already perceive certain structures that are extended, regular, and unlikely to have occurred from an accidental configuration of unrelated objects. So we have the apparent traces of some object whose identity and pose remains to be determined. Microscopy, astrophysical, or abstract art images are prominent examples where this happens; as illustrated in Fig. 1.4, despite having never encountered any of the observed structures, we can still decompose what we see into a small set of structures and relations, in terms of which we can, at least partially, interpret the image.

This observation opens up two avenues for accelerating object detection: first, we can use the structures formed by perceptual organization as the basis for a common 'dictionary' across multiple categories. Namely, if a few structures are common across multiple categories, the computational cost of any task related to extracting and processing these shared parts will be amortized as N_O becomes large. Therefore, only the 'object assembly' task remains, which presumably should be of lower complexity than directly matching the object to the input.

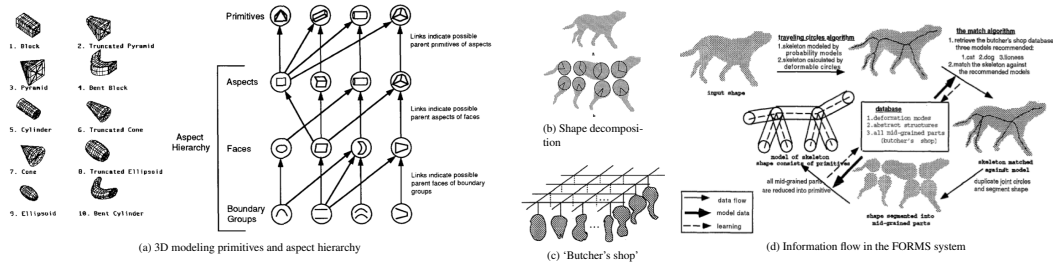


Figure 1.5: Mid-level parts in the aspect hierarchy of [61] and the FORMS system [27].

Second, we can use properties of the formed structures to guide the search over object identity or transformation parameters. Namely, if we constrain object parts to be placed so that their boundaries coincide with grouped image contours we can limit the position, scale, or object orientation, depending on the grouping's complexity.

Early works: Geometry-based 3D object recognition

These ideas were pursued since the earliest days of object recognition. The use of general grouping laws as an intermediary between the image and objects can be traced back to the hierarchical cylindrical/ribbon representations advocated by Marr in [51] and the ACRONYM system [52] respectively. The SCEPRO system [53] and the principles behind it, laid out in [4], expanded the repertoire of contour relations to include other 'gestalt' cues, such as co-linearity and proximity, using non-accidentality as a common measure for quantifying the importance of different cues. The GROPER system [54] used convexity-based line groupings to represent and detect different man-made structures by computing relationships among such groups. Ettinger [55] used curvature-guided contour grouping to represent objects hierarchically as combinations of contours, and explored methods of using this representation to search a library of objects.

Coming to constraining search using grouping, the use of geometric constraints and the associated combinatorics of grouping-based recognition are covered in [56], where it is demonstrated that the expected complexity of matching goes down from exponential to polynomial in the number of image structures if segmentation is available. Viewpoint invariant contour configurations are used to index into object libraries in [4], while other works such as [57–59] use combinations of local features to shortlist and initialize the matching of 3D object models to image observations. This direction was pushed further in the geometric invariant program [5, 7, 60]; a main theme in these works was the computation of invariants from point or contour groups their use to index object libraries in a transformation-invariant manner.

Transition to deformable categories

The early geometric works were focused on rigid object recognition, or in the best case dealt with parametric deformations, such as opening scissors [6]. The representation of deformable object categories in terms of shape groupings was explored in the 90's with more abstract representations, such as aspect hierarchies [61], grouping of 3D geometric primitives [62], shock graphs [63, 64],

the FORMS system of [27] and also the grouping-based pose estimation models of [65]. These approaches were predominantly geometrical, relying on representing and detecting objects with contour-based techniques; in several among them, in particular [61] and [27] the use of shared mid-level parts was clearly articulated and hard-wired into the inference schemes. But given the limited power of contour detection methods at that time, these works either took the segmentation for granted, or were evaluated in scenes with only a small amount of visual clutter.

Shared dictionaries for multi-class and multi-view voting

In the beginning of the previous decade, after consolidating the first successes of category recognition [66–69], the problems related to shared parts were revisited from a statistical perspective. One of the first works in this direction [70] proposed the sharing of weak learners for multi-view and multi-class object detection, using region-based features as inputs to decision stumps. In [71] this idea was applied to the implicit shape model (ISM) of [46], and it was demonstrated that one can use it to share a library of contours across multiple shape-based categories. In its original, point-based formulation, the ISM was also extended to multi-view [72] and multi-class [73] recognition by employing a shared dictionary across different views/categories; more recent work in [74] revisited the dictionary construction stage to ensure a more distinctive per-class distribution of votes for multi-class detection, while [75] combined voting with 3D geometric models recovered using structure-from-motion [76, 77].

These techniques have demonstrated that sharing computation can attain a complexity that is sublinear in the viewpoints, or classes. However, most of these works rely on voting, which discards the correspondence information [6]; this means that the features that give rise to a detection are 'lost' in the voting process and it is therefore hard to train such models. Partial remedies have been proposed in [78] by discriminatively setting the codebook weights and in [79] by reducing the effects of overcounting weights. However it has still not been demonstrated that voting-based techniques compare favorably to energy-based models for detection, such as DPMs [80] - actually, if the final verification stage is not used the performance of voting in itself is dramatically lower than DPMs, as demonstrated e.g. in [74]. So voting can at best be seen as a preliminary attention mechanism that triggers the application of more elaborate models, rather than a full-blown detection algorithm.

Shared structures in hierarchical, energy-based models

Turning to energy-based detection schemes, the integration of shared parts in category recognition has been pursued in the hierarchical, grammatical, or compositional framework. One of the earliest works in this direction was [27], but the most influential works came in the previous decade, together with the first data-driven category recognition systems.

In the 'image parsing' work of S.C. Zhu and collaborators [20, 81] image interpretation is pursued in terms of AND-OR grammars, which can be seen as templates for defining hierarchical mixtures of models. Grouping algorithms for lines, rectangles, parallelism and higher-order relationships, such as 'butting', 'inside' or 'above' are used to guide a stochastic search over generative models for buildings [82], faces [83], deformable categories [84] and scenes [85].

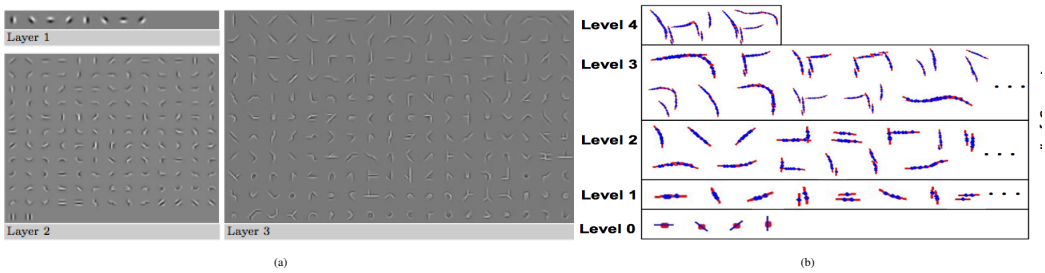


Figure 1.6: Learned contour-based mid-level features by Fidler *et al.* [86] and Zhu *et al.* [87].

A data-driven approach to discovering the shared structure of object categories was proposed by [86] who used agglomerative clustering to recover hierarchical, shape-based representations. For this purpose recursive clustering was used, starting from a Gabor filterbank's output and gradually recovering increasingly complex structures until in the end forming the whole object. In [88] it was demonstrated that this kind of representation can be used for object detection with a complexity that is sublinear in the number of categories.

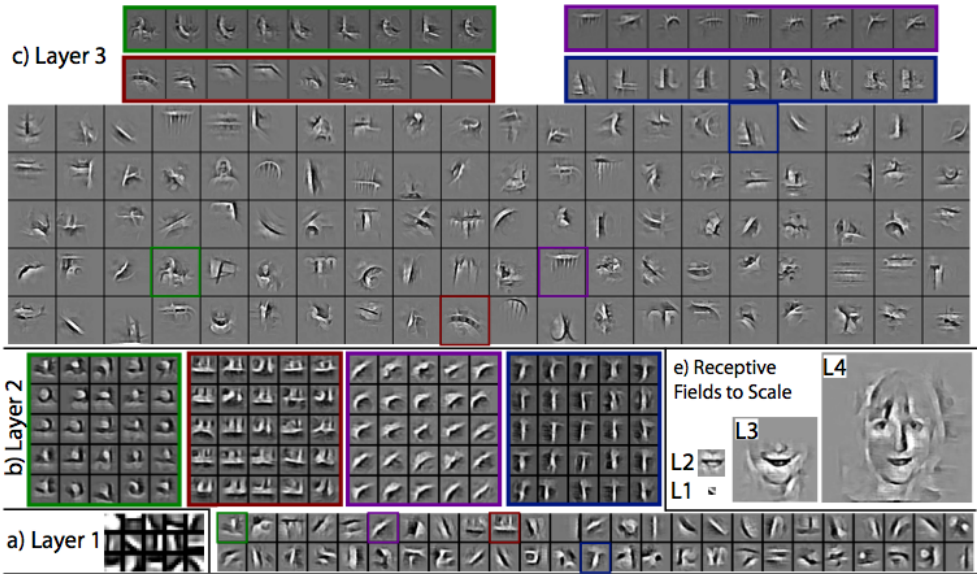
In a similar vein [87] and later [89] use unsupervised learning to recover an hierarchy of tests which can simultaneously be used for detection with multiple categories and views. They demonstrate that the learned shared parts correspond to generic grouping rules at the lowest levels and more semantic configurations at the higher parts of the hierarchy.

A region-based approach is pursued for both learning and detection in [90] and to exploit the shared structures among similar categories in [91, 92]. During training, object models are discovered by computing the maximal common subtree from a set of hierarchical image segmentations, while during testing, the hierarchical segmentation of a new image is matched to the learned tree, which allows for the simultaneous detection and segmentation of objects.

In [93] a Steiner tree formulation is used to learn and perform inference on a hierarchical model. The authors use approximate optimization to identify the optimal manner of putting together low- and intermediate- level structures within images of an object category.

Recently some works have also explored part sharing from a multi-task learning perspective, without necessarily focusing on the computational efficiency aspect; part sharing [94] and root sharing [95] were demonstrated to facilitate the learning of category models from only a limited number of training images. Similarly, [96] use sparse coding to express multiple part filters on a common basis and accelerate detection with DPMs; in [97] this is shown to also yield improved performance on a large-scale category recognition task if properly integrated during training. In [98] locality sensitive hashing [99] is used to index into a large library of model parts, cutting down computation time with lookup-based operations; this facilitates constant-time retrieval of the top-K scoring parts, but the part combination is not addressed.

Finally, the comeback of neural networks [100–111] has shown that given large enough datasets, large hierarchical networks trained essentially with back-propagation [103] can outperform systems that are using hand-crafted features. Ideas such as feature sharing and hierarchical processing are at the core of such models as shown in Fig. 1.7, and presumably play also a role in their success, yet it is still not clear how to exploit them for fast and accurate detection - with the exception



(a)

Figure 1.7: Learned mid-level features in the convolutional network of [100].

of [108], current validation of such models is mostly focused on classification/labeling tasks. One can anticipate that in the next years several of the low-level features used in current vision models will be replaced by such learned features and that some more classic problems computer vision will be 'injected', or revisited through the design of appropriate models, as e.g. in [112] and [113]. The main challenge is to make the best use of their flexibility, while staying in control of their computational complexity - as we can do for instance with deformable part models.

Mid-level segmentation and object proposals

A complementary direction of research that fits with the overall scheme described here is the exploration of mid-level segmentation algorithms to come up with object proposals, effectively avoiding the exhaustive search over object transformations. Some of the earliest works in this direction [114–116] considered coming up with multiple, complementary, and potentially overlapping, segmentations of an image and using the segments as proposals for object hypotheses, mostly in the context of learning. In [91, 117], hierarchical segmentation was used to shortlist detection hypotheses, while in [118] Steiner trees were used to search through a set of segmentations for the pairing of regions to object hypotheses.

A recent twist to this idea is to learn how to come up with good object proposals. Learning 'objectness' measures was proposed in [119] to propose bounding boxes of objects and extended in [120] by introducing faster features and better ranking algorithms. In [121] this idea was advanced to propose regions instead of bounding boxes, using segmentation functions that incorporate Gestalt features in CRF training. This approach was pushed further in [122] where using multiple segmentations seeded at different image points, and computed at different scales, was demonstrated to outperform similar segmentation algorithms [123–125] at the task of recovering

segments that correspond to object hypotheses. The current state-of-the-art method of [126] uses hierarchical segmentation in multiple color spaces to achieve high recall; as mentioned in the first section, this is crucial in order to ensure that object hypotheses do not 'fall through the cracks' of image segmentation. Recent advances in improving fast boundary detection [127] will most likely make this objective easier to achieve.

1.3 Contributions

The main goal of the works presented in this thesis is to integrate shape, and grouping in general, into the detection of deformable object categories. The research presented here has started from relatively simple shape-based models and detection algorithms [24, 128, 129] and then proceeded to increasingly elaborate models for both low-level [35, 130–132] and high-level tasks [41, 133–135], while more recently opening up to other problems that lend themselves to a geometric or grouping-based treatment [136–139]. For simplicity, the remainder of this thesis is organized thematically rather than temporally.

Chapters 2 and 3 present contributions to low-level problems. In Chapter 2 we address the detection of geometric image features, namely boundaries and symmetry axes. We present advances in (i) training feature detectors with machine learning techniques that are resilient to imprecise and ambiguous labeling information and (ii) contour grouping based on a fractional programming formulation of the minimum cost ratio problem. In Chapter 3 we turn to the extraction of invariant features from images. We present techniques to (i) extract dense scale-invariant shape descriptors from images (ii) exploit soft segmentation to remove background clutter and (iii) extend these ideas to surfaces.

Chapters 4 and 5 present contributions to high-level problems. Chapter 4 addresses the task of learning shape-based models from unregistered and unsegmented data. We present advances in (i) the automated construction of statistical shape models, (ii) the training of discriminative classifiers for hierarchical shape models, and (iii) the training of grouping-based models for action categories. Chapter 5 covers new contributions in efficient optimization with shape-based models. We address the problems of (i) detecting deformable models with Branch-and-Bound (ii) parsing hierarchical shape models using Hierarchical A^* and (ii) parsing facades with shape grammars using Reinforcement Learning.

The advances in Chapter 5 are the ones that most closely address the objective of computational efficiency, which has been the theme of this introduction, and all rely on some form of bottom-up/top-down computation, as anticipated. In hindsight, our fastest detection algorithm [41, 135] is using the dense HOG features of [80], while in our facade parsing work [140] we use densely evaluated classifiers. So, the use of shape in the form of contours did not prove to be strictly necessary - this is also the case in convolutional networks, where mid-level features are learned directly from the intensity. However, understanding grouping has been crucial to the development of our algorithms: our starting point has been the use of Hierarchical A^* for shape parsing [133], while Hierarchical A^* was originally developed in [33] in conjunction with contour grouping: both for contours and objects, the main question is how to figure out quickly how and what to put together.

Bibliography

[1] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. PAMI*, vol. 24, no. 4, pp. 509–522, 2002.

[2] D. Lowe, "Object Recognition from Local Scale-Invariant Features," in *Proc. ICCV*, 1999.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005.

[4] D. Lowe, *Perceptual Organization and Visual Recognition*. Kluwer, 1985.

[5] J. L. Mundy and A. Zisserman, Eds., *Geometric invariance in computer vision*. Cambridge, MA, USA: MIT Press, 1992, ISBN: 0-262-13285-0.

[6] W. E. L. Grimson, *Object Recognition by Computer*. MIT Press, 1991.

[7] A. Zisserman, D. A. Forsyth, J. L. Mundy, C. A. Rothwell, J. Liu, and N. Pillow, "3D object recognition using invariance," *Artificial Intelligence*, vol. 78, pp. 239–288, 1995.

[8] R. Deriche and O. D. Faugeras, "Tracking line segments," in *Proc. ECCV*, 1990.

[9] A. Blake and M Isard, *Active Contours*. Springer Verlag, 1998.

[10] C. Schmid and A. Zisserman, "The Geometry and Matching of Lines and Curves Over Multiple Views," *Int.l Journal of Computer Vision*, vol. 40, no. 3, pp. 199–233, 2000.

[11] N. Navab, R. Deriche, and O. D. Faugeras, "Recovering 3D motion and structure from stereo and 2D token tracking," in *Proc. ICCV*, 1990.

[12] R. Cipolla and P. Giblin, *Visual Motion of Curves and Surfaces*. Cambridge University Press, 1998.

[13] M. Prasad, A. W. Fitzgibbon, A. Zisserman, and L. J. V. Gool, "Finding Nemo: Deformable object class modelling using curve matching," in *Proc. CVPR*, 2010.

[14] G. Klein and D. W. Murray, "Improving the agility of keyframe-based SLAM," in *Proc. ECCV*, 2008.

[15] T. J. Cashman and A. W. Fitzgibbon, "What Shape Are Dolphins? Building 3D Morphable Models from 2D Images," *IEEE Trans. PAMI*, vol. 35, no. 1, pp. 232–244, 2013.

[16] I. Kokkinos and P. Maragos, "An Expectation Maximization approach to the synergy between object categorization and image segmentation," in *Proc. ICCV*, 2005.

[17] ——, "Synergy Between Image Segmentation and Object Recognition Using the Expectation Maximization Algorithm," *IEEE Trans. PAMI*, vol. 31, no. 8, pp. 1486–1501, 2009.

[18] M. Rousson and N. Paragios, "Shape priors for level set representations," in *Proc. ECCV*, 2002.

[19] M. P. Kumar, P. Torr, and A. Zisserman, "Obj-cut," *Proc. CVPR*, 2005.

[20] Z. W. Tu, X Chen, A. Yuille, and S.-C. Zhu, "Image Parsing: Unifying Segmentation, Detection, and Recognition," *Int.l Journal of Computer Vision*, vol. 63, no. 2, pp. 113–140, 2005.

[21] D. Cremers, N. Sochen, and C. Schnorr, "Multiphase dynamic labelling for variational recognition-driven image segmentation," in *Proc. ECCV*, 2004.

[22] D. Cremers, F. Tischhauser, J. Weickert, and C. Schnorr, "Diffusion Snakes: Introducing Statistical Shape Knowledge into the Mumford-Shah Functional," *Int.l Journal of Computer Vision*, vol. 50, no. 3, pp. 295–313, 2002.

[23] D. Cremers, "Dynamical Statistical Shape Priors for Level Set-Based Tracking," *IEEE Trans. PAMI*, vol. 28, no. 8, pp. 1262–1273, 2006.

[24] I. Kokkinos, P. Maragos, and A. Yuille, "Bottom-up and top-down object detection using primal sketch features and graphical models," in *Proc. CVPR*, 2006.

[25] S. Dickinson and Z. Pizlo, Eds., *Shape Perception in Human and Computer Vision*. Springer, 2013.

[26] F. Attneave, "Some informational aspects of visual perception," *Psychological Review*, vol. 61, pp. 183–193, 1954.

[27] S. C. Zhu and A. Yuille, "FORMS: A Flexible Object Recognition and Modeling System," *Int.l Journal of Computer Vision*, vol. 20, no. 3, 1996.

[28] I. Cohen and I. Herlin, "Curves matching using geodesic paths," in *Proc. CVPR*, 1998.

[29] J. Coughlan and A. Yuille, "Bayesian A* tree search with expected o(n) convergence rates for road tracking," in *Proc. EMMCVPR*, 1999.

[30] C Raphael, "Coarse-to-Fine Dynamic Programming," *IEEE Trans. PAMI*, vol. 23, no. 12, 2001.

[31] I. Jermyn and H. Ishikawa, "Globally optimal regions and boundaries as minimum ratio weight cycles.," *IEEE Trans. PAMI*, vol. 23, pp. 1075–1088, 2001.

[32] S. Wang, T. Kubota, J. M. Siskind, and J. Wang, "Salient Closed Boundary Extraction with Ratio Contour.," *IEEE Trans. PAMI*, vol. 27, no. 4, pp. 546–561, 2005.

[33] P. Felzenszwalb and D McAllester, "A Min-Cover Approach for Finding Salient Curves," in *POCV*, 2006.

[34] T. Schoenemann, S. Masnou, and D. Cremers, "The Elastic Ratio: Introducing Curvature into Ratio-based Globally Optimal Image Segmentation," *IEEE Trans. Im. Proc.*, vol. 20, no. 9, pp. 2565–2581, 2011.

[35] I. Kokkinos, "Highly accurate boundary detection and grouping," in *Proc. CVPR*, 2010.

[36] P. Felzenszwalb and D. Huttenlocher, "Pictorial Structures for Object Recognition," *Int.l Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[37] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. CVPR*, 2008.

[38] P. F. Felzenszwalb, R. B. Girshick, and D. A. McAllester, "Cascade object detection with deformable part models," in *Proc. CVPR*, 2010.

[39] M. Jordan, "Graphical Models," *Statistical Science*, vol. 19, pp. 140–155, 2004.

[40] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," in *Proc. CVPR*, 2004.

[41] I. Kokkinos, "Rapid deformable object detection using dual-tree branch-and-bound," in *Proc. NIPS*, 2011.

[42] I. Biederman, "Recognition-by-components: a theory of human image understanding," *Psychological Review*, no. 2, pp. 115–147, 1987.

[43] M. Welling, M. Weber, and P. Perona, "Unsupervised learning of models for recognition," in *Proc. ECCV*, 2000.

[44] S. Agarwal and D Roth, "Learning a sparse representation for object detection," in *Proc. ECCV*, 2006.

[45] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. CVPR*, 2003.

[46] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Proc. ECCV*, SLCV workshop, 2004.

[47] G Csurka, C Dance, L Fan, J Willamowski, and C Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV*, 2004.

[48] R. Fergus, P. Perona, and A. Zisserman, "A sparse object category model for efficient learning and exhaustive recognition," in *Proc. CVPR*, 2005.

[49] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering object categories in image collections," in *Proc. ICCV*, 2005.

[50] C. Lampert, M. Blaschko, and T. Hofmann, "Beyond sliding windows: object localization by efficient subwindow search," in *Proc. CVPR*, 2008.

[51] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 200, no. 1140, pp. 269–294, 1978.

[52] G. Russell, R. Brooks, and T. Binford, "The ACRONYM model-based vision system," in *Proc. IJCAI*, 1979.

[53] D. G. Lowe, "Three-Dimensional Object Recognition from Single Two-Dimensional Images," *Artif. Intell.*, vol. 31, no. 3, pp. 355–395, 1987.

[54] D. W. Jacobs, "GROPER: A grouping-based object recognition system for two-dimensional objects," in *IEEE Workshop on Computer Vision*, 1987.

[55] G. Ettinger, "Large hierarchical object recognition using libraries of parameterized model sub-parts," in *Proc. CVPR*, 1988.

[56] W. E. L. Grimson, "The combinatorics of object recognition in cluttered environments using constrained search," in *Proc. ICCV*, 1988.

[57] C. Goad, "Special purpose automatic programming for 3d model-based vision," in *From Pixels to Predicates*, A. Pentland, Ed., 1986, pp. 371–391.

[58] W. E. L. Grimson and T. Lozano-Pérez, "Localizing Overlapping Parts by Searching the Interpretation Tree," *IEEE Trans. PAMI*, vol. 9, no. 4, pp. 469–482, 1987.

[59] N. Ayache and O. D. Faugeras, "HYPER: a New Approach for the Recognition and Positioning of Two-Dimensional Objects," *IEEE Trans. PAMI*, vol. 8, no. 1, pp. 44–54, 1986.

[60] Joseph L. Mundy and Andrew Zisserman and David A. Forsyth, *Applications of Invariance in Computer Vision*. Springer, 1993.

[61] S. Dickinson, A. Pentland, and A. Rozenfeld, "3-D Shape Recovery Using Distributed Aspect Matching," *IEEE Trans. PAMI*, 1992.

[62] A. Zisserman, J. L. Mundy, D. A. Forsyth, J. Liu, N. Pillow, C. A. Rothwell, and S. Utcke, "Class-based grouping in perspective images," in *IEEE International Conference on Computer Vision*, 1995.

[63] K. Siddiqi and B. Kimia, "Parts of Visual Form: Computational Aspects," *IEEE Trans. PAMI*, vol. 17, pp. 239–251, 1995.

[64] Y. Keselman and S. Dickinson, "Generic model abstraction from examples," *IEEE Trans. PAMI*, vol. 27, pp. 1141–1156, 2001.

[65] S. Ioffe and D. A. Forsyth, "Probabilistic Methods for Finding People," *Int.l Journal of Computer Vision*, vol. 43, no. 1, pp. 45–68, 2001.

[66] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[67] C. Papageorgiou and T.Poggio, "A Trainable System for Object Detection," *Int.l Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.

[68] H. Schneiderman and T. Kanade, "Object Detection Using the Statistics of Parts," *Int.l Journal of Computer Vision*, vol. 56, no. 3, pp. 151–177, 2004.

[69] P. Felzenszwalb and D. Huttenlocher, "Efficient matching of pictorial structures," in *Proc. CVPR*, 2000.

[70] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing visual features for multiclass and multiview object detection," in *Proc. CVPR*, 2004.

[71] A. Opelt, A. Pinz, and A. Zisserman, "Boundary-fragment-model for object detection," in *Proc. CVPR*, 2006.

[72] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool, "Towards multi-view object class detection," in *Proc. CVPR*, 2006.

[73] K. Mikolajczyk, B. Leibe, and B. Schiele, "Multiple object class detection with a generative model," in *Proc. CVPR*, 2006.

[74] N. Razavi, J. Gall, and L. V. Gool, "Scalable multi-class object detection," in *Proc. CVPR*, 2011.

[75] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and pose estimation," in *Proc. ICCV*, 2011.

[76] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3D," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.

[77] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. PAMI*, vol. 32, no. 8, pp. 1362–1376, 2010.

[78] S. Maji and J. Malik, "Object detection using a max-margin Hough transform," in *Proc. CVPR*, 2009.

[79] O. Barinova, V. S. Lempitsky, and P. Kohli, "On detection of multiple object instances using Hough transforms," in *Proc. CVPR*, 2010.

[80] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[81] S. C. Zhu and D. Mumford, "Quest for a Stochastic Grammar of Images," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 4, pp. 259–362, 2007.

[82] F. Han and S.-C. Zhu, "Bottom-up/top-down image parsing with attribute grammar," *IEEE Trans. PAMI*, vol. 31, no. 1, pp. 59–73, 2008, ISSN: 0162-8828.

[83] Z. Xu, H. Chen, S. C. Zhu, and J. Luo, "A Hierarchical Compositional Model for Face Representation and Sketching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 955–969, 2008.

[84] J. Porway and B. Yao and S.C. Zhu, "Learning compositional models for object categories from small sample sets," in *Object Categorization: Computer and Human Vision Perspectives*, Cambridge University Press, 2008.

[85] Y. Zhao and S. C. Zhu, "Image parsing with stochastic scene grammar," in *Proc. NIPS*, 2011, pp. 73–81.

[86] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: learning a hierarchy of parts," in *Proc. CVPR*, 2007.

[87] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille, "Unsupervised structure learning: hierarchical recursive composition, suspicious coincidence and competitive exclusion," in *Proc. ECCV*, 2008.

[88] S. Fidler, M. Boben, and A. Leonardis, "A coarse-to-fine taxonomy of constellations for fast multi-class object detection," in *Proc. ECCV*, 2010.

[89] L. Zhu, Y. Chen, A. Torralba, W. T. Freeman, and A. L. Yuille, "Part and appearance sharing: recursive compositional models for multi-view," in *Proc. CVPR*, 2010, pp. 1919–1926.

[90] S. Todorovic and N. Ahuja, "Extracting subimages of an unknown category from a set of images," in *Proc. CVPR*, 2006.

[91] N. Ahuja and S. Todorovic, "Learning the taxonomy and models of categories present in arbitrary images," in *Proc. ICCV*, 2007.

[92] S. Todorovic and N. Ahuja, "Learning subcategory relevances for category recognition," in *Proc. CVPR*, 2008.

[93] D. Parikh, L. Zitnick, and T. Chen, "Unsupervised learning of hierarchical spatial structures in images," in *Proc. CVPR*, 2009.

[94] P. Ott and M. Everingham, "Shared parts for deformable part-based models," in *Proc. CVPR*, 2011.

[95] Y. Aytar and A. Zisserman, "Tabula rasa: model transfer for object category detection," in *Proc. ICCV*, 2011.

[96] H. O. Song, S. Zickler, T. Althoff, R. B. Girshick, M. Fritz, C. Geyer, P. F. Felzenszwalb, and T. Darrell, "Sparselet models for efficient multiclass object detection," in *Proc. ECCV*, 2012.

[97] R. Girshick, H. O. Song, and T. Darrell, "Discriminatively activated sparselets," in *Proc. ICML*, 2013.

[98] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *Proc. CVPR*, 2013.

[99] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality.," in *SOTC*, 1998.

[100] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. ICCV*, 2011.

[101] G. E. Hinton and S. Osindero, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, p. 2006, 2006.

[102] M. Ranzato, J. Susskind, V. Mnih, and G. Hinton, "On deep generative models with applications to recognition.," in *Proc. CVPR*.

[103] A. Krizhevsky, I. Sutskever, G., and Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. NIPS*, 2012.

[104] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant features through topographic filter maps," in *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR'09)*, IEEE, 2009.

[105] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR'10)*, IEEE, 2010.

[106] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. ECCV*, 2010.

[107] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proc. CVPR*, 2013.

[108] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," 2013.

[109] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. ICML*, 2009.

[110] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *CVPR*, 2011.

[111] R. Socher, B. Huval, B. P. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Proc. NIPS*, 2012.

[112] S. M. A. Eslami, N. Heess, and J. M. Winn, "The shape boltzmann machine: a strong model of object shape," in *Proc. CVPR*, 2012.

[113] Y. Tang, R. Salakhutdinov, and G. Hinton, "Deep lambertian networks," in *ICML*, 2012.

[114] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proc. CVPR*, 2006, pp. 1605–1614.

[115] T. Malisiewicz and A. A. Efros, "Improving spatial support for objects via multiple segmentations," in *BMVC*, 2007.

[116] C. Pantofaru, C. Schmid, and M. Hebert, "Object recognition by integrating multiple image segmentations," in *Proc. ECCV*, 2008.

[117] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," in *Proc. CVPR*, 2009.

[118] O. Russakovsky and A. Y. Ng, "A Steiner tree approach to object detection," in *Proc. CVPR*, 2010.

[119] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *Proc. CVPR*, 2010.

[120] E. Rahtu, J. Kannala, and M. B. Blaschko, "Learning a category-independent object detection cascade," in *Proc. ICCV*, 2011.

[121] I. Endres and D. Hoiem, "Category-independent object proposals," in *Proc. ECCV*, 2010.

[122] J. Carreira and C. Sminchisescu, "Constrained parametric Min-Cuts for automatic object segmentation," in *Proc. CVPR*, 2010.

[123] S. Bagon, O. Boiman, and M. Irani, "What is a good image segment? A unified approach to segment extraction," in *Proc. ECCV*, 2008.

[124] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Proc. CVPR*, 2007.

[125] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. PAMI*, vol. 33, no. 5, pp. 898–916, 2011.

[126] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, "Segmentation as selective search for object recognition," in *Proc. ICCV*, 2011.

[127] J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: a learned mid-level representation for contour and object detection," in *Proc. CVPR*, 2013.

[128] I. Kokkinos and A. Yuille, "Scale invariance without scale selection," in *Proc. CVPR*, 2008.

[129] I. Kokkinos and A. Yuille, "Unsupervised learning of object deformation models," in *Proc. ICCV*, 2007.

[130] I. Kokkinos, "Boundary detection using F-measure, Filter- and Feature- boost," in *Proc. ECCV*, 2010.

[131] S. Tsogkas and I. Kokkinos, "Learning-based symmetry detection in natural images," in *Proc. ECCV*, 2012.

[132] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno, "Dense segmentation-aware descriptors," in *Proc. CVPR*, 2013.

[133] I. Kokkinos and A. Yuille, "HOP: Hierarchical Object Parsing," in *Proc. CVPR*, 2009.

[134] I. Kokkinos and A. Yuille, "Inference and Learning with Hierarchical Shape Models," *Int.l Journal of Computer Vision*, vol. 93, pp. 201–225, 2 2011.

[135] I. Kokkinos, "Bounding part scores for rapid detection with deformable part models," in *2nd Parts and Attributes Workshop, in conjunction with ECCV 2012*, 2012.

[136] M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Proc. CVPR*, 2010.

[137] I. Kokkinos, M. Bronstein, R. Littman, and A. Bronstein, "Intrinsic shape context descriptors for deformable shapes," in *Proc. CVPR*, 2012.

[138] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios, "Parsing Facades with Shape Grammars and Reinforcement Learning," *IEEE Trans. PAMI*, vol. 35, no. 7, pp. 1744–1756, 2013.

[139] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *Proc. CVPR*, 2012.

[140] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, L. Van Gool, and N. Paragios, "Shape grammar parsing via reinforcement learning," in *Proc. CVPR*, 2011.

Chapter 2

Contour Detection in Natural Images

This chapter describes contributions to the extraction of geometric features, such as boundaries and symmetry axes from natural images. We have developed machine learning techniques that deal with challenges particular to boundary [1] and symmetry [2] detection, as well as a linear-fractional programming approach to grouping [3] that simplifies the solution of the minimum cost ratio problem.

These techniques have been evaluated on the Berkeley segmentation benchmark [4], demonstrating consistent improvements. The contour detection evaluations provided in here have also been updated with respect to the earlier publications [1, 3] so as to compare our latest results with the current state-of-the-art [5, 6].

2.1 Learning-based boundary detection

Good low-level features are often a determining factor of success in tasks such as image segmentation, object recognition and correspondence, while front-end failures can propagate to later stages of processing, making it crucial to achieve good performance in these first steps.

Early approaches to boundary detection were originally based on simplified analytic signal models, such as step edges, used in Canny's criteria for edge detection [7, 8] or lines and edges, used in the design of quadrature filter pairs [9–11]. As the developed detectors had been established as optimal, most of the subsequent research effort was devoted to grouping them into larger and more reliable assemblies, using variational [12–14] or statistical [15–17] techniques. While these techniques yielded visible improvements, taking a different route turned out to yield larger rewards; first, the 'compass operator' [18] did away with the linear, convolution-based tradition of boundary detection, and instead detected discontinuities by measuring dissimilarity in a nonlinear, statistical manner, that also facilitated the treatment of diverse cues. And, second, the introduction of human annotated datasets [4, 19] facilitated the formulation of boundary detection as a statistical learning problem and also made it possible to measure progress systematically.

Following these works, consistent improvements in performance on these benchmarks have been achieved [1, 5, 6, 20–25], relying primarily on the design of new features and the application of appropriate machine learning techniques to the problem. In what follows we describe our contributions to learning-based boundary detection including (i) the adaptation of machine learning techniques to address the particularities of the boundary detection task and (ii) the use of additional, efficiently computable features.

Customizing Adaboost for boundary detection

In [1] we build on the Anyboost framework [26] to develop variants of Adaboost [27] that take into account some aspects of training that are particular to boundary detection. This allows us to first develop F-measure boosting to optimize the F-measure, which better matches the evaluation measure used for testing, and also to accommodate a certain amount of labeling ambiguity by relying on Multiple Instance Learning (MIL). In [1] some further ideas were pursued to work with large datasets, such as combining boosting with Monte Carlo to pick a subset of the training examples in the spirit of Filterboost [28], as well as using discriminative dimensionality reduction [29] to compactly encode high-dimensional features. As these turned out to be unnecessary after a careful reimplementation of the training algorithms, we focus on the first two aspects, which remain central to our current approach.

Background - Anyboost: Our starting point is *Anyboost*, which gives us freedom in designing variants of Adaboost by phrasing it as optimization. In particular, given a training set of N input-output pairs, $(X_i, y_i), X_i \in R^d, y_i \in \{-1, 1\}, i = 1, \ldots, N$ we can measure the performance of a classifier $f: R^d \rightarrow R$ in terms of a loss function:

$$C(f) = \sum_{i=1}^{N} c(f(X_i), y_i) \tag{2.1}$$

that penalizes the deviation of the prediction $f(X_i)$ from the label y_i. For instance when using the exponential loss, $c(f(X_i), y_i) = \exp(-y_i f(X_i))$, $C(f)$ provides us with a differentiable upper bound to the number of misclassifications. Adaboost can be understood as a method to minimize Eq. 4.10 with a function of the form:

$$f_T(X) = \sum_{t=1}^{T} a_t h_t(X), \tag{2.2}$$

where the functions $h_t(x)$ belong to a family of simple functions $\mathcal{H}$ ('weak learners') but their combination f_T can result in a complex classifier ('strong learner'). Adaboost is iterative, obtaining $f_t(x)$ from $f_{t-1}(x)$ by only changing the t-th term.

This loss-based formulation of Adaboost is clearly articulated in its Anyboost variant [26]. Considering that the scores of the classifier f_t at round t form an N-dimensional vector $\mathbf{f}_t = [f_t(X_1), \ldots, f_t(X_N)]^T$, the update to $\mathbf{f}_t$ that most rapidly decreases Eq. 4.10 is along the direction:

$$\mathbf{g}_t = \left[-\frac{\partial C}{\partial \mathbf{f}_1}, \ldots, -\frac{\partial C}{\partial \mathbf{f}_N} \right]. \tag{2.3}$$

Since f_{t+1} is obtained by adding to f_t a single $h_t \in \mathcal{H}$, Adaboost resorts to finding the function $h^* \in \mathcal{H}$ that is closest to $\mathbf{g}$ as measured by their inner product:

$$h^* = \operatorname*{argmax}_h \langle \mathbf{g}, \mathbf{h} \rangle = \operatorname*{argmax}_h \sum_i g_i h(X_i), \tag{2.4}$$

where we denote by g_i the i-th element of $\mathbf{g}_t$, and by $\mathbf{h}$ the vector formed by the values of h on the training set. Once h^* is chosen, α_t can be found by minimizing $C(f_{t-1} + \alpha h^*)$. For

$c(f(X_i), y_i) = \exp(-y_i f(X_i))$, these last two steps yield Adaboost [27].

F-measure boosting: Building on Anyboost we can replace the common exponential loss used in Adaboost with *better performance measures* for boundary detection. In particular, due to the 1D nature of boundaries, we have substantially more negatives that positives during both training and testing; this can skew our classifier towards being very negative. The F-measure is appropriate for measuring performance on imbalanced datasets; it is defined as the geometric mean of the classifier's *precision*, p and *recall*, r, which are in turn defined using the counts of true positives (t), false alarms (f) and misses (m), as follows:

$$F = \frac{2pr}{p+r}, \quad \text{where} \quad p = \frac{t}{t+f}, \quad r = \frac{t}{t+m} \tag{2.5}$$

$$t = \sum_{i=1}^{N} [\hat{y}_i = 1][y_i = 1], \;\; f = \sum_{i=1}^{N} [\hat{y}_i = 1][y_i = -1], \;\; m = \sum_{i=1}^{N} [\hat{y}_i = -1][y_i = 1]. \tag{2.6}$$

In Eq. 2.6 $\hat{y}_i$ is the estimated label, y_i is the correct label, and $[\cdot]$ is binary, indicating if $\cdot$ is true. Since $[\hat{y}_i = 1]$ and $[\hat{y}_i = -1]$ are non-differentiable, we use the sigmoidal function to construct a differentiable approximation in terms of the real-valued classifier's output:

$$\sigma_l(f(X)) = \frac{1}{1 + \exp(-l f(X))} \simeq [\hat{y}_i = l] \tag{2.7}$$

Substituting this in Eq. 2.6 yields a differentiable approximation to the F-measure:

$$\tilde{F} = \frac{2\tilde{p}\tilde{r}}{\tilde{p}+\tilde{r}} = \frac{\tilde{t}}{\tilde{t} + \frac{1}{2}(\tilde{f} + \tilde{m})}, \tag{2.8}$$

where we denote by $\tilde{f}, \tilde{m}, \tilde{t}$ the respective approximations to f, m, t. This expression can now be optimized through Anyboost: by setting $C(f) = -\tilde{F}$ and applying the chain rule of differentiation we can compute the quantity $\frac{\partial C}{\partial f(X_i)}$ required to guide the weak learner selection at each step.

Dealing with annotation inconsistencies: By using Anyboost we can also deal with *inconsistencies in human annotations*. As shown in Fig. 2.1, there is a certain amount of inconsistency among the annotations provided by different humans in the Berkeley benchmark regarding both the spatial localization and the orientation of boundaries; the position variability is due to the ambiguity about where the reflection on water begins (blue circle), while the orientation variability is due to the difference in granularity, giving horizontal boundaries for the reflection and vertical boundaries for the whiskers (orangle circle). Since our classifiers use orientation- and position-dependent features as inputs, we need to take this into account during training.

In particular, since the final decision is obtained by maximizing over all candidate orientations, a point will be labeled as positive if it is a boundary along any orientation, and negative otherwise. This matches the setting of Multiple Instance Learning (MIL) [30–32]. Standard, 'single instance' learning assumes training samples come in feature-label pairs. Instead, MIL takes as a training sample a set of features ('bag') and its label. A bag should be labeled positive if at least one of its features is classified as positive, and negative otherwise.

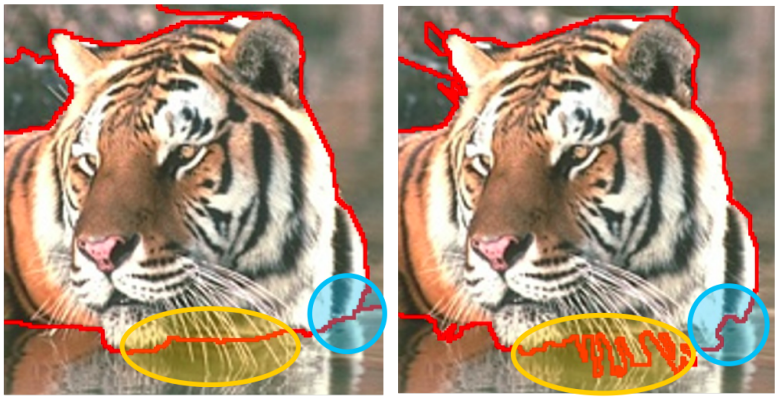


Figure 2.1: Inconsistency of human annotations in the Berkeley segmentation dataset: different humans interpret an image at varying levels of granularity, resulting in annotations that are potentially inconsistent. As shown by the blue circle, the exact position of the boundaries may vary, while as shown by the orange ellipse, pixels indicated as boundaries by two annotators may be associated with different orientations.

The MIL setup allows us to train a classifier in a manner that automatically deals with missing orientation information. For this we extract features at $N = 8$ orientations, obtaining a 'bag' of features $\mathcal{X}_i = \{X_{i,1}, \ldots, X_{i,N}\}$ at each point i. In the same manner we pool multiple spatial positions, to account for the spatial uncertainty, and insert all of these in the feature bag. For each feature $X_{i,j}$ our classifier provides a probability estimate $\pi_{i,j} \doteq P(y_i = 1|X_{i,j}) = \sigma_1(X_{i,j})$; the final decision is taken by maximizing over the individual responses:

$$\pi_i = P(y_i = 1|\mathcal{X}_i) = \max_{j \in [1,\ldots N]} \pi_{i,j} = \max_{j \in [1,\ldots N]} \sigma_1(f(X_{i,j})). \tag{2.9}$$

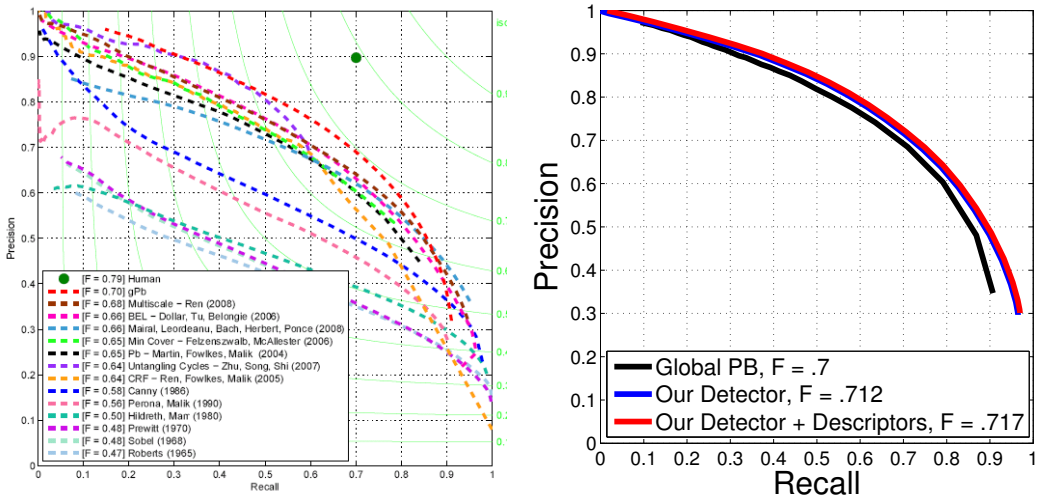
In Adaboost, according to Eq. 2.3, we search for the effect that perturbing any of the classifier's outputs will have in the overall score; the $\max$ operation in Eq. 2.9 is not differentiable, so we can either substitute it with a 'noisy-or' combination [33, 34], or use a subdifferential of π_i instead of the gradient:

$$\partial \pi_i = \frac{d\pi_{i,j^*}}{df(X_{i,j^*})}, \quad \text{where } j^* = \arg\max_j \pi_{i,j}. \tag{2.10}$$

For boundary detection the subdifferential turned out to deliver more precise results, while for the case of symmetry detection, presented in the following section, the noisy-or combination delivered better results. We refer to [35] for a thorough review of noisy-or as well as other MIL variants.

Efficiently computable features

The 'global Pb' detector of [36] is using a very small feature set, which however takes a long time to extract from images. In [3] we consider complementing that feature set with other, easily computable features. We originally considered using SIFT features in [3], computed around a sparse



(a) Pre-2010 boundary detectors vs. global Pb

(b) Our detector vs. global Pb

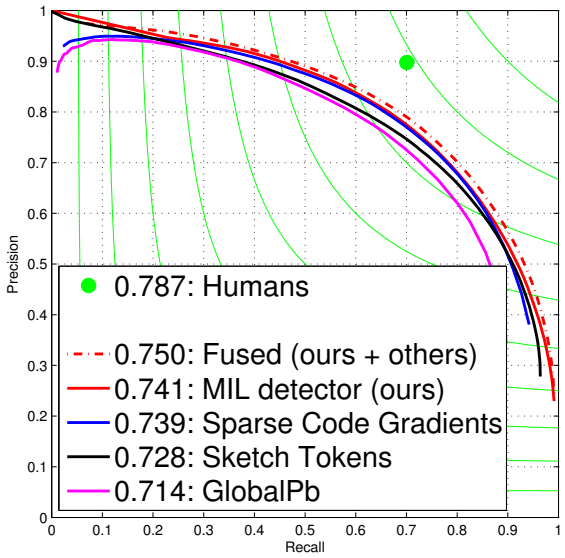
Figure 2.2: Benchmarking results on the BSD300 dataset: the left image compares the performance of boundary detectors developed before 2010 to the global Pb detector [36]. The right image compares our detector [1] to global Pb: our detector's F-measure 0.712, while together with descriptors it increases to 0.717 - the F-measure of the global-Pb detector is .70.

set of locations short-listed by another detector. In [1] we turned to Daisy-like descriptors [37], as their polar sampling pattern allows to efficiently accommodate rotations. We further introduced multi-scale Gaussian, Derivative-of-Gaussian and Laplacian-of-Gaussian features, rapidly computed using recursive (IIR) implementations [7, 38] as well as multi-scale Gabor features, again computed using recursive implementations [39].

Benchmarking Results

We provide results corresponding to two different stages of our boundary detector's development: First, we report the results obtained in [1], where we had demonstrated the merit of (i) using machine learning algorithms that are better adapted to the task at hand and (ii) adding more features. These results, shown in Fig. 2.2 were evaluated on the BSD300 dataset [4], comprising 200 training and 100 test images. We then report our latest results on the BSD500 dataset [36], comprising 300 training and 200 test images and using our current implementation of the ideas laid out above; these results are shown in Fig. 2.3.

To validate our contribution in learning, we first train a classifier using exactly the same features as the global Pb (gPb) detector of [36]; these features include multi-scale color and texture gradients computed by the compass operator, as well as 'spectral gradients', obtained from the directional derivatives of the eigenvectors found by normalized cuts. Both we and [36] use the F-measure for training, so we are optimizing essentially the same cost; the difference lies in the



	ODS	OIS	AP
Fused	0.750	0.771	0.774
MIL	0.741	0.761	0.808
SCG	0.739	0.757	0.772
ST	0.727	0.745	0.780
gPb	0.714	0.736	0.719

Figure 2.3: Left: Precision-Recall curves on the BSD500 dataset, using our latest classifier and the global Pb detector of [36], the Sparse Code Gradients of [5] and the sketch token classifiers of [6]. Our MIL-based detector outperforms the two competing methods - while score-level fusion of the three classifiers results in an F-measure of 0.75. On the right we report the 'Optimal Dataset Scale' and 'Optimal Image Scale' F-measures, and 'Average Precision' metrics used in [36].

training algorithm, which allows us to (i) accommodate uncertainty in the labeling and (ii) recover nonlinear decision rules, by using Adaboost. In [36] a combination of the gPb detector with the Ultrametric Contour Map segmentation algorithm resulted in a improvement of the gPb detector's F-measure from .7 to .71. Our detector achieves an F-measure of .712 without needing any segmentation post-processing. To validate our contribution in feature extraction we repeat the training procedure, but now introducing the new features obtained from the appearance descriptors; these provide an additional boost in performance, increasing it to .717.

We continue with the results presented in Fig. 2.3, which have been obtained from our ongoing research on learning boundary detection; the first substantial difference is that we have replaced the nonlinear classifier obtained from Adaboost with a nonlinear classifier obtained with Random Fourier Expansions [40]; this is simpler to train and also performs better. A second difference is that we now have a preliminary, trainable contour localization module. Breaking up the task into two distinct steps turned out to be very helpful in achieving high recall. Finally, we extract Daisy descriptors on top of the boundary detector's response, implementing classifier stacking. This gave again a small, but substantial boost in performance when compared to using image gradient features. All of these modifications have certain technical novelties -to be presented more thoroughly once consolidated- but in principle are similar to the techniques outlined above in the sense that we (i) train non-linear classifiers and (ii) extract a richer feature set (iii) account for inconsistencies in the labeling. Qualitative results are provided in Fig. 2.4.

Figure 2.4: Sample results from the Berkeley benchmark: we show in the middle the response of our detector, and on the right the ground truth, obtained by averaging the human boundaries.

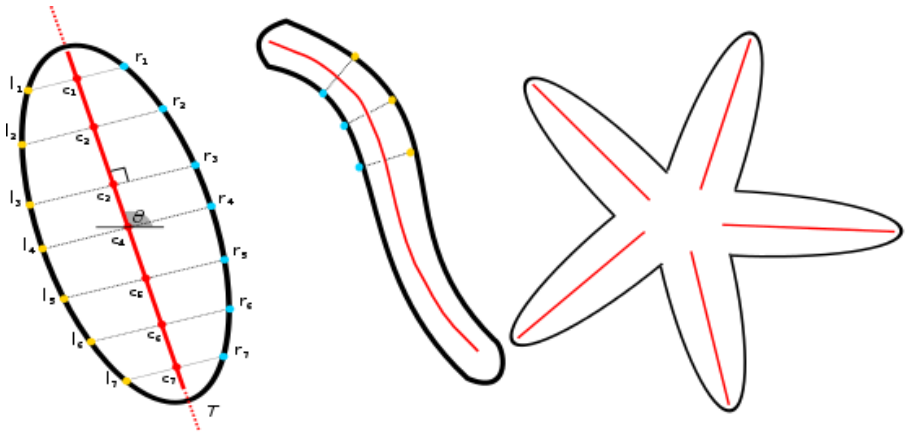


Figure 2.5: We aim at extracting ribbon-like structures that mark local, approximate reflection symmetry; these can be understood as 'generic part' detectors and can be used as inputs to region-based detection algorithms.

2.2 Learning-based symmetry detection

Inspired by the success that machine learning has found in boundary detection, in [2] we applied machine learning to symmetry detection in natural images. Even though symmetry is a rather generic term that can mean different types of structures, we focus on ribbon-like structures, namely contours having local and approximate reflection symmetry, as illustrated in Fig. 2.5. Such structures can serve object recognition by providing proposals for object part locations [41, 42], while potentially building a bridge between region- and contour- based approaches: symmetry axes correspond to regions, while also being organized along 1D structures. Symmetry axes have also been considered in a 'transfer learning' setting for object classes [43], while they naturally fit with the recent quest for 'objectness' and generic region proposals [44–46].

Symmetry detection is commonly seen as a process that follows segmentation [47–51] but in our work we are interested in directly detecting symmetry from natural images. Several works have studied the latter problem [52–56] under the names of *g*rayscale symmetry-, skeleton-, ridge-valley-, crease-, or ribbon- detection, but they all use hand-crafted criteria to detect symmetries. Instead, we use human annotations to learn how to detect symmetry axes.

Our contributions lie in three directions: first, annotating the Berkeley segmentation dataset with symmetry axis information; second, using statistical features at multiple scales and orientations; and third using multiple-instance learning to deal with the inherently multi-scale nature of symmetry. We detail these three points in the following.

Dataset Construction

We build on the Berkeley Segmentation Dataset (BSD300) [4] and use a combination of automated symmetry detection [57] with human post-processing, which is necessary for two reasons.

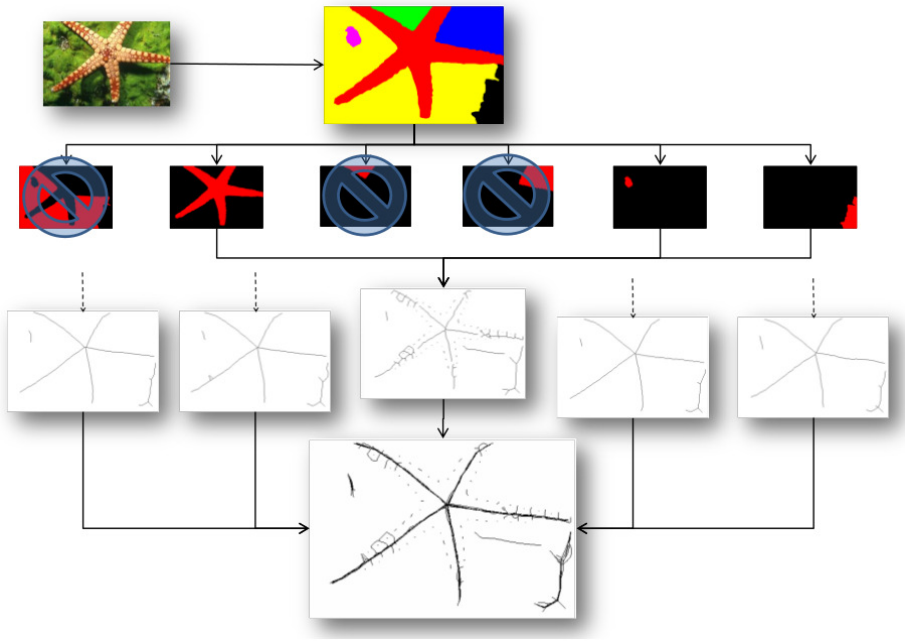


Figure 2.6: Construction of an image skeleton: the user examines individually the segments in the ground truth of [4] and rejects the ones that deliver inappropriate skeletons. This is repeated for every segmentation and the union of the resulting skeletons forms the symmetry ground-truth.

First, background segments are occluded by the foreground regions, which severely distort their shape; the resulting symmetry axes often do not correspond to either semantically meaningful parts or symmetric regions. Second, skeleton extraction often produces spurious branches.

We use the procedure illustrated in Fig. 2.6 to control the ground truth construction: given a segmentation of the input image, the user examines each segment separately and decides whether its skeleton should be included in the final image skeleton map. We apply this procedure separately to each of the 5-8 segmentations provided per image on BSDS300 and aggregate the validated skeleton maps by taking their union. In Fig. 2.6 we show the partial and final skeletons obtained from the segmentations of an image.

Feature Extraction

For feature extraction we develop statistical features, which can be understood as adaptations of the compass operator [4, 18], to symmetry detection. The compass operator measures the cue variation across two image regions i, j by comparing their respective cue histograms, R_i, R_j. For instance using the χ^2-distance gives us a scalar quantity:

$$H_{i,j} = \frac{1}{2} \sum_k \frac{(R_i(k) - R_j(k))^2}{R_i(k) + R_j(k)}, \tag{2.11}$$

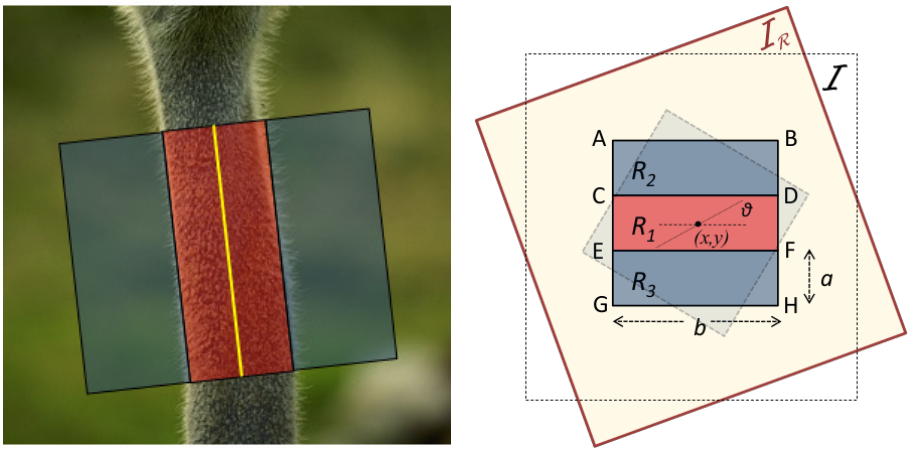
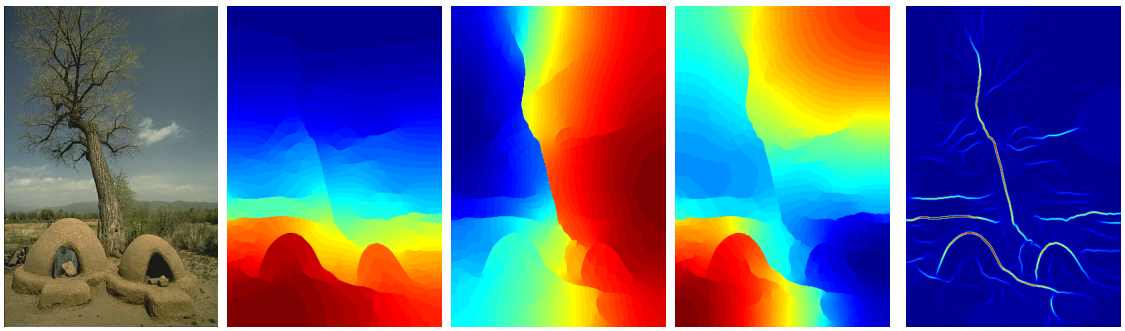


Figure 2.7: Left: The feature contents of the middle part are not similar to either the contents of the left or the right parts. Therefore the strength of the symmetry in the vertical axis (in yellow) is high. Right: Rectangles used to rapidly calculate sums with integral images. Using rotated filters on the original image I is equivalent to using axis-aligned filters on a rotated image, I_R.

that measures region dissimilarity. By virtue of being statistical, it can deal with a broader set of discontinuities other than step edge-like transitions, for instance having different bi- or even multi-modal distributions on each side. Two adaptations are required to apply this operator to our case: first, we need to redefine the regions over which the statistics are being computed, and second, we need to deal with the inherent multi-scale nature of symmetry detection.

Regarding the first adaptation we consider *three* adjacent rectangles as shown in Fig. 2.7, instead of two; the middle rectangle corresponds to the symmetric, coherent part, and the other two correspond to its lobes, which are supposed to be dissimilar to the center. The intuition is that if a symmetric structure is present, at least at a given combination of scale and orientations the statistics of the middle rectangle should be different from those of its surrounding rectangles. Instead of hand-coding such a rule, we extract the associated features and leave it to the learning algorithm to figure out how to combine them. In particular for every pair of rectangles we measure their region-based dissimilarity $H_{i,j}(x, y, \theta, s)$, where $i, j \in \{1, 2, 3\}$ indicate which two rectangles are being compared, while x, y, θ, s determine the shape of the rectangles: θ denotes orientation, s denotes scale, and (x, y) the center of the middle rectangle.

For the second adaptation, we need to consider more neighborhoods and also use larger scales. To alleviate the resulting computational burden we first create a Gaussian pyramid of the original image before feature extraction, and also replace the filtering operations with recursive filters; in particular for axis-aligned rectangles these can be implemented with integral images [58], while instead of using rotated rectangles we use rotated versions of the original image again in conjunction with integral images, as in [59]. This is repeated for the brightness, color (LAB), and texture features (textons - [4]). With thirteen scales and eight orientations per scale, the overall feature extraction and scoring procedure currently takes about 5 seconds in Matlab for a 321×481 image.



(a) Input (b) First eigenvectors of symmetry-based Normalized Cut relaxation (c) Spectral symmetry

Figure 2.8: Spectral symmetry feature: using as input image (a) and the output of a 'first stage' symmetry detector we can setup a generalized eigenvector problem, whose first three eigenvectors are shown in (b). The resulting spectral symmetry feature is shown in (c).

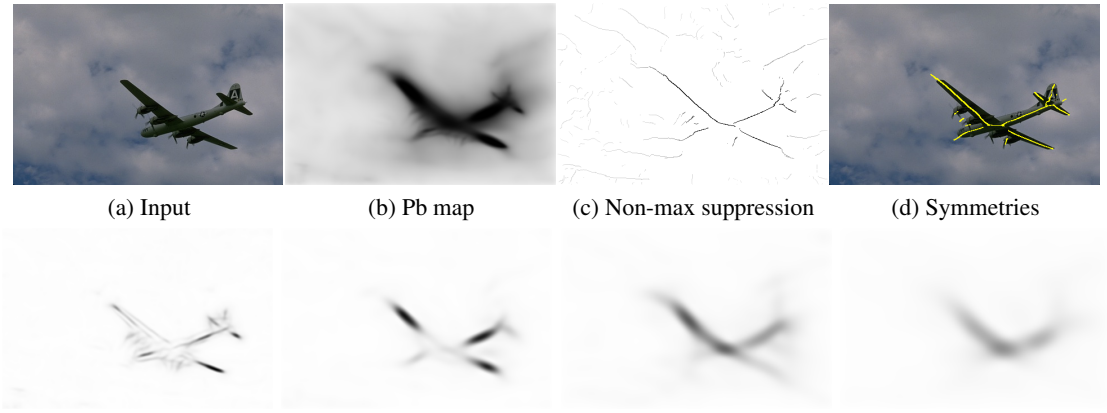
As shown in Fig. 2.8 we can also construct a 'spectral symmetry' feature to elicit a 'global' measure of coherence in the image. For this we use relaxation to the Normalized Cut problem [60] used in [36], but replace their first-stage boundary detector with our own first-stage symmetry detector. This is the computationally most demanding cue, requiring computation in the order of a minute per image, so we treat this as optional, to be used when time is not important.

Since our features are scale- and orientation- dependent, training our detector in the standard supervised setting would require choosing a scale and orientation combination for every symmetry point. Instead, we leave this decision to MIL, as we also did for boundary detection, using features at 8 orientations and 13 scales for each pixel's bag. We show the results of the trained symmetry detector in the top row of Fig. 2.9. We estimate the probability-of-symmetry at every pixel and every scale and orientation combination, resulting in a 4D symmetry map. These probabilities are combined with the noisy-or rule into an aggregate symmetry response. As this can result in a diffuse response, a non-maximum suppression step is used for thinning prior to thresholding.

In the bottom row of Fig. 2.9 we show how the response map is computed, by showing the per-scale results, obtained by taking the noisy-or of the orientations at a fixed scale. We note that our detector has learned to automatically pick the dominant local image scale, which could potentially be useful for tasks such as image segmentation, or to provide generic part proposals that are supported by regions.

Results

We evaluate our detector using our symmetry axis ground truth for BSDS300. We first assess the hardness of the task at hand by measuring human performance on it; this can be computed by picking the binary map delivered by a single user, matching it with the binary maps of the other users, and then going in turns. An F-measure for humans can thereby be obtained, which can be understood as an upper bound of what we can expect to achieve with a computer. The estimated



(a) Input (b) Pb map (c) Non-max suppression (d) Symmetries

Probability responses at increasing scales

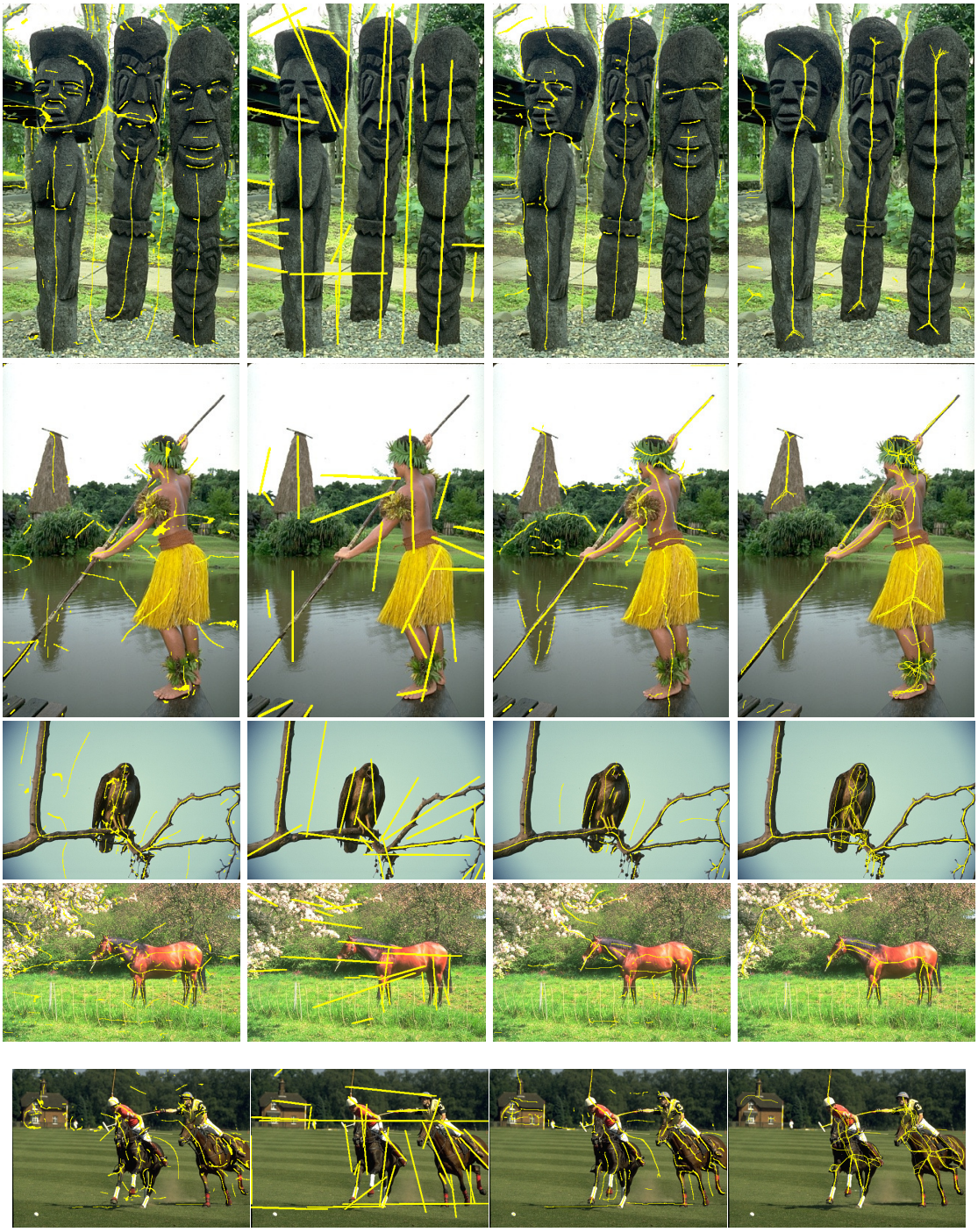
Figure 2.9: Top: processing steps from the initial image to the final symmetry axis map. Bottom: probability responses before non-maximum suppression at increasing scales from left to right.

score was $F = 0.73$ and the corresponding iso-curve is shown in Fig. 2.11. This is substantially below the F-measure estimated for humans on boundary detection –$F = 0.78$– which indicates the difficulty of symmetry detection.

In Fig. 2.11 we show the precision-recall curves of our method; in order to confirm the importance of each separate feature used to train our detector, we compare with the results of detectors trained with only a subset of the available features. Our main observation is that the largest part of the performance improvement is due to the multi-cue combination, as illustrated by the precision-recall curves of the different cues in Fig. 2.11: when using only Brighntess gradients the performance is very similar to Linderberg's ridge detector, implemented as in [41]. But since our learning-based framework allows us to combine different cues in a data-driven way, adding more cues consistently improves performance, and we attain a uniformly better performance throughout the precision-recall spectrum.

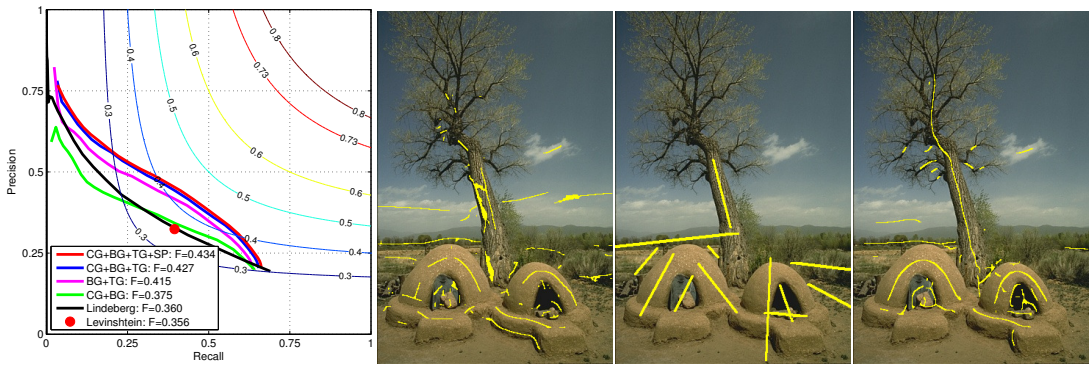
We also compare our algorithm to the more recent, learning-based approach of Levinshtein et al. [56]; since their algorithm returns binary results we have a single point corresponding to the precision and recall of their detector, amounting to an F-measure of 0.355. We note that they solve a slightly different problem (delivering line segments for symmetry), and they do not use our dataset for training, so the comparison is somehow skewed in favor of our method; still there is a quite substantial difference, as also suggested by the qualitative results shown in Fig. 2.10. Our technique also delivers continuous contours, which can be subsequently broken and re-grouped at will; as such, we do not make "early commitments" from which it may be hard to recover post-hoc.

Our dataset and implementation is publicly available at [61]. We are currently exploring ways to integrate these cues in subsequent tasks, such as image segmentation [45], 'objectness' estimation [44] and region/symmetry-based object recognition with deformable models [41, 43].



(a) Lindeberg [41, 52] (b) Levinshtein [56] (c) Our work [2] (d) Ground truth

Figure 2.10: Qualitative results for all three compared methods and respective ground-truth.



(a) Precision-recall curves (b) Lindeberg [41, 52] (c) Levinshtein et al [56] (d) Our work [2]

Figure 2.11: Quantitative and qualitative comparison of our detector with other methods. In (a) we plot the performance of our detector trained with different feature combinations to illustrate the boost due to color and spectral features. The ground truth F-measure is represented by the red iso-curve. For (b) we use the implementation of [41] and for (c) the available code of [56]. .

2.3 Contour grouping with linear-fractional programming

Despite any advances in contour detection, detector failures due to occlusions, poor illumination, or strong shading are inevitable. This calls for a contour grouping stage that will provide higher-level modules with long and informative structures by linking fragmented contours. Grouping has also been argued [62, 63] and empirically demonstrated in [20, 64, 65] to eliminate isolated edge fragments, thereby boosting contour detection performance in the high-precision regime.

Contour grouping is commonly phrased in terms of optimizing a length-normalized saliency measure [64–66]. This ensures scale-invariance, cancelling the short contour bias of un-normalized criteria, e.g. [12], but the resulting optimization problem becomes harder.

Our contribution consists in phrasing this optimization task as a fractional programming problem, which in turn can be solved using linear programming [67]. Our approach has two positive aspects: first, we have the same flexibility as [65], namely we can solve both open and closed contour grouping. Second, we use standard linear programming, resulting in substantially simpler implementations than those proposed e.g. in [65]. Our implementation is available at [68].

Cost Ratio Optimization

We use a graph-based formulation for grouping, which views contours as cycles in a graph. Following [69] we use a bidirected graph, representing each line segment with two nodes -one for each direction; we will be referring to such nodes as *conjugate*. Closed contours amount to cycles of this graph, i.e. paths that begin and end at the same node. Furthermore, introducing connections between conjugate nodes [65] allows us to also associate open contours with graph cycles by using two ‘u-turn’ connections at the curve’s endpoints, as shown in Fig. 2.12 on the right.

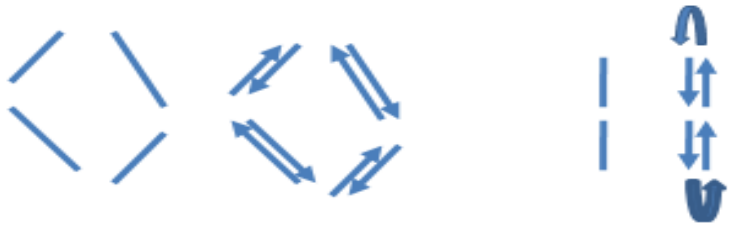


Figure 2.12: Graph construction following [69] and [65]: Left: A bidirected graph is constructed by separately treating edges with different directions. Right: Connections among conjugate nodes help detect open contours as cycles in a graph.

Turning to finding good contour groups, an idea that has been broadly used in the grouping literature is to normalize any contour-dependent energy by the contour's length, so as to ensure scale-invariance; this has first been introduced in the Minimum Ratio Weight Contours of [66] and extended in the works of [64, 70, 71]. In the continuous setting this amounts to scoring a candidate curve Γ in terms of the ratio of two curvilinear integrals:

$$C(\Gamma) = \frac{\int_\Gamma \left(E(s) + a\kappa^2(s)\right) ds + c_T}{\int_\Gamma 1 ds} = \frac{\mathcal{W}(\Gamma)}{\mathcal{L}(\Gamma)}, \tag{2.12}$$

where κ is the curve's curvature, $E(s) = -\log P_B(s)$ is a local term formed in terms of the boundary strength and c_T is an, optional, termination penalty that favors longer curves. We thus have an Elastica-type [13] smoothness penalty combined with a local boundary strength to measure the quality of a curve, and a length normalization term which ensures the score will remain invariant to image scaling, modulo the complexity term, c_T. Using the rightmost fraction, we will refer to $W(\Gamma)$ as the 'weight' of the curve and to $L(\Gamma)$ as its length.

Mapping this continuous setting to the discrete, graph-based setting is detailed in [3], but is mostly straightforward, involving the substitution of integrals in Eq. 2.12 by line-based approximations. This allows us to write the numerator and denominator of the grouping criterion in Eq. 2.12 for any cyclic path traversing nodes $(e_1, \ldots, e_C, e_1)$ as:

$$\mathcal{W}(\Gamma) \simeq \sum_{i=1}^{C} w_{e_i,e_{i+1}} + c_T, \;\; \mathcal{L}(\Gamma) \simeq \sum_{i=1}^{C} l_{e_i,e_{i+1}} \tag{2.13}$$

which in turn allows us to formulate our problem as a minimum cost ratio cycle detection problem, i.e. finding a graph cycle that has the minimum cost, when divided by the length of the path.

Fractional-Linear Programming Grouping

Finding the minimum cost ratio cycle on a graph is a well-studied combinatorial optimization problem that can be addressed with a variety of discrete optimization techniques, covered in [66]. However these solutions require implementing nontrivial algorithms such as finding zero-weight

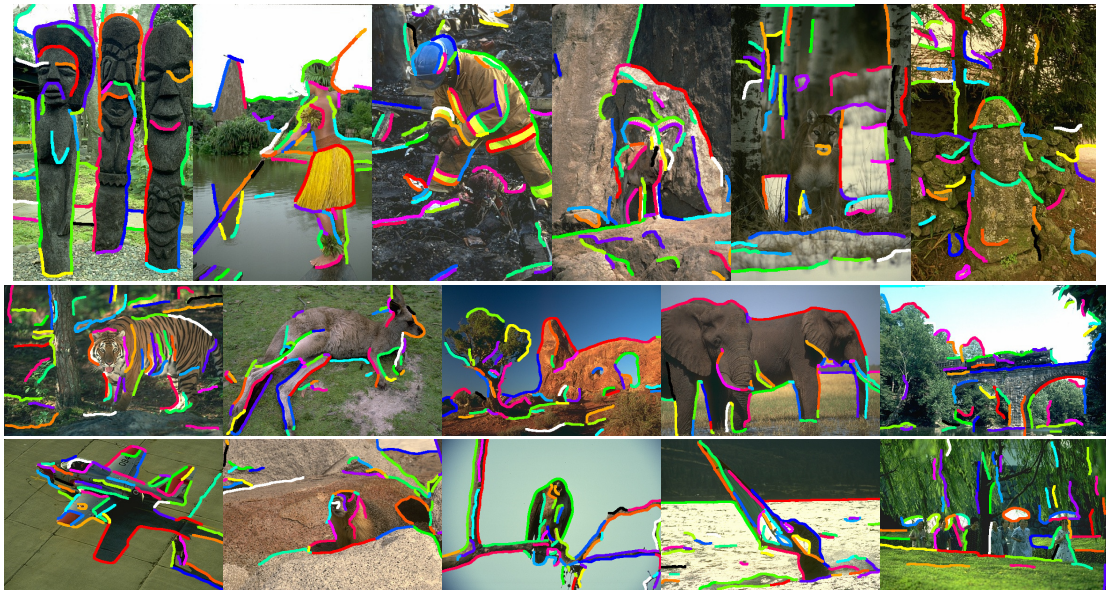


Figure 2.13: The first (strongest) 50 groupings found by our algorithm. Please see in color.

cycles [66], maintaining priority queues and hashing schemes [64], approximately finding circular paths with largest area [65], and using linear [66] or binary search [71] for the optimal cost value, and solving a combinatorial problem for each such value. This makes them hard for non-experts -at least the author- and impedes their broader adoption. Here we propose an approach based on fractional-linear programming, whose main advantage is being substantially simpler to implement and understand: its implementation amounts to setting up a linear program, which can be solved using standard libraries such as MOSEK.

Since in Eq. 2.13 we express the numerator and denominator of our cost criterion in terms of the weights/lengths of graph connections, we formulate our optimization problem in terms of variables that indicate whether a graph connection is used. We therefore now switch to using indexes for connections, while if a connection between nodes i, j is indexed by k we will denote the 'conjugate' connection between j', i' as k'. We use v_k to indicate whether the k-th connection is used, and relax the constraint $v_k \in \{0, 1\}$ to $v_k \in [0, 1]$. We introduce the node-connection adjacency matrix A, whose entry $A_{i,k}$ is $+1$ when connection k departs from node i and -1 when it arrives. Finally, depending on the kind of group we want to extract (closed/open contour) we introduce appropriate constraints to ensure that the solution delivered by the linear program will correspond respectively to a closed contour (no pair of conjugate nodes is used) or an open contour (only conjugate nodes are used).

Putting these together, our optimization problem becomes:

$$\min \quad \frac{\sum_k v_k w_k + c_T}{\sum_j v_k l_k} \tag{2.14}$$

$$\text{s.t.} \quad v_k \geq 0, \;\; v_k \leq 1, \;\; \forall k \tag{2.15}$$

$$\sum_k v_k A_{i,k} = 0, \forall i \tag{2.16}$$

$$\text{Open Curves :} \quad v_k - v_{k'} = 0 \tag{2.17}$$

$$\sum_{k \in K'} v_k = 2 \tag{2.18}$$

$$\text{Closed Curves :} \quad v_k + v_{k'} \leq 1, \tag{2.19}$$

$$\sum_{k \in K'} v_k = 0 \tag{2.20}$$

The optimized quantity in Eq. 2.14 is the ratio between the aggregated cost of the utilized connections and their length, in direct analogy to Eq. 2.12. Constraint Eq. 5.4 guarantees that the connections will form a cycle. The open-curve constraint Eq. 2.17 assures that for each used connection k its conjugate connection k' will also be used, ensuring that an open curve will travel back in the same way that it went from one endpoint to another. On the flip side, the closed curve constraint Eq. 2.19 allows the use of only one of the conjugate connections. The set K' appearing in Eq. 2.18 and Eq. 2.20 consists of the connections among conjugate nodes, i.e. points where a curve turns around. Eq. 2.18 thus constrains each open curve to have two such nodes, one in its middle and another in the end, while Eq. 2.20 prohibits the use of such nodes for closed curves.

Having described our optimization problem we can turn to solving it. As detailed in [67], an optimization problem of this form can be converted into a linear program. In specific, for $y = \frac{x}{e^T x + f}$ and $z = \frac{1}{e^T x + f}$, the following two optimization problems are equivalent:

$$P1: \quad \begin{aligned} &\min \quad \frac{c^T x + d}{e^T x + f} \\ &Gx \prec 0 \\ &Ax = b \end{aligned} \qquad\qquad \begin{aligned} &\min \quad c^T y + dz \\ &Gy - hz \prec 0, \quad z \succ 0 \\ &Ay - bz = 0, \quad e^T y + fz = 1 \end{aligned}$$

Having built the matrices for the original fractional programming problem, solving the equivalent problem can be done efficiently using any sparse LP library. We detect contours greedily, by iteratively solving the fractional-linear program above, and for every edge that participates in a grouping all of its connection weights are set to infinity at the next iteration. Fractional solutions can occur when the flow along a patch splits into two halves at a certain edge and merges later on at a subsequent edge. Whenever this happens, we temporarily remove all connections including such edges and solve the optimization problem again. On average, the optimization takes a fraction of a second per contour and approximately 20 seconds for an image containing 200-300 hundred contours. The 50 strongest groupings per image are shown in Fig. 2.13 for several images of the BSD300 dataset. In [3] it was demonstrated that this improved the performance of a baseline detector; it remains to be seen whether this is also the case for the detectors described previously.

Bibliography

[1] I. Kokkinos, "Boundary detection using F-measure, Filter- and Feature- boost," in *Proc. ECCV*, 2010.

[2] S. Tsogkas and I. Kokkinos, "Learning-based symmetry detection in natural images," in *Proc. ECCV*, 2012.

[3] I. Kokkinos, "Highly accurate boundary detection and grouping," in *Proc. CVPR*, 2010.

[4] D. Martin, C. Fowlkes, and J. Malik, "Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues," *IEEE Trans. PAMI*, vol. 26, no. 5, pp. 530–549, 2004.

[5] X. Ren and L. Bo, "Discriminatively trained sparse code gradients for contour detection," in *Proc. NIPS*, 2012.

[6] J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: a learned mid-level representation for contour and object detection," in *Proc. CVPR*, 2013.

[7] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. PAMI*, vol. 8, no. 6, pp. 679–698, 1986.

[8] R. Deriche, "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector," *Int.l Journal of Computer Vision*, vol. 1, no. 2, pp. 167–187, 1987.

[9] C. Morrone and D. Burr, "Feature Detection in Human Vision: a Phase-Dependent Energy Model," *Proceedings of the Royal Society of London B*, vol. 235, pp. 221–245, 1988.

[10] P. Perona and J. Malik, "Detecting and Localizing Edges Composed of Steps, Peaks and Roofs," in *Proc. ICCV*, 1990.

[11] I. Kokkinos, G. Evangelopoulos, and P. Maragos, "Texture Analysis and Segmentation Using Modulation Features, Generative Models and Weighted Curve Evolution," *IEEE Trans. PAMI*, vol. 31, no. 1, pp. 142–157, 2009.

[12] V Caselles, R Kimmel, and G. Sapiro, "Geodesic Active Contours," *Int.l Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[13] D. Mumford, "Elastica and Computer Vision," in *Algebraic Geometry and its applications*, J. Bajaj, Ed., Springer Verlag, 1993, pp. 507–518.

[14] E. Sharon, A. Brandt, and R. Basri, "Completion energies and scale," *IEEE Trans. PAMI*, vol. 22, pp. 1117–1131, 2000.

[15] L. Williams and D. Jacobs, "Stochastic Completion Fields: A Neural Model of Illusory Contour Shape and Salience," *Neural Computation*, vol. 9, no. 4, pp. 837–858, 1997.

[16] J. August and S. Zucker, "Generative Model of Curve Images with a Completely Characterized Non-Gaussian Joint Distribution," in *Workshop on Statistical and Computational Theories of Vision at ICCV*, 2001.

[17] A. Desolneux, L. Moisan, and J.-M. Morel, "Meaningful alignments," *Int.l Journal of Computer Vision*, vol. 40, no. 1, pp. 7–23, 2000.

[18] M. Ruzon and C. Tomasi, "Color edge detection with the compass operator," in *Proc. CVPR*, 1999.

[19] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu, "Statistical edge detection: learning and evaluating edge cues," *IEEE Trans. PAMI*, vol. 25, no. 1, 2003.

[20] X. Ren, C. Fowlkes, and J. Malik, "Scale-invariant contour completion using conditional random fields," in *Proc. ICCV*, 2005.

[21] P. Dollar, Z. Tu, and S. Belongie, "Supervised Learning of Edges and Object Boundaries," in *Proc. CVPR*, 2006.

[22] P. Arbelaez, "Boundary Extraction in Natural Images Using Ultrametric Contour Maps," in *WPOCV*, 2006.

[23] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik., "Using contours to detect and localize junctions in natural images," in *Proc. CVPR*, 2008.

[24] X. Ren, "Multiscale helps boundary detection," in *Proc. ECCV*, 2008.

[25] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce, "Discriminative sparse image models for class-specific edge detection and image interpretation," in *Proc. ECCV*, 2008.

[26] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent in function space," in *Proc. NIPS*, 2000.

[27] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm.," in *ICML*, 1996, pp. 148–156.

[28] J. K. Bradley and R. E. Schapire, "FilterBoost: Regression and Classification on Large Datasets.," in *Proc. NIPS*, 2007.

[29] D. Cook and H. Lee, "Dimension reduction in binary response regression," *J. Am.n St.l Ass.n*, vol. 94, 1999.

[30] T. G. Dietterich, R. H. Lathrop, and T. Lozano-perez, "Solving the Multiple-Instance Problem with Axis-Parallel Rectangles," *Artificial Intelligence*, vol. 89, pp. 31–71, 1997.

[31] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proc. NIPS*, 2002.

[32] P. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting and object detection," in *Proc. NIPS*, 2006.

[33] S. Ray and M. Craven, "Supervised versus multiple instance learning: an empirical comparison," in *ICML*, 2005.

[34] P. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting and object detection," in *Proc. NIPS*, 2006.

[35] B. Babenko, P. Dollár, Z. Tu, S. Belongie, *et al.*, "Simultaneous learning and alignment: multi-instance and multi-pose learning," 2008.

[36] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. PAMI*, vol. 33, no. 5, pp. 898–916, 2011.

[37] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *Proc. CVPR*, 2008.

[38] R. Deriche, "Recursively implementing the Gaussian and its derivatives," INRIA, Unite de Recherche Sophia-Antipolis, Tech. Rep. 1893, 1993.

[39] A. Bernardino and J. Santos-Victor, "Fast iir isotropic 2-d complex gabor filters with boundary initialization," *IEEE Trans. Im. Proc.*, vol. 15, no. 11, pp. 3338–3348, 2006.

[40] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. NIPS*, 2007.

[41] I. Kokkinos, P. Maragos, and A. Yuille, "Bottom-up and top-down object detection using primal sketch features and graphical models," in *Proc. CVPR*, 2006.

[42] I. Kokkinos and A. Yuille, "Inference and Learning with Hierarchical Shape Models," *Int.l Journal of Computer Vision*, vol. 93, pp. 201–225, 2 2011.

[43] M. Stark, M. Goesele, and B. Schiele, "A shape-based object class model for knowledge transfer," in *Proc. ICCV*, 2009.

[44] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," in *Proc. CVPR*, 2010.

[45] J. Carreira and C. Sminchisescu, "Constrained parametric Min-Cuts for automatic object segmentation," in *Proc. CVPR*, 2010.

[46] I. Endres and D. Hoiem, "Category-independent object proposals," in *Proc. ECCV*, 2010.

[47] H. Blum *et al.*, "A transformation for extracting new descriptors of shape," *Models for the perception of speech and visual form*, 1967.

[48] M. Brady and H. Asada, "Smoothed local symmetries and their implementation," *Int.l Journal of Robotics Research*, vol. 3, no. 3, pp. 36–61, 1984.

[49] S. C. Zhu and A. Yuille, "FORMS: A Flexible Object Recognition and Modeling System," *Int.l Journal of Computer Vision*, vol. 20, no. 3, 1996.

[50] M. van Eede, D. Macrini, A. Telea, C. Sminchisescu, and S. Dickinson, "Canonical skeletons for shape matching," *ICPR*, 2006.

[51] M. Demirci, A. Shokoufandeh, and S. Dickinson, "Skeletal Shape Abstraction from Examples," *IEEE Trans. PAMI*, vol. 31, no. 5, pp. 944–952, 2009.

[52] T Lindeberg, "Edge Detection and Ridge Detection with Automatic Scale Selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 116–132, 1998.

[53] S. Pizer, C. Burbeck, J. Coggins, D. Fritsch, and B. Morse, "Object Shape Before Boundary Shape: Scale-Space Medial Axes," *Journal of Mathematical Imaging and Vision*, vol. 4, pp. 303–313, 1994.

[54] K. Siddiqi and S. Pizer, *Medial Representations*. Springer, 2009.

[55] A. Lôpez, F. Lumbreras, J. Serrat, and J. Villanueva, "Evaluation of methods for ridge and valley detection," *IEEE Trans. PAMI*, vol. 21, no. 4, pp. 373–335, 1999.

[56] A. Levinshtein, S. Dickinson, and C. Sminchisescu, "Multiscale symmetric part detection and grouping," *Proc. ICCV*, 2009.

[57] A. Telea and J. Van Wijk, "An augmented fast marching method for computing skeletons and centerlines," *Eurographics 2002*, 2002.

[58] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[59] B. Catanzaro, B. Su, N. Sundaram, Y. Lee, M. Murphy, and K. Keutzer, "Efficient, high-quality image contour detection," *Proc. ICCV*, 2009.

[60] J. Shi and J. Malik, "Normalized cuts and image segmentation," *T.PAMI*, 1997.

[61] S. Tsogkas, *Learning-based symmetry detection*, `http://www.centrale-ponts.fr/personnel/tsogkas/code.html`, 2012.

[62] A. Shaashua and S. Ullman, "Structural Saliency: the Detection of Globally Salient Structures Using a Locally Connected Network," in *Proc. ICCV*, 1988, pp. 321–327.

[63] T. D. Alter and R. Basri, "Extracting salient curves from images: an analysis of the saliency network.," in *Proc. CVPR*, 1996.

[64] P. Felzenszwalb and D McAllester, "A Min-Cover Approach for Finding Salient Curves," in *POCV*, 2006.

[65] Q. Zhu, G. Song, and J. Shi, "Untangling cycles for contour grouping," in *Proc. ICCV*, 2007.

[66] I. Jermyn and H. Ishikawa, "Globally optimal regions and boundaries as minimum ratio weight cycles.," *IEEE Trans. PAMI*, vol. 23, pp. 1075–1088, 2001.

[67] S Boyd and R Vandeberghe, *Convex Optimization*. Cambridge University Press, 2004.

[68] I. Kokkinos, *Fractional programming grouping*, `http://vision.mas.ecp.fr/Personnel/iasonas/contours.html`, 2013.

[69] S. Mahamud, L. Williams, K. Thornber, and K. Xu., "Segmentation of multiple salient closed contours from real images.," *IEEE Trans. PAMI*, vol. 25, pp. 433–444, 2003.

[70] S. Wang, T. Kubota, J. M. Siskind, and J. Wang, "Salient Closed Boundary Extraction with Ratio Contour.," *IEEE Trans. PAMI*, vol. 27, no. 4, pp. 546–561, 2005.

[71] T. Schoenemann, S. Masnou, and D. Cremers, "The Elastic Ratio: Introducing Curvature into Ratio-based Globally Optimal Image Segmentation," *IEEE Trans. Im. Proc.*, vol. 20, no. 9, pp. 2565–2581, 2011.

Chapter 3

Invariant Image and Surface Descriptors

This chapter describes descriptors that achieve scale and rotation invariance without relying on scale or rotation estimation - this allows us to compute such descriptors *densely*. We first adapt the Fourier Transform Modulus technique [1, 2] to construct *dense* Scale-Invariant Descriptors (SIDs) for images [3, 4]. We then combine scale- and rotation- invariance with robustness to occlusions and background changes by incorporating segmentation information in the construction of 'segmentation-aware' SID and SIFT descriptors [5].

For surfaces in [6] we use the same idea to construct a scale-invariant extension of Heat Kernel Signatures [7]. Our method constructs signatures that are invariant both to isometries, namely deformations that preserve geodesic distances, and to scale changes. In [8] we employ a geometry-based surface charting technique to generalize shape-context [9] to surfaces, and thereby construct meta-descriptors with increased discriminative ability.

These descriptors are validated in point matching, wide- baseline stereo and large-displacement optical flow for images, and surface retrieval and matching for surfaces. Their implementation is available from [10].

3.1 Dense scale-invariant image descriptors

A common problem that emerges in the computation of local descriptors is the variability of the signal scale. The standard approach to cope with this is to use scale selection [11, 12], which consists in estimating a characteristic scale around the few image or shape points where scale estimation can be performed reliably. However, it is often desirable to have a scale-invariant descriptor that can be constructed densely. One such scenario is when using a group of contour points for recognition - ideally we should be able to estimate scale invariant features 'on-demand', at any feature point. One partial remedy for edges was proposed in [13], but this still cannot be guaranteed to work at any image point. This may be necessary for instance when establishing dense image correspondences in the presence of scale changes.

We will argue that one can instead adapt the Fourier Transform Modulus-based image registration technique [1, 2, 14] to the construction of descriptors and thereby guarantee scale- and rotation- invariance at any image point. We start with a brief illustration of the technique for a one-dimensional signal and then present how it applies to image descriptors.

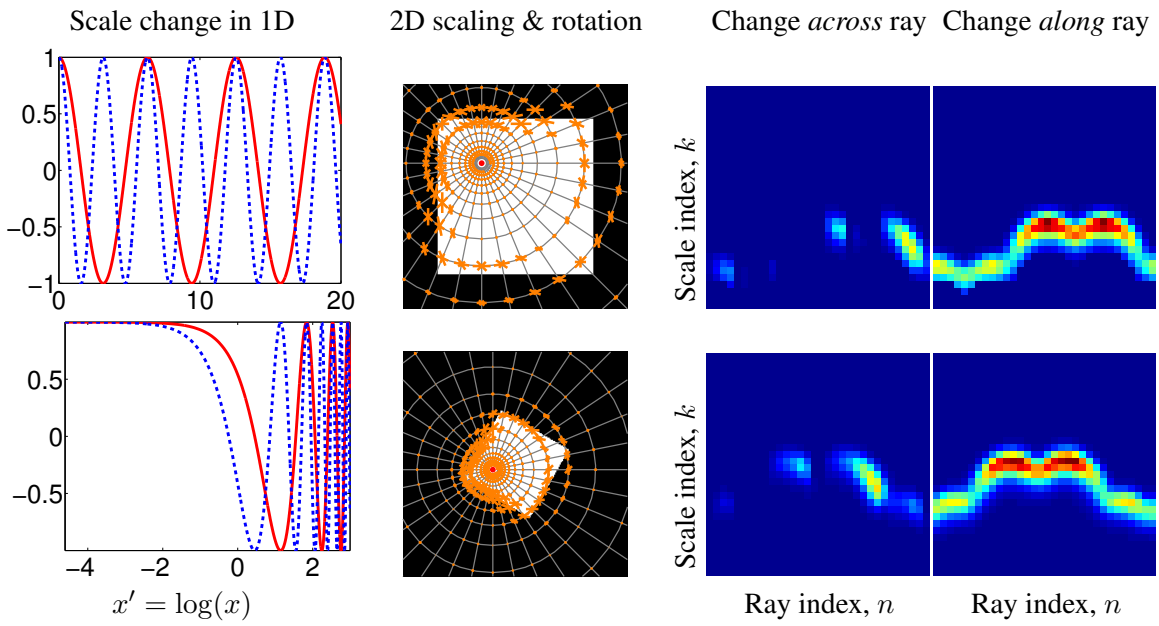


Figure 3.1: Turning scaling into translations for 1D and 2D Signals: The left column demonstrates for a 1D signal how the logarithmic transformation $x' = \log(x)$ turns scaling into translation: the red-solid ($f(x) = \cos(x)$) and blue-dashed ($g(x) = \cos(2x)$) functions differ by a scale factor of two; the transformation $f'(x) = f(\log(x))$ delivers $f'(x) = \cos(\log(x)), g'(x) = \cos(\log(x) - \log 2)$, which differ by a translation. The next columns illustrate the same effect for 2D signals. The second column shows the descriptors computed on a point before and after scaling and rotating an image; the needle length indicates directional derivative magnitude. The next two columns show the respective magnitudes across and along the ray direction, demonstrating that image scaling and rotation is turned to a translation. The point is arbitrary (i.e. not a corner/junction/blob center), therefore scale selection around would not be reliable, or even feasible.

Scale Invariance without Scale Selection

We consider first describing a one-dimensional signal $f(x),\ x > 0$ in a manner that does not change when the signal is scaled as $f(x/a),\ a > 0$. Using the domain transformation $x' = \log(x)$ we can define a new function f' such that

$$f'(x') \doteq f(x), \quad \text{where } x' = \log x, \tag{3.1}$$

which is what we will be referring to as the 'logarithmically transformed' version of f; this is illustrated also in the top row of Fig. 3.1. For this particular transformation, dilating f by a will amount to translating f' by a constant, $\log a$:

$$f'(x' - \log(a)) \quad = \quad f(x/a), \quad \text{or,} \tag{3.2}$$

meaning that we turn dilations of f to translations of f'.

Based on this transformation, we can extract a scale-invariant quantity based on the fact that if $g(x)$ and $G(\omega)$ are a Fourier transform pair, $g(x-c)$ and $G(\omega)e^{-j\omega c}$ will be a transform pair as well (by the shifting-in-time property). Defining $f_a(x') = f'(x' - \log(a))$, and denoting by $F_a(\omega)$ the Fourier Transform of $f_a(x)$ we then have:

$$\mathcal{F}_a(\omega) = \mathcal{F}_1(\omega)e^{-j\log(a)\omega}, \quad \text{or,} \tag{3.3}$$

$$|\mathcal{F}_a(\omega)| = |\mathcal{F}_1(\omega)|. \tag{3.4}$$

From Eq. 3.4 we conclude that changing a will not affect the Fourier Transform Modulus $|\mathcal{F}_a(\omega)|$ of f_a, which can thus be used as a descriptor of f that is invariant to scaling.

As shown in the bottom row of Fig. 3.1, a 2D scaling and rotation can similarly be converted into a translation with a log-polar transformation of the signal - and then eliminated with the FTM technique. The principle behind this approach is commonly used in tasks involving global transformations such as image registration [1, 2] and texture classification [14] and is broadly known as the 'Fourier Transform Modulus' (FTM) technique.

Scale Invariant Descriptor (SID) construction

Our main contribution consists in adapting the FTM technique to the construction of local descriptors. As such, our technique has the potential to apply to scenes with local, non-rigid transformations. This requires firstly a discrete formulation. We construct a descriptor around a point $\mathbf{x} = (x_1, x_2)$ by sampling its neighborhood along K rays leaving $\mathbf{x}$ at equal angle increments $\theta_k = 2\pi k/K,\ k = 0, \ldots, K-1$. Along each ray we use N points whose distances from $\mathbf{x}$ form a geometric progression $r_n = c_0 a^n$. The signal measurements on those points provide us with a $K \times N$ matrix:

$$h[k, n] = f\left[x_1 + r_n \cos(\theta_k), x_2 + r_n \sin(\theta_k)\right], \tag{3.5}$$

With this sampling scheme image scaling and rotation will turn into horizontal and vertical translation of h. From the time-shifting property of the Discrete-Time Fourier Transform (DTFT) we know that if $h[k, n] \overset{\mathcal{F}}{\leftrightarrow} H(j\omega_k, j\omega_n)$ are a DTFT pair, we will then have:

$$h[k-c, n-d] \overset{\mathcal{F}}{\leftrightarrow} H(j\omega_k, j\omega_n)e^{-j(\omega_k c + \omega_n d)}, \tag{3.6}$$

therefore taking the absolute of the DTFT yields a scale- and rotation- invariant quantity.

One subtlety is that image scaling does not result in a cyclic permutation of the array elements, but rather introduces new observations at fine scales and removes others at coarse scales when usampling and vice versa for downsampling. Experimentally it turns out that scaling factors in the order of 2 or 3 do not have noticeable effects on the descriptor's invariance, but after that performance starts dropping.

Apart from a discrete formulation, we also need to preprocess the image so as to (a) discount illumination changes, (b) allow sparse sampling, and (c) allow for efficient dense computation. For illumination invariance we had originally [3] considered using the monogenic signal [15], but currently [4, 5] rely on Daisy-like descriptors [16] as these are simpler to compute. Namely,

instead of the signal values, we sample the directional derivatives of the signal, along a set of orientations offset by the current ray's orientation (see e.g. Fig. 3.1 for the components along, and perpendicular to the ray). For sparse sampling we showed [3] that invariance requires a 'foveal' smoothing pattern [17–19], using a smoothing scale that is linear in the distance from the descriptor's center. This is incorporated in the construction of Daisy and our own adaptation. Finally, for memory- and time- efficient dense computation we combine Daisy [16] with steerable filtering [20] and recursive Gaussian convolution [21]. This allows us to compute 136x170 descriptors for a 700x1000 image in approximately 10 seconds [4].

Results

Starting with qualitative results, in Fig. 3.2 we show the values of the lowest frequency coefficients of densely computed descriptors on two images related by scaling and rotation. We see that the descriptor values are effectively invariant, despite a scaling factor in the order of 2. In the last column we use the two points on the left image as references and 'query' the right one for points having similar descriptors. We then show the similarity of 'query' point descriptors to the red (left) and the green (right) reference points, which indicates the discriminative ability of the descriptor - even though locally the structures are similar, the context helps disambiguate them.

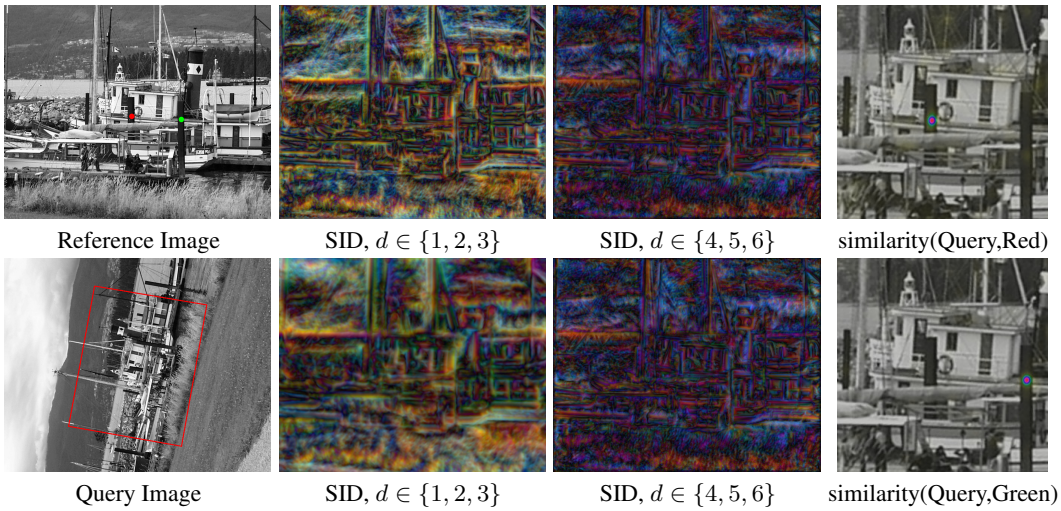


Figure 3.2: Visualization of dense SID: the location of the reference image within the query image is indicated by the red box; the scaling transformation amounts to an area change in the order of four. We align query descriptors after their computation, and visualize three of their dimensions as R, G, and B channels - demonstrating that they are effectively invariant. The bottom row shows as a hue map the similarity between query descriptors and reference points (colorful means large).

For quantitative evaluations we use the dataset, code and protocol of [22]: ground truth correspondences between two images of an identical scene are used to evaluate interest point matches

that are found based on descriptor similarities. Even though our descriptors can be computed densely, we use the Hessian- and Harris- Laplace interest point operators of [22], in order to compare with SIFT descriptors on equal grounds. We do not compare with Hessian- and Harris- affine detectors, as our descriptors are not designed to cope with affine transformations. We refer to [4] for a more thorough description of the evaluation procedure.

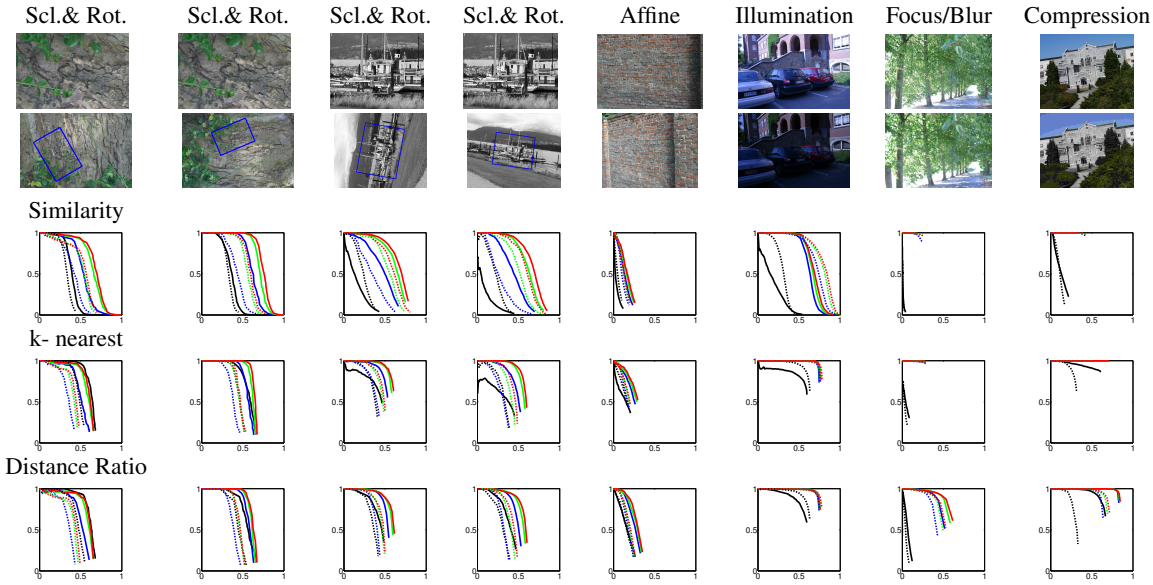
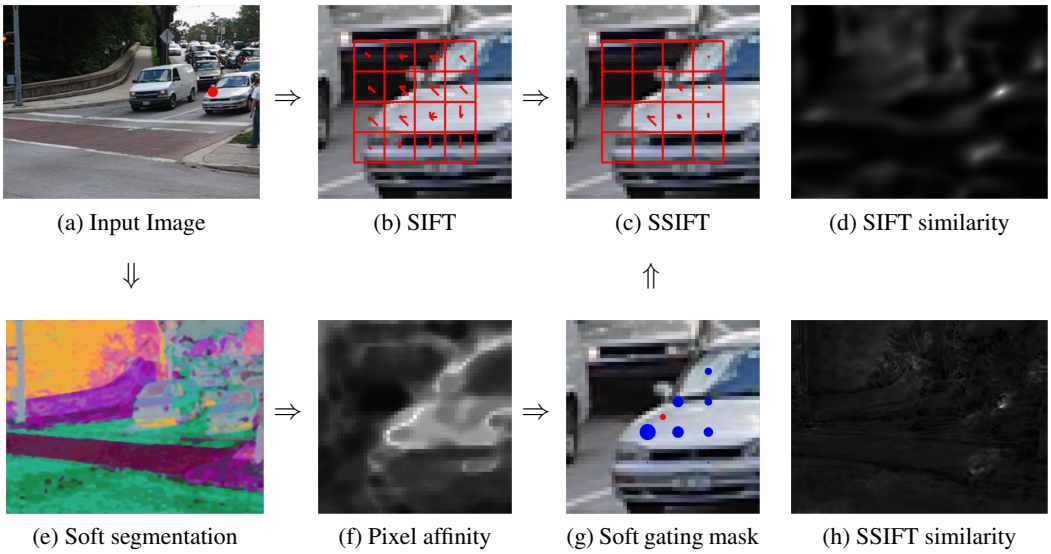


Figure 3.3: Precision-Recall curves comparing SIFT (black) to our descriptor computed with a maximal ring size of 100-red 150-green and 200-blue pixels; solid curves are for Hessian-Laplace interest points, dashed for Harris-Laplace. Left: scaling and rotation transformations. Right: other transformations.

In Fig. 3.3 we use all three of the evaluation measures proposed in [22] ('similarity', 'k-nearest', 'distance-ratio'), to compare our descriptor to SIFT. According to all three criteria our descriptor outperforms SIFT for a broad range of scale changes; moreover, as the radius of the descriptor shrinks, performance degrades, but gracefully. However, above a certain amount of image scaling the results become ambiguous, with SIFT performing equally well or better, as is the case in the first column, particularly for small radii. From several additional experiments that we have conducted with synthetic transformations we have concluded that for area changes up to an order of 9 (equivalently, resolution changes by a factor of 3) our approach can give results that are at least as good as those of SIFT.

One caveat is that the dimensionality of our descriptor, as implemented in [4] is one order of magnitude larger than SIFT, which could skew the results in our favor; in on-going work we have reduced our descriptor's dimensionality to 128, while maintaining an advantage over SIFT for point matching. Still, what we believe is the main advantage of our approach is that it provides us with densely computable scale- and rotation- invariant descriptors. Our descriptors can thus be used for tasks such as dense image correspondence, as described next.



(a) Input Image (b) SIFT (c) SSIFT (d) SIFT similarity

(e) Soft segmentation (f) Pixel affinity (g) Soft gating mask (h) SSIFT similarity

Figure 3.4: Overview of our method: when extracting a SIFT descriptor around the red point in image (a), background structures influence the descriptor entries, as shown in image (b). To deal with this we use soft segmentation embeddings, shown as an RGB image in (e) for the first 3 dimensions of [23] to measure the probability that neighboring points belong with the descriptor's center, as shown in (f). Using this we construct a gating signal, shown in (g) which yields the Segmentation-aware SIFT (SSIFT) descriptor, shown in (c). SSIFT contains effectively no measurements from the background. In (d) and (h) we illustrate the similarity between the descriptor at the red point of image (a) and descriptors computed densely in the image, using SIFT and SSIFT; we observe that for SSIFT the similarity function is more sharply peaked around the position of the red point, indicating its higher distinctiveness.

3.2 Segmentation-aware descriptors

A practical concern about our SID descriptors is that as we sample features at distant 'rings' of our descriptor it becomes increasingly likely that the measurements will come from points belonging to different objects, and are therefore likely to be different in new images. This problem is present in any other descriptor, e.g. SIFT, but becomes more pronounced in SID because of the large distance of the outer rings.

In [5] we address this problem using segmentation information to reduce the effects of occlusion and background changes. As illustrated in Fig. 3.4, we incorporate segmentation information in descriptor construction through a soft 'gating' mask that modulates local measurements, effectively shuning those parts of the image which apparently do not belong to the same object/surface as the descriptor's center. This idea can be traced back to the structure-preserving filtering used in nonlinear diffusion [24, 25], the bilateral filter [26], the segmentation-sensitive normalized convolution [27], and also to the self-similarity descriptor of [28]. What is different is that (i) we use a

Figure 3.5: Soft segmentation cues: We show as RGB maps the first three coordinates of the embeddings provided by the normalized cut eigenvectors of [29] 'Eigen' (middle row) and the PCA subspace formed in the Global boundary technique of [23] ('SoftMask', bottom row).

'mid-level' segmentation module to reason about which pixels go together, rather than relying on low-level cues, and (ii) we combine this with SIFT/SID descriptor construction, rather than image smoothing. This systematically increases the performance of both SID/Daisy and SIFT descriptors on multi-layered motion estimation and wide-baseline stereo.

Soft segmentation masks

Since our goal is to use, rather than solve, segmentation we have avoided hard segmentation schemes, which will inevitably come with errors, but rather preferred to use algorithms that do not strongly commit to a single segmentation; we use their results to determine the affinity of a pixel to its neighbors in a soft manner, and then incorporate it into descriptor construction.

We explore two different approaches to extracting such soft segmentations. Firstly, we use the approach of [29], which first estimates a boundary-based affinity using the 'intervening contour' technique of [30], and then 'globalizes' these affinities by finding the eigenvectors of the relaxed Normalized cut criterion [30]. We also use the soft segmentation masks of Leordeanu et al [23], which are two orders of magnitude faster to compute. There the authors construct local color models around each pixel and form a large set of figure/ground segmentations, which are then projected to a lower dimensional subspace through PCA; this provides us with a low-dimensional pixel embedding in a soft segmentation space. For simplicity, we refer to the eigenvector embeddings of [29] as 'Eigen', and to the soft segmentation masks of [23] as 'SoftMask'. Fig. 3.5 shows the first three coordinates of the 'Eigen'/'SoftMask' embeddings as an RGB image. Note that the embeddings from Gb have higher granularity, which makes them a bit more noisy, but also better suited to capturing smaller features.

We use these quantities to construct a soft affinity mask $\mathbf{w}^{[i]}$ between a point $\mathbf{x}$ and every other point on its descriptor grid, $\mathbf{G}^{[i]}(\mathbf{x})$ as a decreasing function of their embedding distance:

$$\mathbf{w}^{[i]} = \exp\left(-\lambda \|y(\mathbf{x}) - y(\mathbf{G}^{[i]}(\mathbf{x})\|_2^2\right), \tag{3.7}$$

where $y(\cdot)$ is the embedding of point $\cdot$, and λ determines the softness of the mask. Given this mask we modify the measurements extracted around each grid point as $\mathbf{D}'^{[i]} = \mathbf{w}^{[i]}\mathbf{D}^{[i]}$ where for SID $\mathbf{D}^{[i]}$ is the concatenation of oriented Gaussian derivatives at grid point i, while for SIFT $\mathbf{D}^{[i]}$ are the entries of the SIFT cell positioned at $[i]$. As $\mathbf{w}^{[i]} \in [0, 1]$, this has the effect of discounting measurements that come from the background (occluders, background objects), and ensuring that a descriptor is mostly affected by points belonging to the same region.

Results

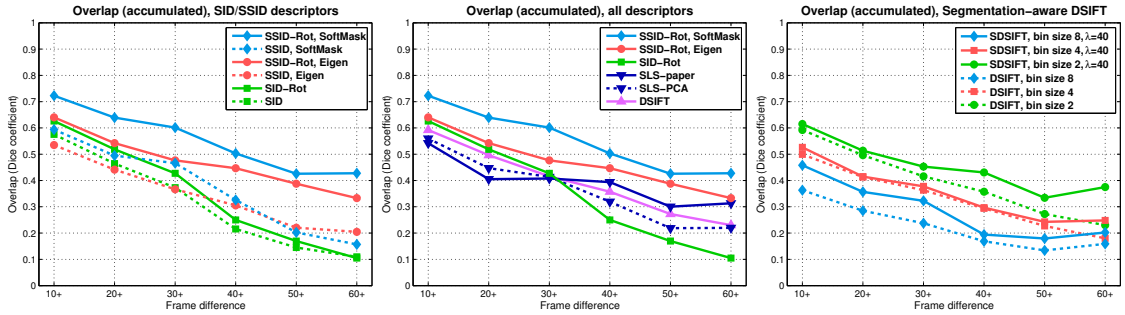
We evaluate the merit of our technique on large-displacement optical flow and wide baseline stereo. We use both 'Eigen' and 'SoftMask' embeddings, and several dense descriptors: SID, dense SIFT -DSIFT [31], and Scale-Less SIFT (SLS) [32]. We use SLS both in its original form and a PCA variant made available by the authors: we refer to them as SLS-paper and SLS-PCA. For SID we also consider a scale-invariant, but rotation-equivariant version, called SID-Rot; this can be obtained by applying the FTM technique only on the scale dimension. We will use the 'S' prefix to indicate 'Segmentation-aware'; for instance 'SSID' stands for the SID variant.

For *large-displacement optical flow* we use the Berkeley/JHU Motion Dataset (Moseg) [33, 34], where ground truth comes in the form of segmentation masks for roughly one every ten frames in a video sequence. We evaluate SSID/SSID-Rot with 'Eigen' and 'SoftMask' embeddings against DSIFT, SLS, SID and SID-Rot. To establish correspondences we use SIFT-Flow [35], which combines dense descriptor matching with optical flow regularization. For any image pair we use the estimated flow to register the segmentation mask in the second image to the first one, and compute their Dice coefficient.

Quantitative results are provided in Fig. 3.6. The first image compares the results for all SID/SSID variants: SSID and SSID-Rot outperform their raw counterparts, SID and SID-Rot by large, in particular for large frame displacements; furthermore, SID-Rot outperforms SID, apparently due to the absence of strong rotation variation. The second image compares the best results obtained from our approach against the other dense descriptors. The best overall results are obtained by SSID-Rot with 'SoftMask' embeddings, followed by the same descriptor with 'Eigen' embeddings. The last image uses the 'SoftMask' embeddings in conjunction with DSIFT, using three different SIFT scales; again, SDSIFT outperforms DSIFT across all scales, albeit with a λ in Eq. 3.7 that may need to be set separately at different scale, as detailed in [5].

Qualitative results are shown in Fig. 3.7; there we use the estimated flow to warp the image in the second column to the image in the first column, therefore a good registration should bring the object in alignment with the segmentation mask. The latter is superimposed on the warped images for visual validation. Again, SSID-Rot outperforms SID-Rot, which in turn is better than other descriptors. Similar improvements are observed when comparing SDSIFT with DSIFT.

For *wide-baseine stereo* we use the dataset of [36], which contains multi-view sets of high-resolution images with ground truth depth maps; we use the 'fountain' set as it contains wider



(a) SID vs. SSID (b) All descriptors (c) DSIFT vs. SDSIFT

Figure 3.6: Average DICE coefficient results on the Moseg Dataset for different descriptor settings; the results are accumulated, so the first bin includes all frame pairs, the second bin includes frame pairs with a displacement of 20 or more frames, and so on.

Figure 3.7: Large displacement image matching using SIFT flow, for the different descriptors considered in our work. We warp image 2 to image 1 using SIFT-Flow with different descriptors. The ground truth segmentation masks of image 1 are overlaid in red; we observe that the segmentation-aware variant SSID-Rot does best, and is better than SID-Rot.

baselines. For these kinds of image pairs occlusions become very pronounced and can hamper any appearance-based matching algorithm. The wide-baseline stereo algorithm described in [37] handles occlusions by using latent occlusion masks; these are updated through an iterative process that gradually refines the depth estimates. Our main result is that we can attain performance comparable to this iterative scheme while using a single-shot algorithm.

We use a set-up similar to that of [37], namely we discretize 3D space into $k = 50$ bins, we use epipolar constraints and the range of the scene to restrict the candidate matches, and use Tree-

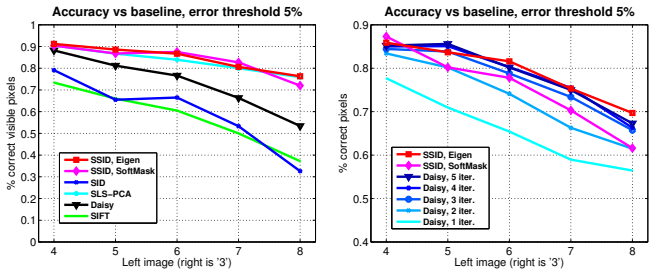


Figure 3.8: Descriptor matching accuracy as a function of stereo baseline: The left image shows accuracy for visible pixels only, and right image compares the iterative variant of Daisy to our single-shot approach, on both visible and occluded pixels.

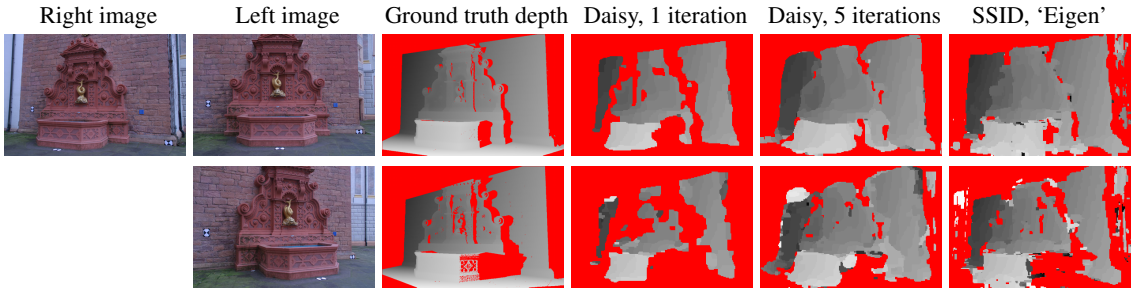


Figure 3.9: Daisy-based vs. SSID-based wide-baseline stereo, for two different baselines (images from [36]): we observe that while employing a single-shot algorithm, we obtain similar, or better, results to those of the iterative Daisy algorithm at convergence.

Reweighted Message Passing [38] to enforce piecewise smoothness. We first evaluate descriptor accuracy only on visible pixels using the ground truth visibility maps from [36]. We use SID, SSID, DSIFT, Daisy and SLS-PCA; for DSIFT, Daisy and SLS-PCA we align the descriptors with the epipolar lines for rotation invariance, as in [37]. As shown in the left image of Fig. 3.8, our segmentation-aware descriptors outperform most others, except occasionally for SLS-PCA, which is however more than two orders of magnitude slower. In the right image of Fig. 3.8 we compare to the full-blown, iterative Daisy stereo algorithm for 5 iterations; our SSID-based algorithm performs comparably, or better than Daisy on most baselines, while being single-shot and not relying on the calibration data to rotate patches. Figure 3.9 displays depth estimates at two baselines.

We note that our descriptors are of higher dimensionality than Daisy - we are now working on reducing their dimensions to make the results directly commensurate. We are also considering ways of extending these results to higher-level tasks, such as object classification, or detection. The recent results of [39] suggest that incorporating segmentation features can largely increase detection accuracy, we are therefore interested in examining whether segmentation-aware features can yield a complementary increase in performance.

3.3 Scale-invariant surface descriptors

We now describe how the idea of achieving scale-invariance without scale selection transfers to the analysis of surfaces. A rich set of applications around non-rigid surface analysis emerge with the growth of internet repositories of geometric data, such as Google Warehouse [40], and also the advent of commercial depth sensors, such as Microsoft's Kinect. Surface retrieval is challenging because of changes in scale, orientation, non-rigid deformations, missing data, and differences in shape formats and representations. We begin by outlining the work of [7] on Heat Kernel Signatures and then describe the results of two collaborative works [4, 6] on making these descriptors scale-invariant, and context-aware, respectively.

Heat Kernel Signatures

A well-established fact in image processing is that the Gaussian scale space of an image $I(\mathbf{x})$ can be obtained by solving the heat diffusion equation:

$$\left(\Delta + \frac{\partial}{\partial t}\right) u = 0, \tag{3.8}$$

with initial condition $u(\mathbf{x}, 0) = I(\mathbf{x})$, where Δ is the Laplacian operator; the solution $u(\mathbf{x}, t)$ of this partial differential equation delivers at any point a one dimensional signature $u(\mathbf{x}, \cdot)$ that captures multi-scale image properties around $\mathbf{x}$. Heat propagation on non-Euclidean domains is governed by a similar diffusion equation:

$$\left(\Delta_X + \frac{\partial}{\partial t}\right) u = 0, \tag{3.9}$$

where now Δ_X denotes the *Laplace-Beltrami operator* on the surface X, a Riemannian equivalent of the Laplacian. The solution of Eq. 3.9 for a point heat distribution $u_0(\mathbf{x}) = \delta(\mathbf{x} - z)$ is called the *heat kernel* $K_{X,t}(x, z)$ and can be presented as [41]:

$$K_{X,t}(\mathbf{x}, \mathbf{z}) \quad = \quad \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(\mathbf{x}) \phi_i(\mathbf{z}), \tag{3.10}$$

where $\lambda_0, \lambda_1, ... \geq 0$ are eigenvalues and $\phi_0, \phi_1, ...$ are the corresponding eigenfunctions of the Laplace-Beltrami operator, satisfying $\Delta_X \phi_i = \lambda_i \phi_i$. Approximating Eq. 3.10 with only the first few (50) eigenpairs of the Laplace-Beltrami operator allows for an efficient computation of the heat kernel, with negligible loss in accuracy.

Sun *et al.* [7] constructed the *heat kernel signature* (HKS) of a point $\mathbf{x}$ by sampling the diagonal of the heat kernel at a set of q time values: $\mathbf{h} = (h(t_1), ..., h(t_q))$, where $h(t) = K_{X,t}(\mathbf{x}, \mathbf{x})$. The HKS captures multi-scale shape curvature information and is invariant to isometries of X, by virtue of being intrinsic. As shown in Fig. 3.10, this means intuitively that as long the as surface deformations do not 'stretch' or 'squeeze' X the descriptors do not change. Furthermore, since there exist methods for Laplacian discretization on different shape representations such as meshes, point clouds, volumes, and implicit surfaces, heat kernel descriptors are particularly versatile.

Figure 3.10: Invariance of the first three components of the HKS (shown as R, G, and B channels, respectively), for a shape undergoing isometric transformations.

Several extensions and variations of HKS have been proposed: in [42] the *wave kernel signature* (WKS) was proposed as a solution to the excessive sensitivity of the HKS to low-frequency information, in [43] a generalization of HKS to a broader, and learnable, family of descriptors is proposed, while [44] developed volumetric variants of HKS. As of today, these descriptors achieve state-of-the-art performance in shape retrieval [45].

Scale-invariant heat kernel signatures

A notable disadvantage of heat kernel descriptors is their sensitivity to shape scaling. Given a shape X and its version X' uniformly scaled by the factor of a, the eigenfunctions and eigenvalues of X' will be given by $\lambda' = a^{-2}\lambda,\ \phi' = a^{-1}\phi$, so the corresponding HKS at x satisfies:

$$h'(t) \quad = \quad \sum_{i\geq 0} e^{-\lambda_i a^{-2} t}\phi_i^2(x)a^{-2} = a^{-2}h(a^{-2}t). \tag{3.11}$$

Typically, the scaling factor a is unknown a priori. Even though global shape normalization is possible, a local method of removing scale dependence is desirable, for instance to accommodate local scaling transformations, or large surface holes. Furthermore we would like to achieve this densely, while only a tiny fraction of surface points allow for some reliable scale selection. In [6] we adapt the FTM technique to the problem at hand, discarding the dependence of h on the unknown scaling factor a; this results in a *Scale-Invariant Heat Kernel Signature* (SI-HKS).

For this, at each point x we sample the heat kernel scale at a geometric progression of time intervals, denoted here with slight abuse of notation as $h(\tau) = h(\alpha^\tau) = h_{\alpha^\tau}(x, x)$. In words, the value of our new function at time τ equals the diagonal of the heat kernel at time α^τ. In this setup, the HKS of the scaled shape becomes $h'(\tau) = a^{-2}h(\tau + 2\log_\alpha a)$. We have thus constructed a function that turns surface scaling into function translation, and multiplication by a constant. In order to remove the dependence on the multiplicative constant a^{-2}, we take the logarithm and then

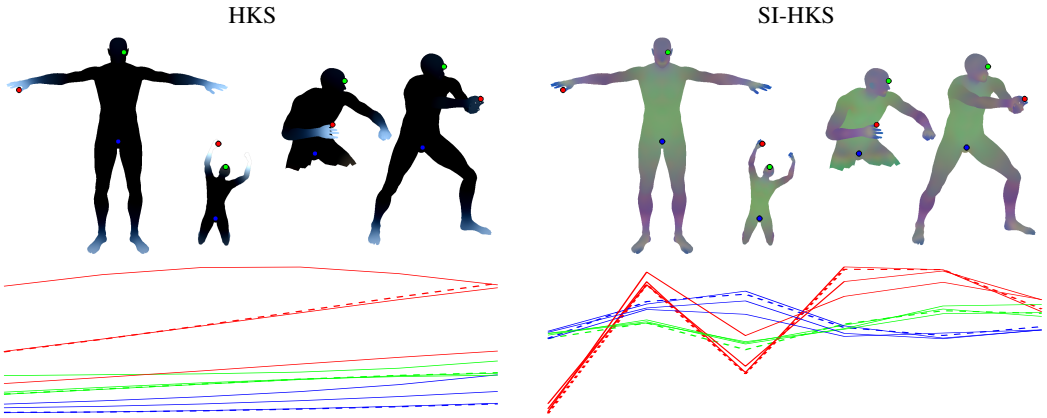


Figure 3.11: Top: three components of the HKS (left) and the proposed SI-HKS (right), represented as RGB color and shown for different shape transformations (null, isometric deformation+scale, missing part, topological transformation). Bottom: HKS (left) and SI-HKS (right) descriptors at three points of the four shapes (different points are coded with red, green, and blue; dashed line shows the null shape descriptor). We observe that the SI-HKS descriptors are substantially more robust to the deformations and stay closer to the null shape descriptor.

the derivative with respect to the scale variable; denoting by

$$\tilde{h}(\tau) \quad = \quad \frac{d}{d\tau} \log h(\tau) \tag{3.12}$$

we obtain a *scale-equivariant* HKS: scaling a surface by α will result in a transformation of $\tilde{h}'(\tau)$ as $\tilde{h}'(\tau) = \tilde{h}(\tau + 2\log_\alpha a)$. We are now again in the setting described in the introductory section, and can obtain an invariant signature by taking the Fourier Transform Modulus of $\tilde{h}'$. Figure 3.11 shows an example of SI-HKS construction; the Laplacian is computed on a triangulated mesh of a human shape undergoing different deformations. We observe that our scale-invariant descriptor is almost entirely invariant to the combination of scaling and isometric bending transformations.

Results

We evaluate our descriptor in the setup of Shape Google [40], using HKS/SI-HKS to construct global shape descriptors following the bag of features paradigm used in image retrieval [46]. First, performing clustering in the HKS space, a *geometric vocabulary* consisting of representative heat kernel signatures ("geometric words") is constructed. Next, for each point on the shape, the HKS is replaced by the index of the most similar geometric word in the vocabulary. Finally, the distribution of geometric words on the shape is computed, resulting in a bag of features representation, which is used for retrieval based on the L_1 distance.

We use the dataset and protocol of the SHREC 2010 robust large-scale shape retrieval benchmark [47]. The dataset consists of 715 shapes from 13 shape classes with simulated transformation and 456 "distractor" shapes from different classes. The query set consists of 13 shapes taken from the dataset (null shapes), with simulated transformations of different type and strength applied to

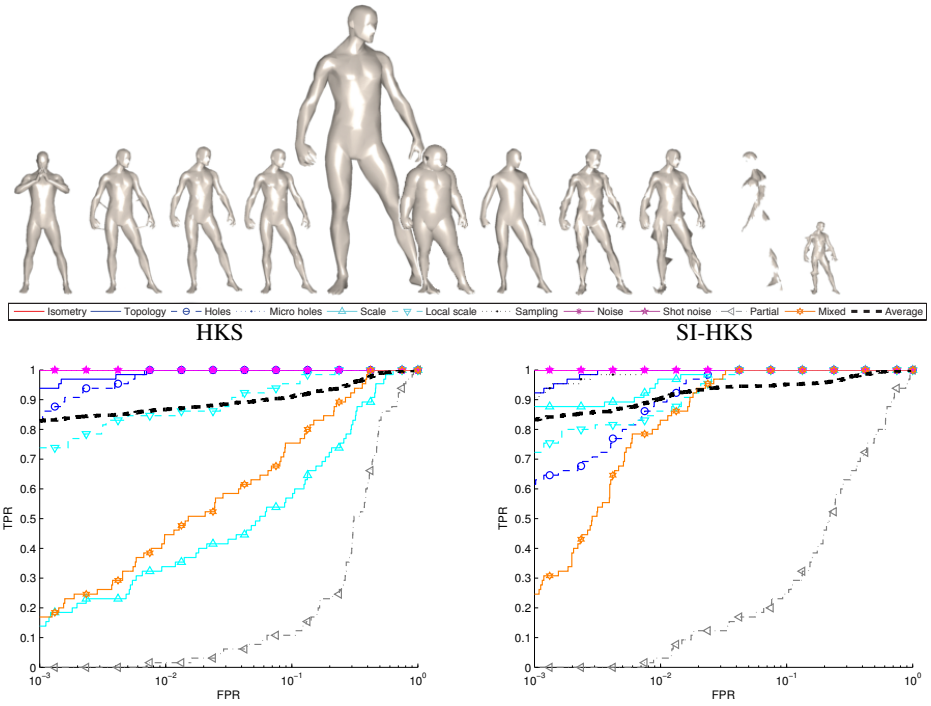


Figure 3.12: ROC curves comparing the shape retrieval performance of bags of features based on HKS and our SI-HKS descriptors. Examples of shape transformations classes are shown above.

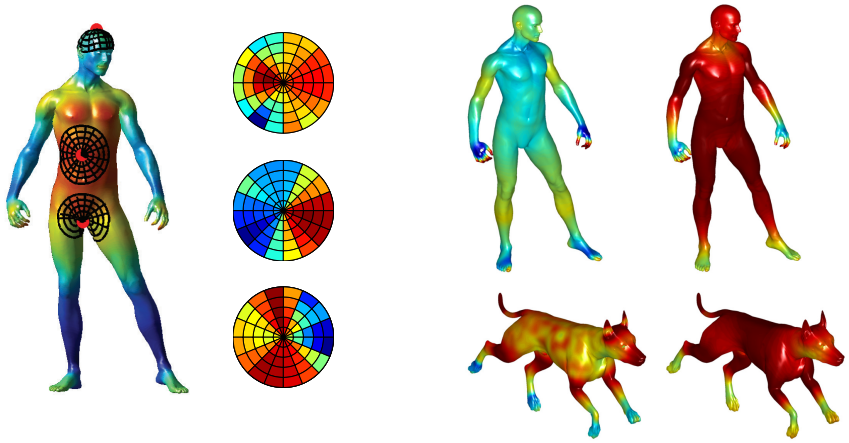
them (Figure 3.12, top). Each query has only one correct corresponding null shape in the dataset. Numerical computation aspects are detailed in [4, 6].

Table 3.1 reports the mean average precision of shape retrieval using bags of features based on HKS and SI-HKS local descriptors. SI-HKS shows dramatic improvement (from 27.42% to 98.21% mAP and from 30.34% to 65.07%) in the *scale* and *mixed* transformations classes, respectively. Overall in all transformation classes and strengths SI-HKS performs better than HKS (90% vs. 85.00%). These results are also reflected in the Receiver-Operating-Curve (ROC) plots of Fig. 3.12.

	Strength				
Transform.	**1**	**≤2**	**≤3**	**≤4**	**≤5**
Isometry	100.00	100.00	100.00	100.00	100.00
Topology	100.00	98.08	97.44	96.79	96.41
Holes	100.00	100.00	97.44	95.19	90.13
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	0.98	40.68	43.31	33.72	27.42
Local scale	100.00	100.00	98.72	89.38	80.22
Sampling	100.00	100.00	100.00	100.00	99.23
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	100.00
Partial	7.54	5.70	4.51	3.58	2.95
Mixed	53.13	55.86	47.77	37.54	30.34
Average	94.94	93.12	90.84	87.82	85.00

	Strength				
Transform.	**1**	**≤2**	**≤3**	**≤4**	**≤5**
Isometry	100.00	100.00	100.00	100.00	100.00
Topology	96.15	96.15	94.87	93.27	92.69
Holes	100.00	100.00	100.00	94.71	89.97
Micro holes	100.00	100.00	100.00	100.00	100.00
Scale	91.03	95.51	97.01	97.76	98.21
Local scale	100.00	100.00	97.44	89.38	82.08
Sampling	100.00	100.00	100.00	100.00	97.69
Noise	100.00	100.00	100.00	100.00	100.00
Shot noise	100.00	100.00	100.00	100.00	100.00
Partial	17.43	10.31	9.57	8.06	6.61
Mixed	56.47	57.44	63.59	67.47	65.07
Average	97.05	95.16	94.03	92.54	90.79

Table 3.1: Retrieval Performance (mAP in %) using HKS (left) and SI-HKS (right) local descriptors.



ISC construction SI-HKS vs ISC + SI-HKS

Figure 3.13: The Intrinsic Shape Context (ISC) descriptor is constructed as a histogram of some field, using an intrinsic local polar coordinate system to define the histogram bins; on the left we show such coordinate systems on three different points; the associated descriptors are shown in the second row. The ISC construction allows us to extract 'meta descriptors' on top of any existing descriptor, such as SI-HKS. Shown in the next two columns are the normalized Euclidean distances between (i) the descriptor at a reference point on the right hand (white dot) and descriptors computed (ii-top row) on other points of the same shape, or (iii-bottom row) a distinct shape using the SIHKS descriptor (left) and the ISC-SIHKS (right). We observe that the introduction of spatial context improves the discriminative ability and localization of the descriptor, while still being able to generalize to similar structures on the dog surface.

3.4 Intrinsic Shape Context descriptors

In [8] we go beyond point signatures and develop the *intrinsic shape context* (ISC), a generalization of shape contexts [9] to surfaces. Unlike spin images [48] or more recent works in this direction [49], our ISC descriptor is invariant to isometric non-rigid surface deformations.

Our construction consists in forming a local chart around every surface point, and averaging within the chart's compartments a vector field, such as the HKS or SI-HKS. As shown in Fig. 3.13, we can thereby construct meta-descriptors that improve the discriminative power of any point signature. Using systematic evaluations we have verified that introducing spatial context consistently improves performance in matching and retrieval.

Intrinsic Shape Context Construction

We start by describing the original Shape Context (SC) descriptor in a formulation that will serve as a stepping stone for its generalization to surfaces. The Shape Context descriptor describes a field $I(x)$, $x \in \mathbb{R}^2$ around a point x_i by averaging I over a grid formed by N_θ angular and N_ρ

radial bins centered at x_i. This results in a $N_\rho \times N_\theta$-dimensional descriptor:

$$\mathcal{S}_{\rho,\theta}(x_i) \quad = \quad \frac{\int_{\mathbb{R}^2} \pi_{\rho,\theta}(x) I(x) dx}{\int_{\mathbb{R}^2} \pi_{\rho,\theta}(x) dx}, \tag{3.13}$$

where $\pi_{\rho,\theta} = \pi_\rho \pi_\theta$ denotes the membership function of angular bin θ and radial bin ρ,

$$\pi_\rho(x) = \begin{cases} 1 & \|x - x_i\|_2 \in R_\rho, \\ 0 & \text{else}, \end{cases}, \quad \pi_\theta(x) = \begin{cases} 1 & \angle(x - x_i) \in R_\theta \\ 0 & \text{else}, \end{cases} \tag{3.14}$$

with R_ρ, R_θ denoting the supports of the radial and angular bins centered around ρ, θ, respectively. Thus $\mathcal{S}_{\rho,\theta}$ is the average of $I(x)$ over the bin ρ, θ; the hard binning can be replaced by soft binning using probabilistic membership functions, which amounts to forming a weighted average.

There are certain technical challenges involved in extending this scheme to surfaces, involving (i) the development of robust numerical schemes for charting with angular and radial coordinates, (ii) the averaging of a field that is defined over a discretized surface, and (iii) the orientation ambiguity that arises in the construction of local coordinate systems on surfaces. We have addressed these in [8], as we briefly review below.

For *surface charting* around a point we 'shoot' and then track geodesics going outwards from vertex x_i. These provide us with the counterpart of rays in a log-polar mapping, i.e., they are surface loci with constant intrinsic angular coordinate. The set of directions in which rays are being shot is established by partitioning the *1-ring* neighborhood of x_i into segments of equal angle. These directions are propagated using the unfolding technique of [50] which tracks a geodesic along adjacent triangles as depicted in the right image of Fig. 3.14.

We recover the surface counterpart of circles by computing geodesic distances between the point x_i and all surface points x_j using the fast marching method (FMM) [51]. We can then recover equidistant points as r-level sets of the function $d_X(x_i, x) = r$, where we denote by $d_X : X \times X \to \mathbb{R}_+$ the geodesic distance function, measuring the length of the shortest path on the mesh between any pair of vertices.

In order to *average a field* over radial and angular surface bins, we develop intrinsic equivalents to the definitions in Eq. 3.13. In particular the ISC descriptor $\mathcal{S}_{\rho,\theta}(x_i)$ at point x_i is given by

$$\mathcal{S}_{\rho,\theta}(x_i) \quad = \quad \frac{\int_X \pi_{\rho,\theta}(x) I(x) d\mu_X(x)}{\int_X \pi_{\rho,\theta}(x) d\mu_X(x)}, \;\; d\mu_X(x) = \frac{1}{3} \sum_{\substack{x_i, x_j \in N_1(x) \\ (x_i, x_j) \in E}} \text{area}(x, x_i, x_j) \tag{3.15}$$

where $d\mu_X(x)$ is the local area element, equal to one third the area of the 1-ring neighborhood of x, and $\pi_{\rho,\theta} = \pi_\rho \pi_\theta$, as previously, denotes the membership function of intrinsic angular bin θ and intrinsic radial bin ρ. As detailed in [8], to compute π_ρ and π_θ intrinsically we estimate geodesic distances to the radial-polar grid elements using the Fast Marching Method; instead of thresholding, we turn these distances into soft membership functions, thereby ensuring robustness to irregular surface sampling schemes.

Finally, we eliminate the *rotation ambiguity* that shows up during descriptor construction; in particular, our method for constructing $\mathcal{S}_{\rho,\theta}$, could equally well result in $\mathcal{S}_{\rho,\theta+c \bmod 2\pi}$, if the first ray was chosen with another offset, c. For this we use again the FTM technique and obtain an invariant quantity by taking the absolute of the Fourier transform of $\mathcal{S}_{\rho,\theta}$ along θ.

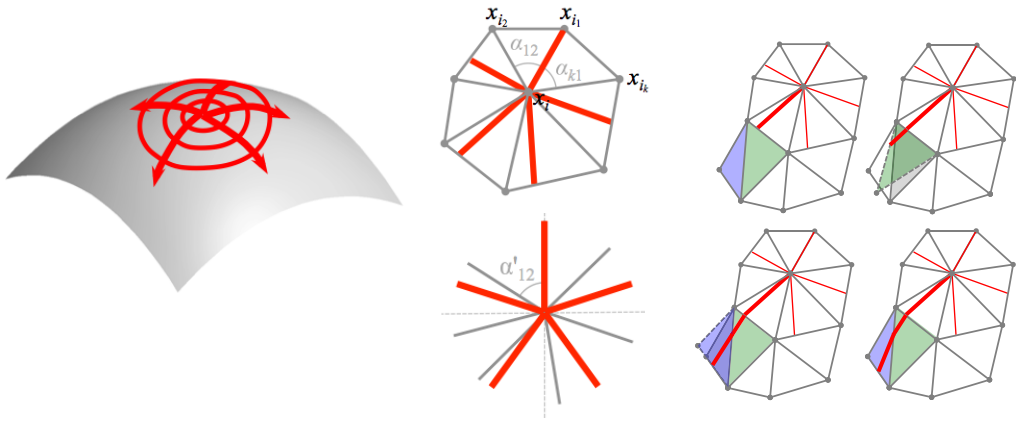


Figure 3.14: Geometric computations involved in the construction of ISC. Left: 'Outward shooting' creation of the intrinsic polar grid: we regularly sample a set of directions around a point and use them to initialize trajectories on the surface; these serve as the 'rays' of our intrinsic grid. Middle: The creation of a regular intrinsic angular chart at surface point x_i involves mapping its 1-ring onto the plane by using a uniform angle scaling factor, dividing the plane into equal angular segments, and mapping them back onto the 1-ring. Right: unfolding-based propagation of direction from 1-ring triangle: The green triangle adjacent to the 1-ring triangle is mapped to the plane of the latter, shown dashed, on the top right. The 1-ring triangle's direction (red line) is continued in the green triangle until it hits an edge. This is repeated for the next adjacent triangle (blue, on the bottom row), and continued until the resulting polyline's length reaches some threshold.

Experimental results

We have experimented with ISC based on HKS [7] and SIHKS [6] dense descriptors, as detailed in [8]. Starting with some qualitative results, the right part of Figure 3.13 visualizes the distance maps computed in the descriptor space from a reference point on the human shape to the rest of the points on that shape, as well as to the points of the dog shape. Two phenomena are clearly visible: First, using the same base descriptor (SIHKS) the ISC gives significantly better feature localization, in the sense that the distance grows fast as one moves away from the reference point. Second, the ISC exhibits better discriminative ability by assigning higher relative distances to the points of the distinct dog shape, while the raw SIHKS confuses between the reference point and some points on the dog's paws.

The performance of our descriptor was evaluated quantitatively on the SHREC'10 robust correspondence benchmark [52]. Figures 3.15–3.16 show the cumulative matching scores (CMC) of the raw HKS and SIHKS descriptors and the ISC descriptors based on them. 1000 points were sampled from each shape using farthest point sampling. Points from the transformed shape were matched to the null shape and ordered using the L_2 distance between the corresponding descriptors. A CMC curve shows the percentage of feature points that had correct match among the first k candidate matches; we note that the first 20 matches capture a substantial amount of correct matches, often higher than 50%. The use of spatial contexts consistently improves the descriptor

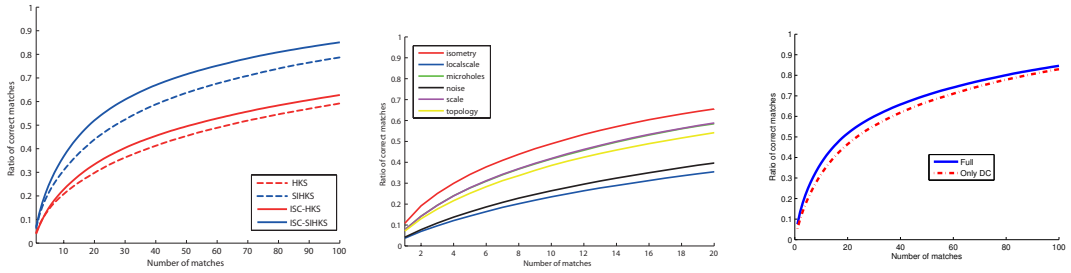


Figure 3.15: Cumulative matching score curves (CMC) averaged over all types of shape transformations for (a) different descriptors, (b) the ISC-SIHKS descriptor and different types of shape transformations, (c) the full-blown ISC-SIHKS descriptor versus using only the DC components.

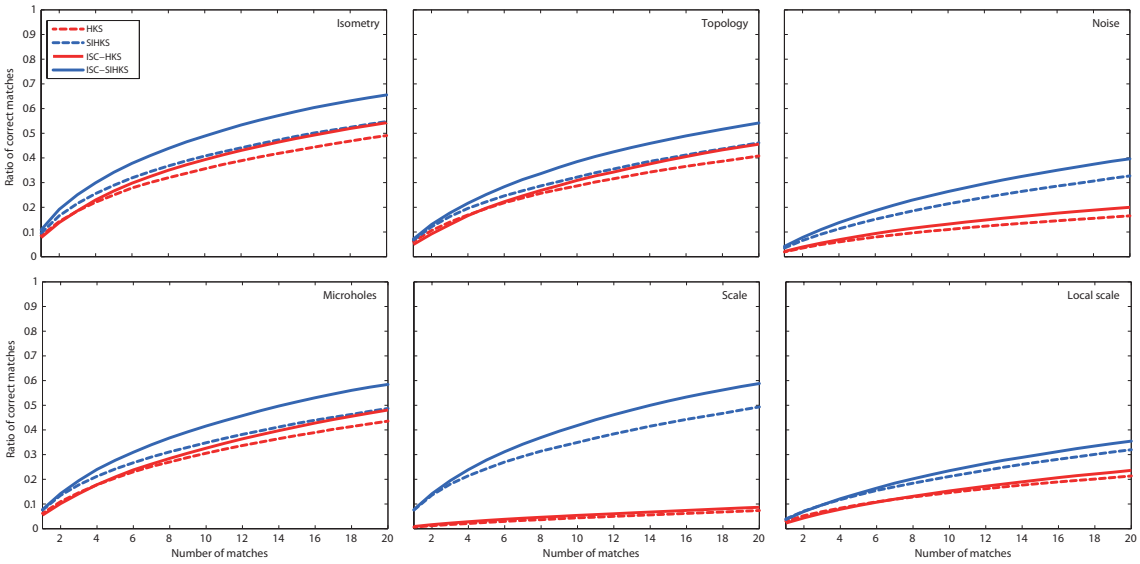


Figure 3.16: CMC curves of different shape descriptors broken down by different types of shape transformations.

performance under all transformations, on the average by over 10%. Moreover, the use of exclusively the DC components of the Fourier Transform -i.e. the result of averaging over orientations- harms performance when compared to using the full-blown descriptor.

Bibliography

[1] D. Casasent and D. Psaltis., "Position, rotation, and scale invariant optical correlation," *Applied Optics*, vol. 15, no. 7, pp. 258–261, 1976.

[2] G. Wolberg and S. Zokai, "Robust image registration using log-polar transform," in *Proc. ICIP*, 2000.

[3] I. Kokkinos and A. Yuille, "Scale invariance without scale selection," in *Proc. CVPR*, 2008.

[4] I. Kokkinos, M. Bronstein, and A. Yuille, "Dense scale-invariant descriptors for image and surfaces," INRIA, Tech. Rep. RR-7914, 2012.

[5] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno, "Dense segmentation-aware descriptors," in *Proc. CVPR*, 2013.

[6] M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Proc. CVPR*, 2010.

[7] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Computer Graphics Forum*, 2009.

[8] I. Kokkinos, M. Bronstein, R. Littman, and A. Bronstein, "Intrinsic shape context descriptors for deformable shapes," in *Proc. CVPR*, 2012.

[9] S. Belongie, Malik, and J. Puzicha, "Shape context: a new descriptor for shape matching and object recognition," in *Proc. NIPS*, 2000.

[10] I. Kokkinos, *Dense image and surface descriptors*, `http://vision.mas.ecp.fr/Personnel/iasonas/descriptors.html`, 2013.

[11] T. Lindeberg, "Feature Detection with Automatic Scale Selection," *Int.l Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[12] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int.l Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] K. Mikolajczyk, A. Zisserman, and C. Schmid, "Shape recognition with edge-based features," in *Proc. BMVC*, 2003.

[14] M. Porat and Y. Zeevi, "The Generalized Gabor Scheme of Image Representation in Biological and Machine Vision," *IEEE Trans. PAMI*, vol. 10, no. 4, pp. 452–468, 1988.

[15] M. Felsberg and G. Sommer, "The monogenic signal," *Trans. Signal Processing*, vol. 49, no. 12, pp. 3136–3144, 2001.

[16] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *Proc. CVPR*, 2008.

[17] E. L. Schwartz, "Spatial Mapping in the Primate Sensory Projection: Analytic Structure and Relevance to Perception," *Biological Cybernetics*, vol. 25, no. 4, pp. 181–194, 1977.

[18] T. Lindeberg and L. Florack, "Foveal Scale-Space and the Linear Increase of Receptive Field Size as a Function of Eccentricity," *CVIU*, vol. 97, no. 2, pp. 209–241, 1996.

[19] A. Berg and J. Malik, "Geometric blur for template matching," in *Proc. CVPR*, 2001.

[20] W. T. Freeman and E. H. Adelson, "The Design and Use of Steerable Filters," *IEEE Trans. PAMI*, vol. 13, no. 6, pp. 891–906, 1991.

[21] R. Deriche, "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector," *Int.l Journal of Computer Vision*, vol. 1, no. 2, pp. 167–187, 1987.

[22] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. PAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.

[23] M. Leordeanu, R. Sukthankar, and C. Sminchisescu, "Efficient closed-form solution to generalized boundary detection," in *Proc. ECCV*, 2012.

[24] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. PAMI*, vol. 12, no. 7, pp. 629–639, 1990.

[25] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[26] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. ICCV*, 1998.

[27] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proc. ICCV*, 2003.

[28] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. CVPR*, 2007.

[29] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. PAMI*, vol. 33, no. 5, pp. 898–916, 2011.

[30] J. Shi and J. Malik, "Normalized cuts and image segmentation," *T.PAMI*, 1997.

[31] A. Vedaldi and B. Fulkerson, *VLFeat: an open and portable library of computer vision algorithms*, `http://www.vlfeat.org`, 2008.

[32] T. Hassner, V. Mayzels, and L. Zelnik-Manor, "On SIFTS and their scales," in *CVPR*, 2012.

[33] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Proc. ECCV*, 2010.

[34] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *Proc. CVPR*, 2007.

[35] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense Correspondence Across Different Scenes," *IEEE Trans. PAMI*, vol. 33, no. 5, pp. 978–994, 2011.

[36] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *Proc. CVPR*, 2008.

[37] E. Tola, V. Lepetit, and P. Fua, "DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo," *IEEE Trans. PAMI*, vol. 32, no. 5, pp. 815–830, 2010.

[38] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. PAMI*, vol. 28, no. 10, pp. 1568–1583, 2006.

[39] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun, "Bottom-up segmentation for top-down detection," in *Proc. CVPR*, 2013.

[40] A. M. Bronstein, M. M. Bronstein, M. Ovsjanikov, and L. J. Guibas, "Shape Google: a computer vision approach to invariant shape retrieval," in *Proc. NORDIA*, 2009.

[41] P. Jones, M. Maggioni, and R. Schul, "Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels," *PNAS*, vol. 105, no. 6, p. 1803, 2008.

[42] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature-a quantum mechanical approach to shape analyis," in *Proc. CVPR*, 2011.

[43] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, "LDA-Hash: improved matching with smaller descriptors," *IEEE Trans. PAMI*, vol. 34, no. 1, pp. 66–78, 2012.

[44] D. Raviv, M. M. Bronstein, A. M. Bronstein, and R. Kimmel, "Volumetric heat kernel signatures," in *Proc. ACM Multimedia Workshop on 3D Object Retrieval*, 2010.

[45] A. Bronstein, M. Bronstein, L. Guibas, and M. Ovsjanikov, "Shape Google: geometric words and expressions for invariant shape retrieval," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 1, p. 1, 2011.

[46] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," in *Proc. CVPR*, 2003.

[47] A. Bronstein, M. Bronstein, U. Castellani, B. Falcidieno, A. Fusiello, A. Godil, L. Guibas, I. Kokkinos, Z. Lian, M. Ovsjanikov, *et al.*, "SHREC 2010: robust large-scale shape retrieval benchmark," in *Proc. Eurographics Workshop on 3D Object Retrieval*, 2010.

[48] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *Trans. PAMI*, vol. 21, no. 5, pp. 433–449, 1999.

[49] A. Zaharescu, E. Boyer, K. Varanasi, and R Horaud, "Surface feature detection and description with applications to mesh matching," in *Proc. CVPR*, 2009.

[50] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Efficient computation of isometry-invariant distances between surfaces," *SIAM J. Scientific Computing*, vol. 28, no. 5, pp. 1812–1836, 2006.

[51] R. Kimmel and J. A. Sethian, "Computing geodesic paths on manifolds," *PNAS*, vol. 95, no. 15, pp. 8431–8435, 1998.

[52] A. Bronstein, M. Bronstein, U. Castellani, A. Dubrovina, L. Guibas, R. Horaud, R. Kimmel, D. Knossow, E. von Lavante, D. Mateus, *et al.*, "SHREC 2010: robust correspondence benchmark," 2010.

Chapter 4

Learning Shape Models for Objects and Actions

The task considered in this chapter is to develop object/action models after observing a set of training examples that do not contain any hand-segmented structures; instead we use only weak annotations that come in the form of bounding boxes. We further assume that we have a front-end system that can turn an image, or a video, into a set of geometric primitives, such as contours or point trajectories respectively.

We address three tasks, pursued in [1–3] respectively: (a) bringing a training set of images into registration through non-rigid, class-specific transformation models; this allows us to construct a 'template' for our category and makes subsequent tasks easier by establishing a common, deformation-free 'template' domain (b) discriminative training of hierarchical shape-based models; we use end-to-end training to score candidate image-based object instantiations and (c) discriminative training of a fully-connected part-based model for action recognition; we thereby elicit relative motion information through pairs of part trajectories and jointly recognize and localize actions in videos.

Our results demonstrate that by using a shape-based representation we not only deliver a decision about the presence of an object, but also support it by visual evidence coming in the form of contours/trajectories. This allows for a more detailed interpretation, or 'parsing' of the input.

4.1 Learning object deformation models

Modeling deformations within a category can simplify subsequent learning tasks by factoring out the effects of intra-class shape variability. This idea underlies several works that explicitly [4–6] or implicitly [7, 8] establish correspondences among images in a training set. In our work [1] we build

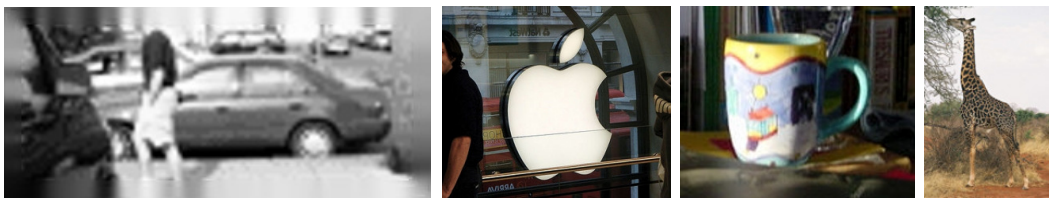


Figure 4.1: Typical images from our training set. Our method can handle substantial amounts of occlusion and does not require manual segmentation annotations.

on the deformable template paradigm and consider that object instances are obtained by deforming a prototypical object, or 'template', through deformation fields modeled by simple parametric models. Given an observed image $I(\mathbf{x})$ and a template $\mathcal{T}(\mathbf{x})$ we assume that a deformation field $\mathbf{S} : R^2 \rightarrow R^2$ maps every template point $\mathbf{x}$ to an image point $\mathbf{S}(\mathbf{x})$ so that:

$$I(\mathbf{S}(\mathbf{x})) \simeq \mathcal{T}(\mathbf{x}). \tag{4.1}$$

There are two broad classes of deformable template models: first, Active Appearance Models (AAMs) [9, 10] model deformations as expansions on a category-specific basis, $\mathcal{S}_1 \ldots \mathcal{S}_N$:

$$\mathbf{S}(\mathbf{x};\mathbf{s}) = (\mathcal{G}_x(\mathbf{x};\mathbf{s}), \mathcal{G}_y(\mathbf{x};\mathbf{s})) \doteq \sum_{i=1}^{N} \mathbf{s}_i \mathcal{S}_i(\mathbf{x}), \tag{4.2}$$

where $\mathbf{s}$ are image-specific expansion coefficients. Second, piecewise-linear, or part-based, deformation models break the deformation field into regions, and account for translation [7, 11–13], scaling, rotation and translation [14], or affine transformations [14, 15] using a separate linear model per region. For instance in [14] in the coordinate system determined by the i^{th} region the template point $\mathbf{x} = (x, y)$ is mapped to the image point $\mathbf{x}' = (x', y')$ as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_i^x \cos(\theta_i) & -s_i^y \sin(\theta_i) \\ s_i^x \sin(\theta_i) & s_i^y \cos(\theta_i) \end{bmatrix} \begin{bmatrix} x - x_i \\ y - y_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & x & y & 0 & 0 \\ 0 & 1 & 0 & 0 & x & y \end{bmatrix} \mathcal{P}_i, \tag{4.3}$$

where (s_i^x, s_i^y), θ_i, and (x_i, y_i) are the scaling, rotation and translation parameters respectively, while $\mathcal{P}_i$, is used for an equivalent, linear parameterization of the deformation.

When learning such models without landmarks there are two different types of unknowns: first, the model parameters ($\mathcal{S}, \mathcal{T}$ for AAMs), which determine the possible deformations and appearance of the object category and, second, the deformation variables ($\mathbf{s}$ for AAMs), which specify the particular transformation needed to model the individual training samples. To tackle this problem we apply the EM algorithm [16], using the M-step to estimate model parameters and the E-step to find the latent deformation variables.

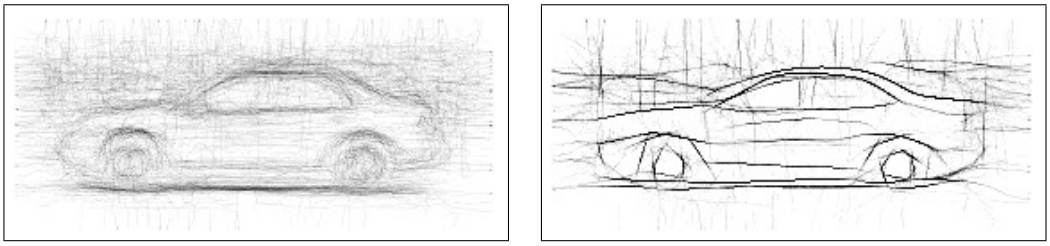
Learning Active Appearance Models

If we know the AAM parameters, $\mathcal{T}(\mathbf{x})$ and $\{\mathcal{S}_i(\mathbf{x}) : i = 1, ..., N_S\}$, we can fit an AAM to an input I by minimizing a least squares criterion $E(\mathbf{s})$ with respect to the expansion coefficients $\mathbf{s}$:

$$E(\mathbf{s}) = \sum_{\mathbf{x}} \left(I(\mathbf{S}(\mathbf{x};\mathbf{s})) - \mathcal{T}(\mathbf{x})\right)^2, \quad \mathbf{S}(\mathbf{x};\mathbf{s}) = \sum_{i=1}^{N} \mathbf{s}_i \mathcal{S}_i(\mathbf{x}) \tag{4.4}$$

AAMs commonly account for appearance variability by using linear models in the expression of $\mathcal{T}$; but we omit this since we use shape-based features. The criterion in Eq. 4.4 is nonlinear in $\mathbf{s}$, but iterative algorithms [9] can deliver good solutions when properly initialized.

However, as we do not know the model parameters, $\mathcal{T}, \mathcal{S}$, we need to resort to an EM-based scheme to jointly estimate all unknowns. Omitting intermediate steps [1, 17] that make the link



Initial After contour-aligned Mean Shift

Figure 4.2: Illustration of the contour alignment used for shape basis initialization: Left: superimposed edge contours from the whole car training set, as aligned by the previous AAM learning iteration. Right: same points, after 6 iterations of contour-aligned Mean-Shift procedure described in the text.

with EM, we phrase the task as optimizing the following criterion:

$$\sum_{\mu=1}^{N} \left[\sum_{\mathbf{x}} \left[I_\mu(\mathcal{G}(\mathbf{x}; \mathbf{s}^\mu)) - \mathcal{T}(\mathbf{x}) \right]^2 + \lambda \sum_j (\mathbf{s}_j^\mu)^2 \right], \tag{4.5}$$

where $\{I_\mu\}, \mu = 1 \dots N$ is our training set and our unknowns consist of the model parameters $\mathcal{T}(\mathbf{x}), \{\mathcal{S}_i(\mathbf{x}) : i = 1, ..., N_S\}$ and the image-specific coefficients associated to every image, $\{\mathbf{s}_\mu\}, \mu = 1 \dots N$; λ determines the penalty for large expansion coefficients, and is set manually.

This criterion can be understood as a log-likelihood expression [1] or a data coding cost [18]. Even though it is non-convex and can have many local minima, we can continuously decrease it with alternating minimization. In particular, minimizing over $\mathcal{T}$ yields

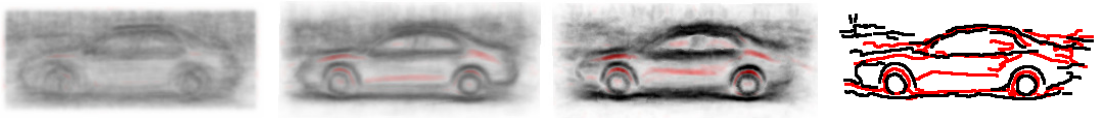
$$\mathcal{T}(\mathbf{x}) = \frac{1}{N} \sum_{\mu=1}^{N} I_\mu(\mathcal{G}(\mathbf{x}; \mathbf{s}^\mu)), \tag{4.6}$$

which updates the template to be the average of the registered input images. Minimizing over $\mathbf{s}$ is done using standard AAM fitting [9]. Minimizing over $\mathcal{S}$ can be done with steepest descent; taking the derivative with respect to the i-th basis element at location $\mathbf{x}$, say $\mathcal{S}_{x,i}(\mathbf{x})$, yields

$$\frac{\partial E}{\partial \mathcal{S}_{x,i}(\mathbf{x})} = \sum_{\mu=1}^{N} \mathbf{s}_i^\mu \left. \frac{\partial I}{\partial x} \right|_{\mathcal{G}^\mu} \left[I_\mu(\mathcal{G}(\mathbf{x}; \mathbf{s}_\mu)) - \mathcal{T}_\mu(\mathbf{x}) \right], \tag{4.7}$$

where $\left. \frac{\partial I}{\partial x} \right|_{\mathcal{G}_\mu}$ denotes the derivative of I along dimension x after warping I to the template grid using $\mathcal{G}(\mathbf{x}; \mathbf{s}_\mu)$; an analogous expression is used for the y coordinate.

One technical concern is that local spatial contractions and expansions in the estimated deformation fields can make template features shrink, or even disappear, leading to trivial minima of Eq. 4.5. We therefore require that template contours are only 'transported', i.e. are not shrunk or dilated, in the direction perpendicular to their orientation. This can be formulated as a problem of



(a) No registration (b) Affine registration (c) AAM registration (d) Contours

Figure 4.3: We illustrate registration results by averaging the edges (black) and ridges (red) of the training set after registration; sharper structures suggest better registration. The contours in (d) are extracted from (c) by non-maximum suppression followed by hysteresis thresholding.

calculus of variations, which leads to an elliptic partial differential equation, detailed in [1]; the solution we used consisted in an iterative update-projection scheme to ensure that the computed deformation bases satisfy this 'transport' constraint.

We follow a greedy procedure, introducing basis elements iteratively, starting from a 'null' model including affine transformations and then gradually adding deformation modes, which are in turn refined by the alternating optimization scheme outlined above. What turned out to be non-trivial was the initialization of the basis elements. Upon convergence of the previous round of registration we gather the positions and orientations of points from all training images into a set $\{(x_i, y_i, \theta_i)\}, i = 1 \dots K$, which we used to construct a nonparametric density in x, y, θ. As in Mean-Shift [19], we find peaks of this density by letting each point move to a position of higher density, but restrict each point to only move along its orientation θ. We call this variant a 'contour-aligned Mean Shift', as it prevents the contraction of curves to points, and results in a set of thin contours, as shown in Fig. 4.2. This provides us with an estimate of the additional deformation that should be applied per image; the new basis element is formed as the first eigenvector found by applying PCA, as in [10].

Learning piecewise linear deformation models

We have also explored the use of a similar, EM-type approach to learn piecewise-linear, part-based deformation models for articulated objects. After AAM learning we used Mean Shift on the registered symmetry maps to initialize the part locations and treated the parameters of the resulting local deformation models as nodes on a tree-structured graph. We then solved the E-step using approximate inference on this graph with Non-parametric Belief Propagation [20], while also using efficient computational techniques to accelerate the estimation of unary terms. These models turned out to be more appropriate for articulated models, like cows or horses, but were also substantially harder to work with and improve due to the approximate and slow inference.

Results

Our main interest has been to apply our framework to real, noisy, unsegmented images and see whether shape information can be extracted without manual annotation. We demonstrate learning results for cars from the UIUC dataset [21] and the ETHZ shape classes (apples, bottles, giraffes, mugs and swans) [22]; indicative examples of the images used for training are shown in Fig. 4.1.

In Fig. 4.3 we visualize the different models by averaging the ridge (red) and edge (black) maps of the training set at different steps of registration. The improvement in registration can be seen by comparing the null model results Fig. 4.3(b) with the ones of the learned AAM in Fig. 4.3(c). The learned model aligns the training images better, as shown by the cleaner average contours obtained after averaging. The contours shown in Fig. 4.3(d) are obtained using nonmaximum suppression followed by hysteresis thresholding. In Fig. 4.4 we show similar results for the ETHZ shape dataset [22], indicating again that while starting from a set of unregistered images we can extract boundary and symmetry information for the whole category.

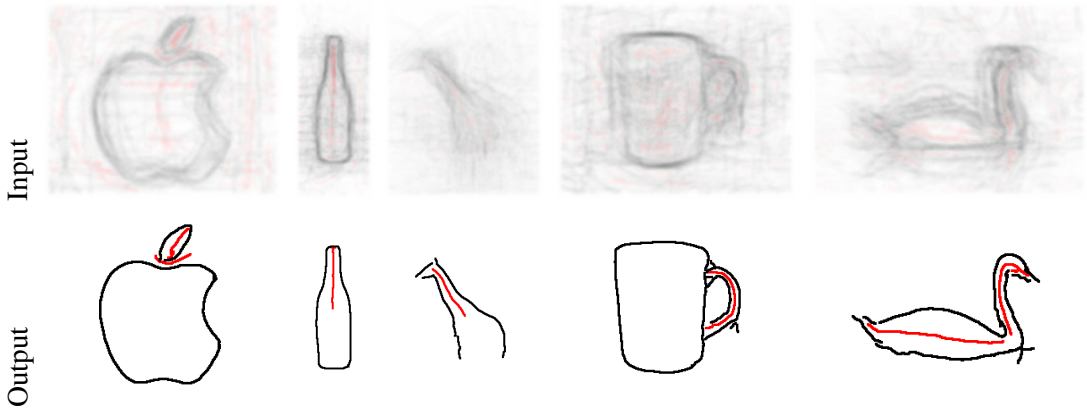


Figure 4.4: AAM registration results for the ETHZ shape dataset.

Finally, in Fig. 4.5 we show some results from on-going work on treating also the viewpoint associated with each image as a discrete latent variable. We optimize with respect to this discrete variable during AAM training, which automatically discovers view-specific models, or appearance clusters, such as the one- and two- floor buses. We have obtained similar results for other mostly rigid object classes, but for highly deformable categories, such as cats our approach currently fails. Interestingly, [23] have recently shown that minimal human annotation can be used to learn 3D active appearance models for highly non-rigid categories, making it conceivable that by some EM-based framework one can fully automate the learning pipeline of 3D deformable models.

Regarding the learning of piecewise-linear deformable models, some indicative results are shown in Fig. 4.6; in quantitative evaluations the results we obtained had turned out to be systematically better than those that we had attained with AAMs, validating that part-based models may be better for complex, articulated deformations. However, as the implementation of this system was too complicated to extend to object detection in cluttered images, we then turned in [2, 24, 25] to hierarchical models; these both made the inference task tractable and the shape-based representation easier to formulate and tune in a discriminative manner, as described in the following section.

Figure 4.5: Unsupervised view-specific AAM learning results on the PASCAL dataset; the models are shown as the average images and edge/ridge maps associated to each cluster after registration.

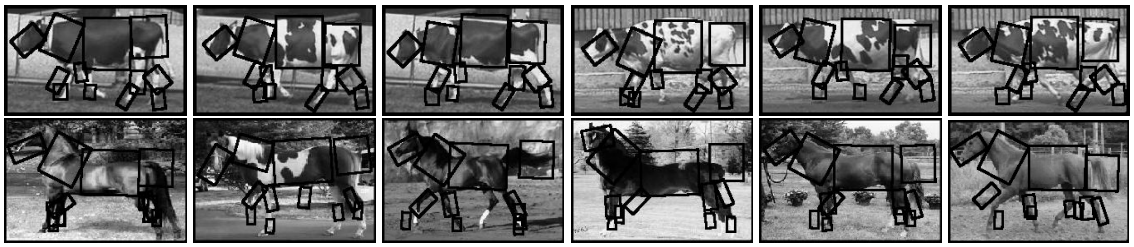
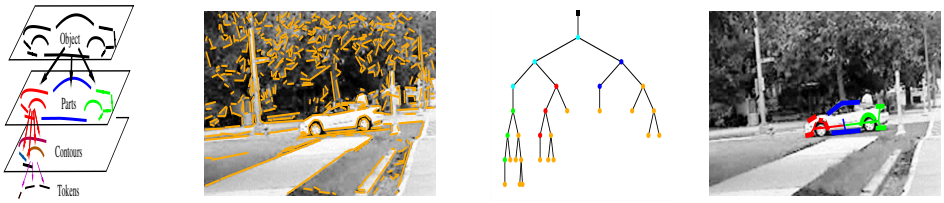


Figure 4.6: Matching results on horses and cows. For visualization only we have manually delineated the parts on the 'template' domain; the model parameters are learned with EM, and the image-specific deformations are estimated using Nonparametric Belief Propagation.

4.2 Learning hierarchical shape models

The problem of learning hierarchical models has gathered increased attention over the last years, as described in Section 1.2, based on arguments about their generalization ability, computational efficiency and representational power. Starting from [24] and continuing in [2] we have developed hierarchical shape models that start by grouping edge and ridge segments ('tokens') into contours,



(a) Object Model (b) Image Tokens (c) Parse Tree (d) Parsed Object

Figure 4.7: Hierarchical models and object parsing: we consider detecting objects by recursively grouping simple tokens (straight edge and ridge segments) extracted from the image. This amounts to building a *parse tree* that indicates how image tokens are composed to form objects. The leaves of this tree ('terminals') are edge/ridge tokens, and the color-coded nodes correspond to intermediate object structures; the root of the tree is the object, shown in (d).

then to parts, and finally to objects, as illustrated in Fig. 4.7. Such models provide extended spatial support for an object hypothesis by establishing relationships between image and model contours - this can make other tasks, such as segmentation, or grasping, easier. Furthermore, our hierarchical representation makes it possible to detect objects efficiently by using the A^* algorithm. We discuss the learning of such models in this section, and the efficient detection aspects in the following chapter.

In the following we will first describe how we recover the structure of these models in a data-driven manner, by building on the AAM learning work described above. We will then describe how one can estimate the parameters of such models in a discriminative setting, to improve performance on object detection tasks. We have applied our models to shape-based detection on the ETHZ dataset and obtained results that were the state-of-the-art at the time of publication.

Structure discovery

As in Section 4.1, we assume that we are provided with a small set of images (in the range of 20 to 50) that contain occlusions, noise, or illumination variations, while other than the object's bounding box we do not require manually-segmented features or landmarks. Our task is to recover a description of the object category in terms of a hierarchy of edge segments, contours, and parts.

Our starting point is the symmetry- and boundary-based description of our category that is provided to us by AAM learning. We exploit the geometric nature of this description to recover possible object parts in a data-driven manner. In particular, we first turn the boundary and symmetry contours into a set of short straight segments, and treat each as a node in a weighted graph as shown in Fig. 4.8, where the weights are determined based on contour continuity and symmetry. The graph is segmented into strongly connected components using Affinity Propagation, [26], which as shown in Fig. 4.8 and Fig. 4.9 typically delivers visually plausible parts that often correspond to semantic structures, such as handles, heads, and wheels.

In several cases the discovered parts are far from perfect; for instance, for all dataset splits the magenta group on the swan column has no semantic interpretation, but is rather a 'leftover' of the Affinity Propagation grouping. However, this decomposition only serves as an initialization for

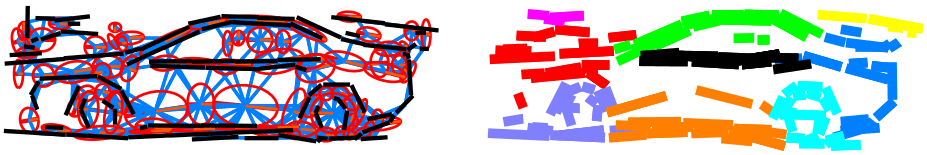


Figure 4.8: Part discovery through pairwise clustering: first the object contours are broken into straight segments, which are seen as nodes on a graph. Ridges are shown as ellipses whose width is proportional to the scale of the ridge. Nodes are connected with edges based on continuity and parallelism. The affinity among nodes is estimated using statistical and geometric information. The object parts shown on the left are obtained using Affinity Propagation.

our shape model, while its score function is then trained discriminatively in an end-to-end manner, as we now describe.

Discriminative model training

Having established the structure of our model, we now turn to learning its parameters; for this we first specify the form of its score function. We use a tree-structured graphical model with nodes $i \in V$ and edges $(i,j) \in E$; each node is connected to a single parent $pa(i)$ at the level above, and several children nodes $ch(i)$ at the level below. The graph has three levels – root node V_r, object parts V_p, and object contours V_c, while the contour nodes are connected to edge and ridge tokens, which serve as our observations $\mathbf{I}$. The position and scale of each node i is encoded in a pose vector $\mathbf{s}_i$. The probability of an object configuration $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_N)$ can be expressed by an exponential form:

$$P(\mathbf{S}) = \frac{1}{Z[\lambda]} \exp\left(-\sum_{i \in V} \phi_i(\mathbf{s}_i, \mathbf{s}_{pa(i)})\right), \quad \phi_i(\mathbf{s}_1, \mathbf{s}_2) = -\lambda_i \log P(\mathbf{s}_1|\mathbf{s}_2) \tag{4.8}$$

For a Bayesian network we would have $\lambda_i = 1,\ \forall i$ and $Z = 1$; but in our case we will learn the λ parameters discriminatively, and will ignore the partition function Z. The $P(\mathbf{s}_i|\mathbf{s}_{pa(i)})$ terms describe the distribution of a child's pose given the pose of its parent, and are set to normal distributions for simplicity. The expression in Eq. 4.8 acts like a prior distribution over configurations.

We further relate the pose of an object contour $\mathbf{s}_i$ to a group of image tokens $\mathbf{h}_i$ in terms of an observation potential, $\psi_i(\mathbf{I}_{\mathbf{h}_i}, \mathbf{s}_i)$, that lends itself to efficient computation using 'integral angles', as detailed in [2]. This potential is used to express our model's data-fidelity term in terms of the contour poses, the image tokens, and the assignment variables as follows:

$$P(\mathbf{I}|\mathbf{S}, \mathbf{H}) = \frac{1}{Z} \exp\left(\sum_{i \in V_c} -\psi_i(\mathbf{I}_{\mathbf{h}_i}, \mathbf{s}_i)\right). \tag{4.9}$$

We indicate missing parts at any level of the hierarchy with a binary variable $\mathbf{y}_i$ for every node i; when a node is missing we enforce its descendants to be missing as well and replace every related

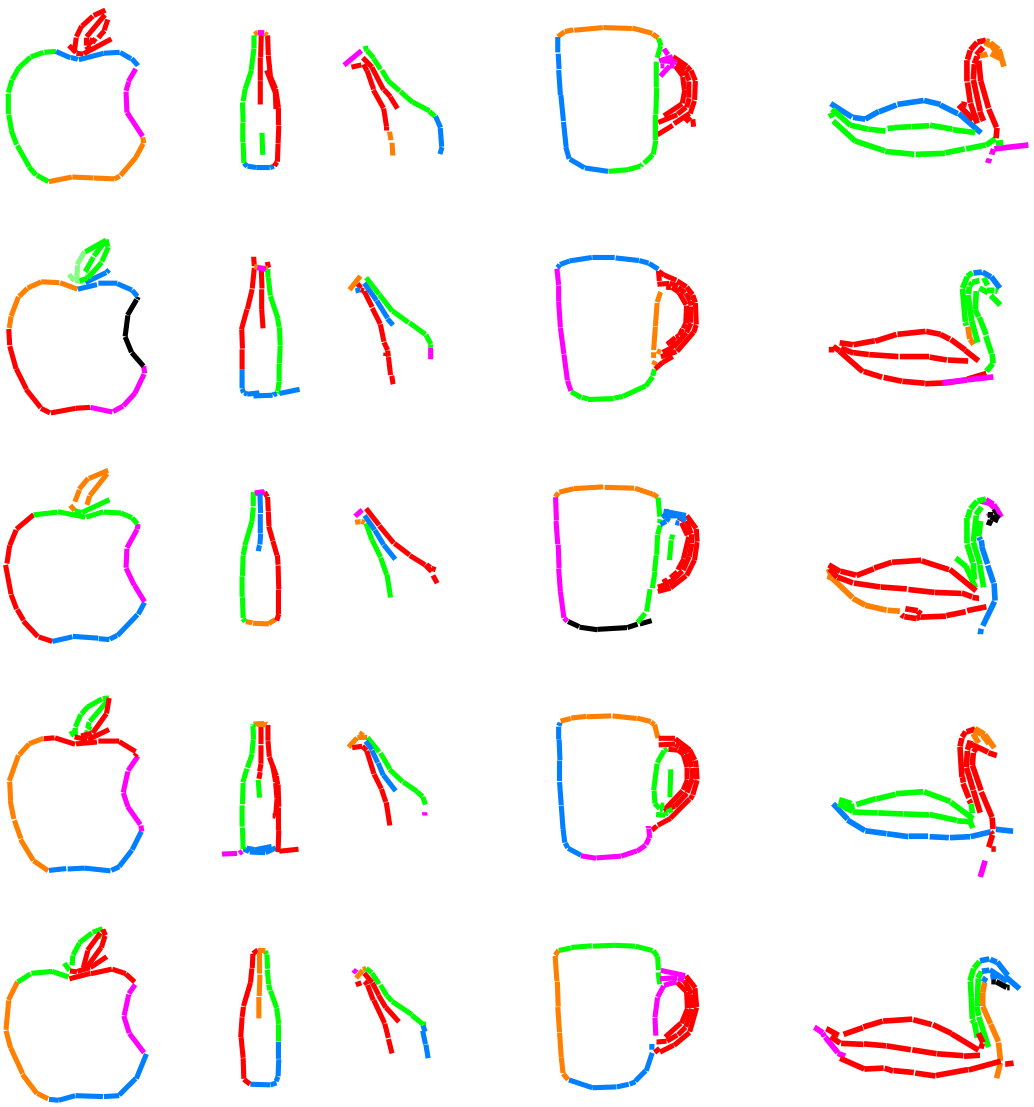


Figure 4.9: Object parts delivered by our method for the ETHZ shape categories, for five different training set splits: boundaries and symmetry axes belonging to the same object part are shown with the same color. *Please see in color.*

Figure 4.10: Positive and negative bags used by Multiple-Instance Learning (MIL) for the 'car' shape model: for each training image we compute a set of object instantiations, visualized as a part-level labelling of the image tokens. Through MIL we learn a classifier that accepts at least one instantiation per positive image and rejects all instantiations of a negative image.

summand with a 'missing' potential function $\phi_j^0 = -\log P(\mathbf{y}_j = 0)$. Combining terms we have:

$$P(\mathbf{I}, \mathbf{S}, \mathbf{H}, \mathbf{y}) = P(\mathbf{I}|\mathbf{S}, \mathbf{H}, \mathbf{y})P(\mathbf{S}, \mathbf{H}, \mathbf{y}) \propto \exp(-C(I, \mathbf{S}, \mathbf{H}, \mathbf{y})), \quad \text{where}$$
$$C(I, \mathbf{S}, \mathbf{H}, \mathbf{y}) = \sum_{i \in \{V_p, V_c\}} \left(\mathbf{y}_i \phi_i(\mathbf{s}_i, \mathbf{s}_{pa(i)}) + (1 - \mathbf{y}_i)\phi_i^0\right) + \sum_{i \in V_c} \mathbf{y}_i \psi_i(\mathbf{I}_{\mathbf{h}_i}, \mathbf{s}_i). \tag{4.10}$$

Equation 4.10 expresses the cost of a candidate configuration $(\mathbf{S}, \mathbf{H}, \mathbf{y})$ as a sum of terms that may not necessarily be commensurate; for instance in [2] we determine a parametric expression for ψ_i that facilitates easy computation, but the parameters are not easy to learn probabilistically. We therefore phrase the learning problem as one of finding a weighting of the summands in Eq. 4.10 that guarantees good detection performance, so that the cost in Eq. 4.10 will be low on shapes belonging to the object category, and high on negatives.

Our training problem is non-standard, as we do not know the correct object configurations; so even at training time we do not know the features that our score function should be using. We only know that for a positive image at least one configuration should be positive, while all configurations composed from a negative image should be negative. This can be addressed using Multiple Instance Learning (MIL) [27] by treating the correct object configuration as a latent variable, and forming a 'bag of features' per training image, as shown in Fig. 4.10: from each image we can form a set ('bag') of features, corresponding to all possible object instantiations. We want to train a classifier that will label at least one instance as positive for a positive image, and all instances as negative for a negative image.

We already considered MIL in Section 2.1 for the treatment of orientation and scale ambiguity when learning boundary and symmetry classifiers. What distinguishes our case now is that we are dealing with a substantially larger set of hidden variables, corresponding to all possible ways of parsing an image – in Fig. 4.10 we show only a small fraction of these possible ways. As detailed in [2, 25], we combine the Deterministic Annealing MIL of [28] with an iterative algorithm, similar to that of [8], to gradually acquire new instances and expand the positive and negative bags of the training set. As shown in Fig. 4.11 this process gradually improves the score function by learning to discriminate among car and car-like structures from the background, and yielding after several iterations a score function that is sharply peaked around the actual location of the car.

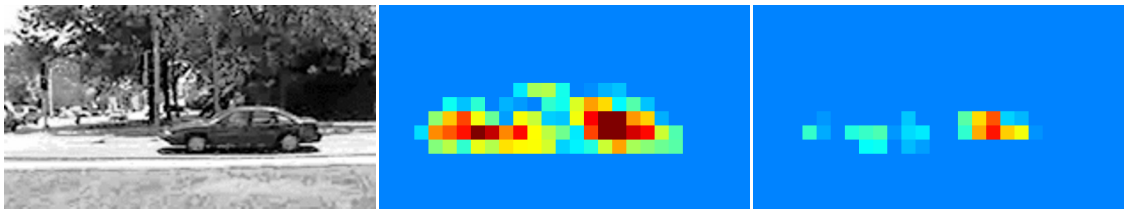


Figure 4.11: Improvement of the parsing cost function for cars: initially (middle) our model mistakes parallel structures for cars, giving low cost to the street region lying to the left of the car. After six iterations of MIL training (right), the cost function indicates more sharply the location of the car.

Results

We validate our method using the UIUC car [21] and the ETHZ shape classes [22]. For all classes we use common parameters and settings during both training and detection. For cars we use 50 images to learn the contours and object parts, and 300 positive and negative images for discriminative training. For the ETHZ classes we use the evaluation protocol in [5]: for each category we use half of its images for training, and the remaining images from that category, and all images of other categories, for testing; we present results averaged over 5 different splits. As negatives we use 300 images from the Caltech background images of [29].

We first show qualitative results of object parsing on these datasets in Fig. 4.12. We observe that our algorithm can deal with real images containing substantial clutter; for example, in the car images only a small fraction of the image tokens is used to build the object.

In Fig. 4.13 we report quantitative results on these benchmarks. On the top-left plot we see that our results on the UIUC dataset compare favorably to those of [30, 31], who use sparse image representations, while our Recall at Equal-Error-Rate (when precision equals recall) is 98% percent, equal to the one reported by Fidler et. al. in [35] with an edge-based representation. In the next plots we report results on the ETHZ dataset and compare to the boundary-based works of Ferrari et. al. [32, 33] and the region-based works Gu et. al. [34]. We plot the recall of our detector (ratio of detected objects) versus the number of false-positives-per-image (FPPI), averaged over the whole dataset and averaged over the 5 trials; we consider a bounding box as correct if its intersection-over-union with a ground truth bounding box is larger than 0.5. Our method outperforms the shape-based methods of Ferrari et. al. [32, 33], while comparing to the region-based method of Gu et. al. [34] we see that our method performs better on mugs and swans, equally well on bottles, slightly worse on apples and systematically worse on giraffes. This difference in performance is to some extent expected: our method can accurately model the outline of objects, which is distinctive for the categories where it performs well, while the regional cues used in [34] can more naturally capture the texture of giraffes.

Figure 4.12: Parsing results: For each image we show object instantiations that are classified as positive by our inference algorithm. We show the parse results at the object part-level, using color to indicate the object part to which a token is assigned; but please note that our algorithm establishes a finer, contour-level relation.

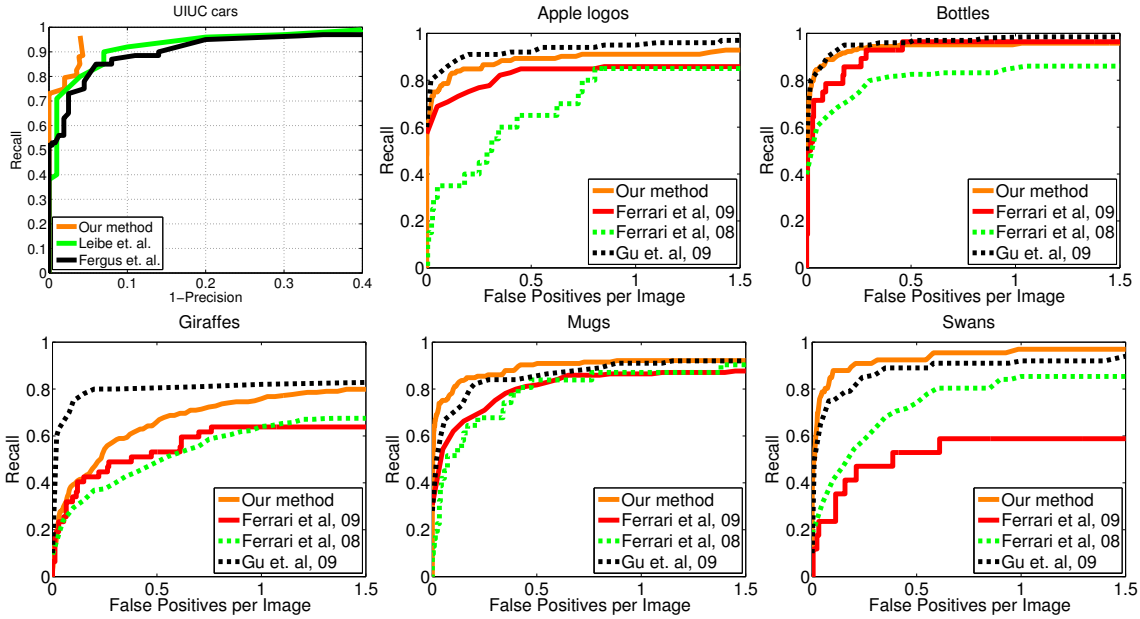


Figure 4.13: First plot: benchmark results on the UIUC dataset; we compare to the sparse, part-based approaches of Fergus [30] et. al. and Leibe et. al. [31]. Next five plots: Benchmark results on the ETHZ classes: comparisons with Ferrari et. al. [32, 33], and Gu et. al. [34].

4.3 Learning mid-level models of actions in videos

In [3] we apply several of the ideas developed in the previous two sections to understanding actions in videos; our goal is to address not only action recognition (*"What action?"*), as is common in the current literature, but also localization (*"Where in the video?"*), by 'parsing' a video into action parts. This suggests applying an analogous procedure of breaking up the input signal into 'mid-level' components through some generic front-end processing, and then learning a score function that guides the assembly of such components into model-based action interpretations.

For this, at the low level we decompose a video into a set of groups of moving points, and summarize each such group by descriptors that capture intensity, motion and appearance statistics. The quality of a candidate instantiation is expressed as the energy of a fully-connected MRF with nodes corresponding to action parts and labels indicating the group-part assignments. During training this score function is learned discriminatively, using Multiple Instance Learning to treat the association of 'groups' with 'parts' as a latent variable. During testing we use TRW-S [36] to efficiently match our model to a video. Our results indicate that apart from having competitive classification performance, our model establishes spatio-temporal part relationships that support more fine-grained decision tasks, such as action localization.

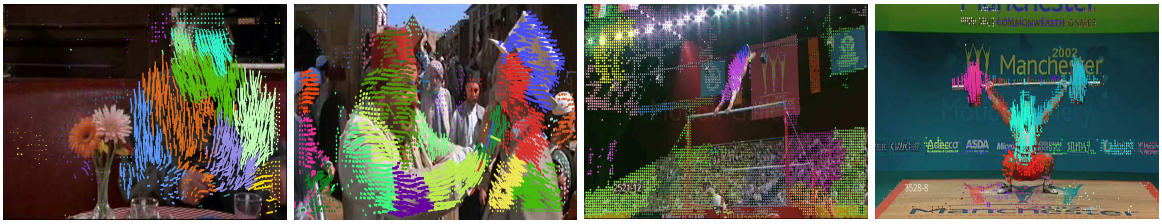


Figure 4.14: Examples of trajectory groups; each group has a distinct color.

Grouping-based video representation and features

Our front-end breaks a video into trajectory groups by combining ideas from tracklets [37, 38] and trajectory grouping [39]. Namely, after first extracting dense point trajectories we group theme based on pairwise similarity. For two trajectories $\{\mathbf{x}_a[t]\}_{t=\tau_a}^{T_a}$ and $\{\mathbf{x}_b[t]\}_{t=\tau_b}^{T_b}$ that coexist over an interval $[\tau_1, \tau_2]$ and are spatially close we define a distance of the form:

$$d(a,b) = \left(\max_{t \in [\tau_1, \tau_2]} |\mathbf{x}_a[t] - \mathbf{x}_b[t]|_2 \right) \left(\frac{1}{\tau_2 - \tau_1} \sum_{t=\tau_1}^{\tau_2} |\dot{\mathbf{x}}_a[t] - \dot{\mathbf{x}}_b[t]|_2 \right). \tag{4.11}$$

The first term is the spatial distance of the trajectory points and the second is the distance of the point velocity estimates. These distances are used to compute a trajectory affinity matrix $w(a,b) = \exp(-d(a,b))$ between all trajectory pairs, which is then used as input to a hierarchical clustering algorithm [40]. Some trajectory groups formed in this way are illustrated in Fig. 4.14. Even though many segments may not be semantically meaningful, we can practically always recover at least some segments that lie within a single object and move along with it.

For unary terms we associate with each such group k a statistical descriptor h_k formed by concatenating Histogram of Gradient (HoG), Histogram of optical Flow (HoF) [41] and Histogram of Motion Boundaries (HoMB) [38] descriptors. We do this efficiently by quantizing trajectory positions on a regular grid where feature extraction operations are efficiently performed in batch mode. We complement this statistical descriptor with a descriptor of the mean group trajectory $g_k[t]$, defined as the mean active tracklet position at any time instance. We thus associate each group k as a pair $G_k = \{h_k, g_k\}$. Finally, at the coarsest level we form a simple bag-of-words (BoW) representation, h_o, of all groups within the video. Consequently, a video S is described as the collection of the groups in combination with the histogram h_0: $S = \{h_o, \{G_k\}_{k=1}^N\}$.

For pairs of groups that coexist in time we also measure how their relative position evolves over time. We do this statistically by forming a histogram of relative trajectory motions, measuring first how the horizontal and vertical differences of two mean trajectories evolve, and then forming a joint histogram through a soft binning operation. This scheme avoids explicit decisions about the temporal structure of low-level data, in a manner similar to how SIFT avoids grouping by using histograms of gradients. We construct this histogram in a scale-sensitive manner to accommodate potential changes in video resolution; namely we construct different pairwise features $\psi(g_{p_i}, g_{p_j}, \sigma)$ for different values of scale σ, and let the inference algorithm pick the right σ.

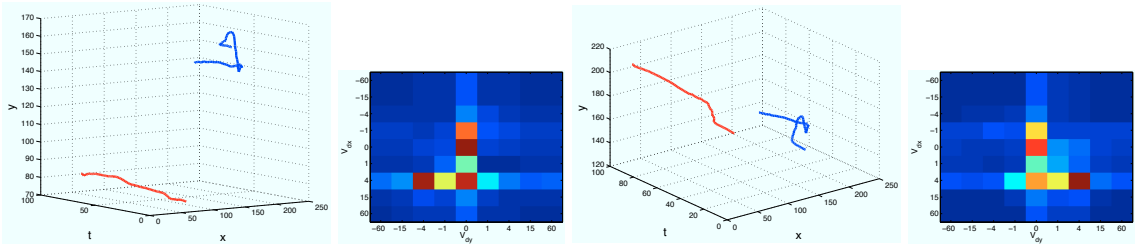


Figure 4.15: Pairs of group trajectories (left) and their corresponding pairwise descriptors (right).

Grouping-based action model

Our model measures the quality of a presumed action configuration in terms of an MRF score; in particular we use a fully connected graph $\mathcal{G} = (V, E)$, with each node $i \in V$ encoding a *part* and each edge $(i, j) \in E$ encoding pairwise *relations* between parts. An additional isolated node F represents the video as a whole. Given a video $\mathbf{x}$ that has been decomposed into N clusters, we consider a vector of discrete *latent variables* $P = [p_1, \ldots, p_{|V|}]$, with $p_i \in \{1, \ldots, N\}$ associating each node i with one of the N trajectory clusters. We also treat scale as a latent scale variable, σ, allowing us to chose the pairwise term scale that is most appropriate to the video at hand.

The score of a set of latent variables, $z = (P, \sigma)$ under an action model is given by:

$$\text{score}_{\mathbf{w}}(z, S) = \langle w_0, h_0 \rangle + \sum_{i=1}^{|V|} \langle w_i, h_{p_i} \rangle + \sum_{i=1}^{|V|} \sum_{j=i+1}^{|V|} \langle w_{i,j}, \psi(g_{p_i}, g_{p_j}, \sigma) \rangle \tag{4.12}$$

where $\mathbf{w} = \{w_0, w_i, w_{i,j}\}$, $i, j \in \{1, \ldots, N\}$ are the model parameters, and h_{p_i} and $\psi(g_{p_i}, g_{p_j}, \sigma)$ are the unary and pairwise features corresponding to the particular choice of latent variables. If the model parameters are known, optimizing with respect to $z = (P, s)$ can be done efficiently using approximate inference; Tree-Reweighted Message Passing (TRW-S) [36] converges in less than a second, and in most cases the duality gap is zero.

Coming to learning $\mathbf{w}$, we face again a Multiple Instance Learning problem: for positive images we should have $\text{score}_{\mathbf{w}}(z, S_p) > 0$ for at least one z, while for negatives we should have $\text{score}_{\mathbf{w}}(z, S_n) < 0$ for all z. We use a ranking-based training criterion, which gave notably improved results, in consistency with [42]. For initialization we set the pairwise weights $w_{i,j}$ to zero; we initialize the weights for the unary terms w_i by setting each w_i equal to the center of a cluster produced by running K-means on the set of vectors h_k belonging to positive examples.

Results

We validate our model on the Hollywood1 Human Action (HOHA) [41] and UCF-Sport [43] datasets. HOHA contains 430 videos and is most challenging, including substantial camera motion, rapid scene changes, clutter, and complex activities that involve actor and object interactions (e.g. 'get out of car', 'pick up phone', or 'kiss'). The UCF-Sports dataset consists of 150 sports videos captured in more constrained environments from sports broadcasts, and include actions

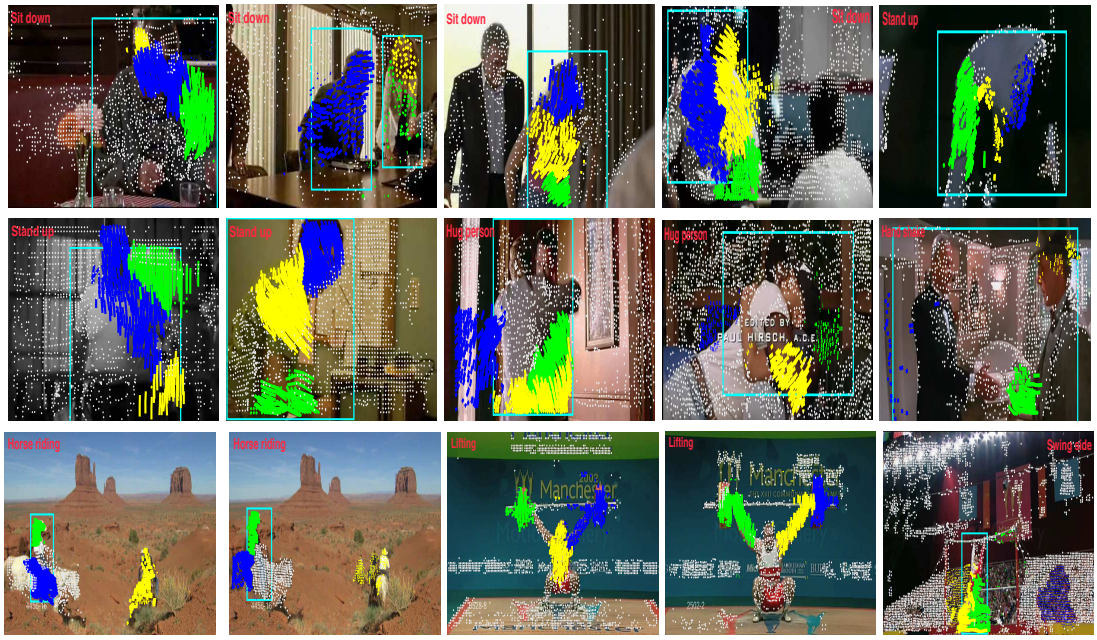


Figure 4.16: Sample frames from different video sequences of the test sets of the HOHA dataset (first two rows) and the UCF-Sports dataset (third row). Colored trajectories represent selected trajectory groups identified by our algorithm (color encodes part assignment), while white dots show trajectories that were not selected as parts. In cyan we show the ground-truth bounding boxes, which are used to assess localization accuracy.

such as 'weight-lifting' and 'swinging-bench', which however still pose challenges due to large displacements, clutter, and intra-class variability. Bounding boxes enclosing the person of interest at each frame were available for the UCF dataset while for HOHA we gathered these on our own.

Starting with qualitative results, we show in Fig. 4.16, and more extensively in [44] some action parsing results of our algorithm on the HOHA and UCF datasets; we note that our algorithm not only provides an action label for a video, but also indicates the spatio-temporal support of the action. Trajectory groups selected by our algorithm typically lie within the manually annotated bounding boxes (in cyan) which suggests that our method selects meaningful groups as parts.

To quantify this claim, we measure a localization score defined as the ratio of selected contours that have substantial overlap with the ground-truth bounding box(es):

$$O(D,L) = \frac{1}{|V| \cdot T} \sum_{i=1}^{|V|} \sum_{t=1}^{T} [\frac{|D_{i,t} \cap L_t|}{|D_{i,t}|} \geq \theta], \tag{4.13}$$

where L_t is the set of points inside the annotated bounding box, $[\cdot]$ indicates if $\cdot$ is true, $D_{i,t}$ is the set of points belonging to the selected trajectory group, and θ is a threshold. Fig. 4.17 illustrates the average localization score across the test videos of each action as a function of θ, as well as

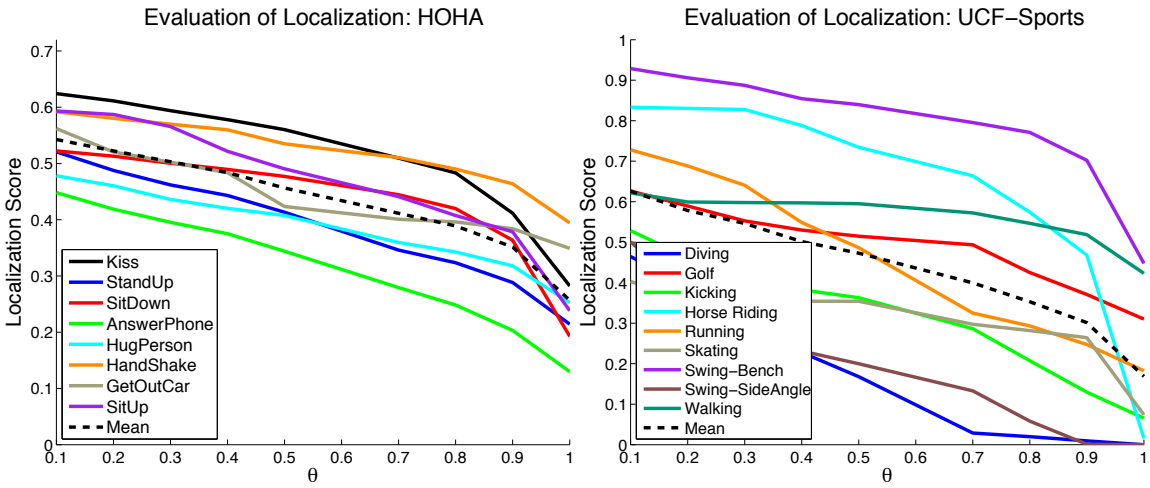


Figure 4.17: Localization scores for the trajectory groups selected by our algorithm as a function of the overlap threshold (θ) for the HOHA dataset (left) and the UCF-Sports dataset (right).

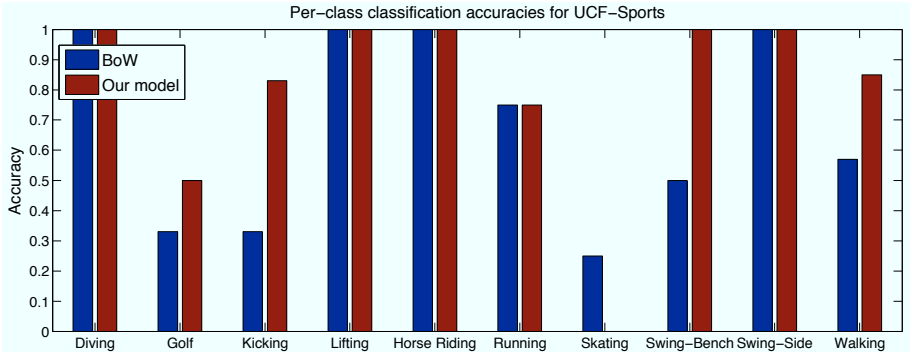
Table 4.1: Performance comparison on HOHA dataset.

Class	**Our Model**	Laptev *et al.* [41]		Yeffet *et al.* [46]	Raptis *et al.* [37]	Matikainen *et al.* [47]	Kläser *et al.* [48]	Sun *et al.* [45]		Shandong *et al.* [49]
		Single	Combined		BoW	BoW	BoW	TTD Combined	TTD-SIFT Combined	BoW
Answer phone	29.5%	26.7%	32.1%	35.1%	26.7%	35.0%	18.6%			**48.3**%
Get out of car	**51.0**%	22.5%	41.5%	32.0%	28.1%	7.7%	22.6%			42.3%
Hand shake	35.4%	23.7%	32.3%	33.8%	18.9%	5.3%	11.8%			**46.2**%
Hug person	30.8%	34.9%	40.6%	28.3%	25.0%	23.5%	19.8%	N/A	N/A	**49.3**%
Kiss	58.4%	52.0%	53.3%	57.6%	51.5%	42.9%	47.0%			**63.6**%
Sit down	38.4%	37.8%	38.6%	36.2%	23.8%	13.6%	32.5%			**47.5**%
Sit up	18.9%	15.2%	18.2%	13.1%	23.9%	11.1%	7.0%			**35.1**%
Stand up	58.0%	45.4%	50.5%	58.3%	**59.1**%	42.9%	38.0%			47.3%
MAP	40.1%	32.9%	38.4%	36.8%	32.1%	22.8%	24.7%	30.3%	44.9%	**47.6**%

the mean localization score across all actions of the two datasets. For $\theta = 0.5$ we get average localization scores of 48.4% and 47.3% for HOHA and UCF-Sports, respectively.

Turning to action classification results, in Table 4.1 we compare the average precision (AP) of other state-of-the-art techniques to that of our approach on the HOHA dataset. When using a baseline that includes only our BoW video representation coupled with an SVM with RBF χ^2 kernel, we obtain a mean AP of 33.4%; adding the pairwise terms drives the performance up to the reported mean AP of 40.1% (a more detailed analysis of the contributions of each aspect of our model is contained in [3]). Our approach is competitive with most schemes and performs better than [41] who use similar low-level features (HoG, HoF). The performance of our method is lower than the multi-kernel learning (MKL) approach of [45] and the recent work of [49], but our scheme could be extended to incorporate the features of [49] or be combined with MKL to improve classification performance. These aspects are in a sense orthogonal to the ability to localize actions, which is one of the main assets of our model.

For the UCF-Sports dataset, the mean per-class classification accuracies are summarized in



Method	BoW	Our Model	Lan *et al*. [50]
Accuracy	67.4%	**79.4**%	73.1%

Figure 4.18: UCF-Sports dataset results: the top row shows per-class classification accuracy, and the bottom row reports mean per-class classification accuracies.

Fig. 4.18; again, we notice a significant improvement over the bag-of-words baseline in most actions, and attain a substantially higher mean accuracy than [50].

Bibliography

[1] I. Kokkinos and A. Yuille, "Unsupervised learning of object deformation models," in *Proc. ICCV*, 2007.

[2] I. Kokkinos and A. Yuille, "Inference and Learning with Hierarchical Shape Models," *Int.l Journal of Computer Vision*, vol. 93, pp. 201–225, 2 2011.

[3] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *Proc. CVPR*, 2012.

[4] B. Frey and N. Jojic, "Transformation-Invariant Clustering Using EM," *IEEE Trans. PAMI*, vol. 25, no. 1, pp. 1–17, 2003.

[5] V. Ferrari, F. Jurie, and C. Schmid., "Accurate object detection with deformable shape models learnt from images," in *Proc. CVPR*, 2007.

[6] T. Jiang, F. Jurie, and C. Schmidt, "Learning shape prior models for object matching," in *Proc. CVPR*, 2009.

[7] Y. Wu, Z. Shi, C. Fleming, and S. C. Zhu, "Deformable template as active basis," in *Proc. ICCV*, 2007.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. CVPR*, 2008.

[9] T. Cootes, G. J. Edwards, and C. Taylor, "Active Appearance Models," in *Proc. ECCV*, 1998.

[10] T. Vetter, M. Jones, and T. Poggio, "A bootstrapping algorithm for learning linear models of object classes," in *Proc. CVPR*, 1997.

[11] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, R. P. Wurtz, and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture," *IEEE Trans. Computers*, vol. 42, no. 3,

[12] P. Felzenszwalb and D. Huttenlocher, "Efficient matching of pictorial structures," in *Proc. CVPR*, 2000.

[13] M. Welling, M. Weber, and P. Perona, "Unsupervised learning of models for recognition," in *Proc. ECCV*, 2000.

[14] S. Yu, M. Black, and Y. Yacoob, "Cardboard people: a parametrized model of articulated motion," in *Proc. CVPR*, 1996.

[15] L. Ladicky, P. H. S. Torr, and A. Zisserman, "Latent SVMs for human detection with a locally affine deformation field," in *BMVC*, 2012.

[16] A. Dempster, N. Laird, and D. Rudin, "Maximum Likelihood from Incomplete Data via the EM algorithm," *Journal of The Royal Statistical Society, Series B*, 1977.

[17] R. Neal and G. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse and Other Variants," in *Learning in Graphical Models*, M. Jordan, Ed., 1998.

[18] S. Baker, I. Matthews, and J. Schneider, "Automatic Construction of Active Appearance Models as an Image Coding Problem," *IEEE Trans. PAMI*, vol. 26, no. 10, pp. 1380–1384, 2004.

[19] D. Comaniciu and P. Meer, "Mean shift: A Robust Approach toward Feature Space Analysis," *IEEE Trans. PAMI*, vol. 24, no. 5, pp. 603–619, 2002.

[20] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky, "Efficient multiscale sampling from products of Gaussian mixtures," in *Proc. NIPS*, 2003.

[21] S. Agarwal and D Roth, "Learning a sparse representation for object detection," in *Proc. ECCV*, 2006.

[22] V. Ferrari, T. Tuytelaars, and L. V. Gool, "Object detection by contour segment networks," in *Proc. ECCV*, 2006.

[23] T. J. Cashman and A. W. Fitzgibbon, "What Shape Are Dolphins? Building 3D Morphable Models from 2D Images," *IEEE Trans. PAMI*, vol. 35, no. 1, pp. 232–244, 2013.

[24] I. Kokkinos and A. Yuille, "HOP: Hierarchical Object Parsing," in *Proc. CVPR*, 2009.

[25] ——, "Inference and learning with hierarchical compositional models," in *Stochastic Image Grammars Workshop*, 2009.

[26] B. Frey and D. Dueck, "Clustering by Passing Messages Between Data Points," *Science*, vol. 315, pp. 972–976, 2007.

[27] T. G. Dietterich, R. H. Lathrop, and T. Lozano-perez, "Solving the Multiple-Instance Problem with Axis-Parallel Rectangles," *Artificial Intelligence*, vol. 89, pp. 31–71, 1997.

[28] P. Gehler and O. Chapelle, "Deterministic annealing for multiple instance learning," in *Proc. AISTATS*, 2007.

[29] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. CVPR*, 2003.

[30] ——, "A sparse object category model for efficient learning and exhaustive recognition," in *Proc. CVPR*, 2005.

[31] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Proc. ECCV*, SLCV workshop, 2004.

[32] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of Adjacent Contour Segments for Object Detection," *IEEE Trans. PAMI*, vol. 30, no. 1, pp. 36–51, 2008.

[33] V. Ferrari, F. Jurie, and C. Schmid, "From Images to Shape Models for Object Detection," *Int.l Journal of Computer Vision*, vol. 87, pp. 284–303, 2009.

[34] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," in *Proc. CVPR*, 2009.

[35] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: learning a hierarchy of parts," in *Proc. CVPR*, 2007.

[36] V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. PAMI*, vol. 28, no. 10, pp. 1568–1583, 2006.

[37] M. Raptis and S. Soatto, "Tracklet descriptors for action modeling and video analysis," in *Proc. ECCV*, 2010.

[38] H. Wang, A. Klaser, C. Schmid, and C. Liu, "Action recognition by dense trajectories," in *Proc. CVPR*, 2011.

[39] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Proc. ECCV*, 2010.

[40] S. Tabatabaei, M. Coates, and M. Rabbat, "Ganc: greedy agglomerative normalized cut," *Arxiv preprint arXiv:1105.0974*, 2011.

[41] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. CVPR*, 2008.

[42] M. B. Blaschko, A. Vedaldi, and A. Zisserman, "Simultaneous object detection and ranking with weak supervision," in *Proc. NIPS*, 2010.

[43] M. Rodriguez, J. Ahmed, and M. Shah, "Action MACH: a spatio-temporal maximum average correlation height filter for action recognition," in *Proc. CVPR*, 2008.

[44] M. Raptis, *Action recognition from mid-level video representations, code and benchmark*, `http://vision.ucla.edu/~raptis/action_parts.html`, 2012.

[45] J. Sun, X. Wu, S. Yan, L. Cheong, T. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Proc. CVPR*, 2009.

[46] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *Proc. ICCV*, 2009.

[47] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: action recognition through the motion analysis of tracked features," in *ICCV workshop on Video-oriented Objected and Event Classification*, 2009.

[48] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. BMVC*, 2008.

[49] W. Shandong, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories," in *Proc. ICCV*, 2011.

[50] T. Lan, Y. Wang, and G. Mori, "Discriminative figure-centric models for joint action localization and recognition," in *Proc. ICCV*, 2011.

Chapter 5

Efficient Optimization of Shape-based Models

In this chapter we develop efficient optimization algorithms for inference with deformable, shape-based models; the main asset of these algorithms is that we either do not compromise accuracy at all, or at least can control the tradeoff between efficiency and accuracy. Another appealing aspect is that the idea of combining 'bottom-up' and 'top-down' computation translates into concrete aspects of these algorithms.

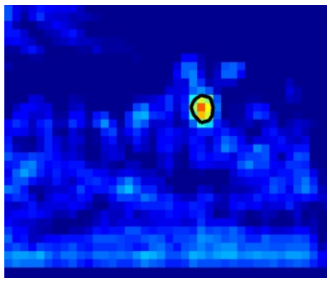
We start in Section 5.1 with our work [1–3] on accelerating detection with deformable part models (DPMs) [4–6] using Branch-and-Bound; in the best case this reduces the complexity of the 'part combination' stage of DPMs from linear to logarithmic in the number of pixels. We also alleviate the 'part computation' bottleneck by constructing efficient probabilistic upper bounds to the part scores, which we then combine with both Branch-and-Bound and the DPM cascades of [6]. We then present A^* parsing for the hierarchical models of Section 4.2 [7, 8], which again involves using score bounds to avoid unnecessary bottom-up computation. We conclude with our work [9] on parsing building facades with shape grammars that can recursively produce an arbitrary number of structures, such as floors, or windows. We use Reinforcement Learning to deal with structure variation and describe how to use bottom-up information to accelerate top-down grammar fitting.

5.1 Branch-and-Bound for Deformable Part Models

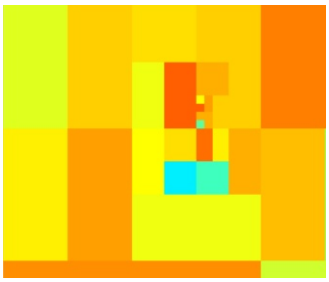
Our approach is based on the observation that an object detector should score high only on a small part of the image domain, presumably where the objects are contained. This is illustrated in the first two images of Fig. 5.1: in (a) we show the detection delivered by the DPM-based bicycle detector of [6] and in (b) its score $S(x)$ evaluated over the whole image domain. The part of the image that scores above a conservative threshold of -1 is encircled by a black contour in (b), which is indeed only a tiny image fraction. Based on this observation we can speed up detection by avoiding to exactly evaluate $S(x)$ at positions where its anticipated score is low.

Branch-and-Bound (BB) is an optimization algorithm that does exactly this: the 'anticipated score' of $S(x)$ in X is computed in terms of an upper bound, $\overline{S}(X)$, which is then used to guide search. We can use BB to find $\arg\max_x S(x)$ with the following scheme: we start from an interval containing all possible object locations, and then iteratively (a) split in two ('branch') the currently explored interval and (b) bound the resulting intervals before deciding where to move next. We stop when X is a singleton, which is guaranteed to be the strongest location. The operation of BB is illustrated in Fig. 5.1 (c), where we show as heat maps the upper bounds of the intervals visited

(a) Input & Detection result



(b) Detector score, $S(x)$



(c) Intervals & bounds used by BB to find $\{\arg\max_x S(x)\}$

Figure 5.1: Motivation for Branch-and-Bound based detection: instead of evaluating the DPM classifier's score exhaustively at all image locations (b), our method, shown in (c) devotes computational resources to promising intervals, as assessed by upper bounding the DPM score.

by our algorithm until convergence. Fine-grained image parts correspond to image locations that seem promising and are thus explored more. Coarse-grained parts have smaller upper bounds, and are thus not refined. We note that unlike pruning-based works [6, 10–13], our method does not sacrifice accuracy and is provably exact [2].

Bounding the DPM score function

As outlined in Section 5.1, DPMs use a score function that exploits both appearance and geometric information for detection; a configuration $\mathbf{x} = (x_0, \ldots, x_P)$ is scored as follows:

$$M(\mathbf{x}) \doteq \sum_{p=0}^{P} m_p(x_p, x_0) = \sum_{p=0}^{P} \underbrace{\langle w_p, H(x_p) \rangle}_{U_p(x_p)} + \underbrace{-\left(x_p - x_0 - \mu_p\right)^T I_p \left(x_p - x_0 - \mu_p\right)}_{B_p(x_p, x_0)}, \quad (5.1)$$

where the unary term $U_p(x_p)$ scores the image observations $H(x_p)$ under the appearance model of node p, and the pairwise term $B_p(x_p, x_0)$ enforces that the relative pose $x_p - x_0$ of parts p and 0 stay close to μ_p; in particular $I_p = \mathrm{diag}(H_p, V_p)$, with $H_p > 0, V_p > 0$.

To score a putative object location x we maximize over all configurations that place the root at x; this gives a score in the form of a sum of 'message' functions, $m_p(x)$:

$$S(x) \doteq \max_{\mathbf{x}: x_0 = x} M(\mathbf{x}) = \sum_{p=0}^{P} \underbrace{\max_{x'} \left(U_p(x') + B_p(x', x) \right)}_{m_p(x)}. \quad (5.2)$$

Branch-and-Bound requires an upper bound $\overline{S}(X)$ to the DPM score $S(x)$ within an interval X:

$$\max_{x \in X} S(x) \leq \overline{S}(X). \quad (5.3)$$

In order to construct this bound we use the following inequality:

$$\max_{x \in X} f(x) + g(x) \leq \max_{x \in X} f(x) + \max_{x \in X} g(x). \quad (5.4)$$

According to Eq. 5.2 we have $S(x) = \sum_p m_p(x)$, so applying this inequality gives:

$$\max_{x \in X} S(x) \leq \sum_p \max_{x \in X} m_p(x). \tag{5.5}$$

This means that we can separately bound the individual part contributions over X, say by $\overline{m}_p(X)$, and construct a valid bound $\overline{S}(X)$ of our score function by adding up the resulting bounds:

$$\max_{x \in X} S(x) \leq \sum_p \max_{x \in X} m_p(x) \leq \sum_p \overline{m}_p(X) \doteq \overline{S}(X). \tag{5.6}$$

To bound in turn $\max_{x \in X} m_p(x)$ we apply Eq. 5.4 using the definition of $m_p(x)$ in Eq. 5.2:

$$\max_{x \in X} m(x) = \max_{(x,x') \in (X \times X')} U(x') + B(x', x) \leq \max_{x' \in X'} U(x') + \max_{(x,x') \in (X \times X')} B(x', x) \doteq \overline{m}(X), \tag{5.7}$$

where we have dropped the p subscript for simplicity. Naively computing the term on the inequality's left hand side would require $|X| \cdot |X'|$ operations, with $|\cdot|$ denoting cardinality; but the right hand side term is composed of two easily computable terms: the first can be computed on a binary tree with fine-to-coarse optimization and the second term can be analytically evaluated for any pair of rectangular domains X, X' in a constant number of operations [3].

Having demonstrated how to bound the score efficiently, we now turn to controlling the bound's tightness; to do this we can break the domains X and X' into smaller subdomains over which the relaxations in Eq. 5.6 and Eq. 5.7 are not too loose - in the limit, for singletons, the bound equals the score. But as the intervals become smaller, their number increases. There is thus a tradeoff between the bound's tightness and the number of operations required to compute it.

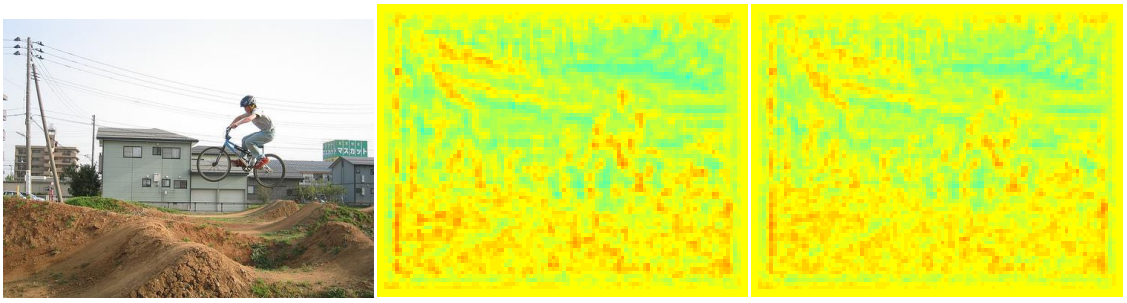
To keep the latter under control, we employ a 'Dual Recursion' algorithm inspired by the Dual Trees of [14], which amounts to starting from coarse part and root intervals, and refining them simultaneously. In [2] we prove that one can use conservative pruning, ensuring that we get exact results while keeping the computation tractable. This pruning scheme is empirically shown to keep the number of operations roughly independent of the interval's resolution. As these ideas were also present, but formulated in a different setting in [14], we call our method Dual Tree Branch-and-Bound (DTBB).

By the binary search nature of Branch-and-Bound in the best case we can find the best-scoring position in $O(P \log N)$ time, where P is the number of parts and N the number of pixels; similarly, detecting all objects above a threshold can be done in $O(MP \log N)$ iterations, where M is the number of objects above threshold. For low values of M this can be much smaller than the $O(PN)$ complexity of [15, 16]. This is also empirically validated by the results presented later.

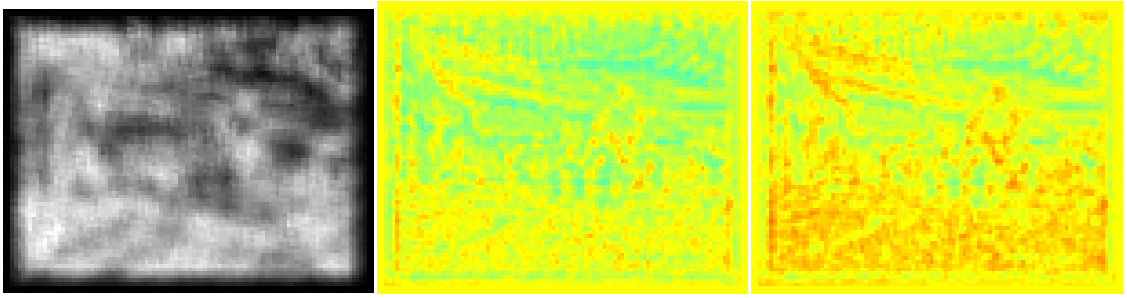
Probabilistic bounds for the part scores

Having described how to accelerate the part combination stage of DPMs, the remaining bottleneck is the computation of the part scores, $U_p(x) = \langle H(x), w_p \rangle$. For a 6×6 part filter, this involves 36 smaller inner products between 32-dimensional HOG cells and their respective weight vectors:

$$s[x] = \sum_y \langle \mathbf{w}_y, \mathbf{h}_{x+y} \rangle, \tag{5.8}$$



(a) Input image (b) Exact part score, $s[x]$ (c) Lookup-based approximation, $\hat{s}[x]$



(d) Approximation variance $m[x]$ (e) Lower bound, $p_e = .05$ (f) Upper bound, $p_e = .05$

Figure 5.2: Illustration of part score approximation and bounding: our goal is to rapidly bound the value of the part score $s[x]$ shown in (b). The bound we propose in Eq. 5.12 is formed in terms of two quantities, the lookup-based approximation $\hat{s}[x]$ of Eq. 5.11, shown in (c) and the approximation error variance, m_x of Eq. 5.13, shown in (d). These two are combined as in Eq. 5.14 to form an interval that contains the actual value with a certain probability of error p_e. The values of the lower and upper bounds for $p_e = 0.5$ are visualized in (e) and (f) respectively.

where $y \in [0, 5] \times [0, 5]$ so we have more than 1000 multiplications and summations per x. A simple method of accelerating this operation is to use vector quantization, as in [17]; this involves an offline stage where we construct a codebook $\mathcal{C} = \{C_1, \ldots, C_K\}$ for $\mathbf{h}$ and then form an array:

$$\Pi[k, y] = \langle C_k, \mathbf{w}_y \rangle, \tag{5.9}$$

of precomputed scores. To approximate Eq. 5.8 at test time, we first vector quantize every HOG cell $\mathbf{h}_x$, obtaining $I[x] = \operatorname{argmin}_k d(C_k, \mathbf{h}_x)$, and then exchange the $36 \cdot 32$ multiplications and summations of Eq. 5.8 with 36 lookup and summation operations as follows:

$$\langle \mathbf{h}_{x+y}, \mathbf{w}_y \rangle \simeq \langle C_{I[x+y]}, \mathbf{w}_y \rangle = \Pi[I[x + y], y] \tag{5.10}$$

$$s[x] \simeq \hat{s}[x] = \sum_y \Pi[I[x + y], y]. \tag{5.11}$$

This approximation could in principle result in a 32-fold acceleration, but in practice cache misses reduce it to being only 5- to 10- fold.

Our contribution lies in alleviating the drop in performance incurred by this approximation, reported in [17] and also validated by our results. For this, we propose to avoid taking $\hat{s}$ at face value, but rather to work with an interval of values that contains s with high probability. For this we treat the approximation error $e[x] \doteq s[x] - \hat{s}[x]$ as a random variable and construct probabilistic upper and lower bounds to $s[x]$ in terms of $\hat{s}[x]$. We do this by using Chebyshev's inequality [18], which ensures that a zero-mean random variable X with second moment $V = E\{X^2\}$ satisfies:

$$P(|X| > \alpha) \leq \frac{V}{\alpha^2}. \tag{5.12}$$

This means that with probability larger than V/α^2, X will be contained in $[-\alpha, \alpha]$, or equivalently X will be contained in $[-\sqrt{V/p_e}, \sqrt{V/p_e}]$ with probability of error smaller than p_e.

Coming to our case, as detailed in [3], we can estimate the second moment of $e[x]$ as

$$m[x] = \sum_y \frac{1}{32} \|\mathbf{e}_{x+y}\|^2 \|\mathbf{w}_y\|^2, \tag{5.13}$$

where $\mathbf{e}_x$ is the L_2 norm of the quantization error at x and $\|\mathbf{w}_y\|$ the L_2 norm of the weight vector corresponding to cell y. The $\|\mathbf{w}\|$ terms can be precomputed, while computing the $\|\mathbf{e}\|$ terms is amortized, i.e. done once per image, and then reused by all objects and parts. This means that practically we can estimate $m[x]$ with 36 multiplication-summations per position x.

Having at our disposal the lookup-based estimate $\hat{s}[x]$ and the second moment $m[x]$ of the approximation error, we can now form probabilistic upper and lower bounds to $s[x]$: since $e[x] = s[x] - \hat{s}[x]$, according to Chebyshev's inequality $s[x]$ will be bounded between:

$$\hat{s}[x] - \sqrt{\frac{m[x]}{p_e}} \leq s[x] \leq \hat{s}[x] + \sqrt{\frac{m[x]}{p_e}}, \tag{5.14}$$

with probability $1 - p_e$. We can understand this bound as constructing a 'buffer' around $\hat{s}[x]$ to make up for the approximations involved; the width of this buffer depends on the magnitude of the approximations, captured by m, and how conservative we want to be, determined by the probability of error, p_e. In Fig. 5.2 we show all of the quantities involved in Eq. 5.14 for $p_e = 0.05$; we have empirically verified that this bound is valid across a broad range of values of p_e.

We have integrated this bound with both our Branch-and-Bound approach and the Cascade-DPMs of [6]; in both cases the bounds are used in a preliminary stage that 'shortlists' promising positions. Our algorithm eventually computes the inner product in Eq. 5.8, but sparingly, only around shortlisted positions. This can be understood as using a coarse bottom-up processing (upper bound) to 'activate' object hypotheses in a first stage; in a second stage these shortlisted hypotheses provide top-down based guidance about where to perform some more time-demanding bottom-up computations (inner product). As demonstrated by our results, this bottom-up/top-down computation scheme can result in substantial computational savings with no impact on detection accuracy.

Results

In our experimental validation we use the same models as [6, 19] and do not address learning; our concern is therefore only the exactness and speed of the optimization.

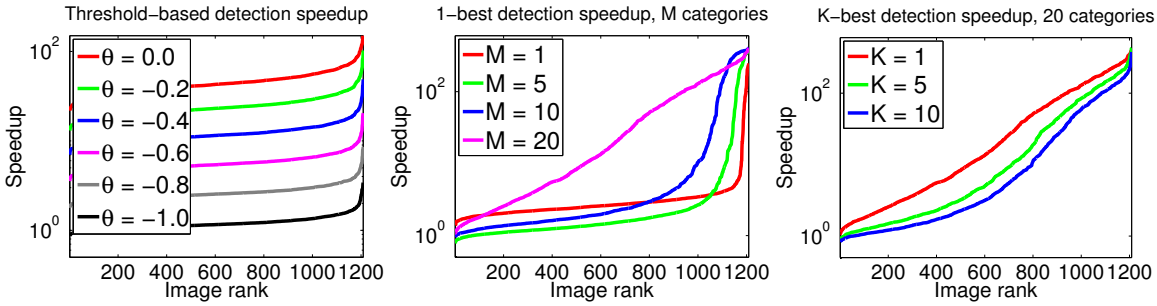


Figure 5.3: Acceleration of the 'part combination' stage using our Branch-and-Bound scheme versus Generalized Distance Transforms [15] on images from the Pascal dataset: (a) Single-object speedup for different thresholds (b) Multi-object speedup, for different numbers of classes (c) Multi-object speedup, for different numbers of retrieved objects.

Starting with the validation of the Branch-and-Bound algorithm that uses the exact part scores, the results we get after part combination are identical to those of [19] other than differences due to floating/double point arithmetic. We therefore do not provide any detection performance curves, but only timing results.

As a first experiment we consider the standard detection scenario where we want to find all objects in an image having score above a certain threshold. We show in Fig. 5.3 (a) how the threshold of our detector affects the speedup we obtain: for 1200 images we calculate the acceleration of our BB-based algorithm over the Generalized Distance Transform (GDT)-based baseline of [15], and then sort the resulting accelerations to produce smooth curves - so images are ranked by acceleration. We observe that for a conservative threshold, $\theta = -1$ or $\theta = -.8$, the speedup is moderate, but as we become less conservative the acceleration can become 10-fold, or even 20-fold.

As a second application we consider the problem of identifying the 'dominant' object present in the image, i.e. the category that gives the largest score. In plots Fig. 5.3 (b),(c) we show results on the Pascal dataset using a similar format as in (a). We compare the time that would be required by GDT to perform detection of all 20 categories considered in Pascal, to that of BB-based search over both positions an categories. In (b) we show the acceleration attained for finding the single-best result as the number of objects (M) increases; we see that for more categories the gains increase, indicating that the cost of our algorithm grows sublinearly in categories. The justification for this is that if many categories are involved, it becomes more likely that one of them will have a high score, 'pop-up' and terminate search. In (c) we show how the 'k' in 'k-best' affects the speedup. For small values of k the gains are more pronounced, and can be more than 100-fold.

Turning to the evaluation of part bounds, there is a certain slack due to the probabilistic nature of the bound, so we need to explore the impact that this may have on object detection accuracy. In Fig. 5.4 we provide precision-recall plots for bicycle detection, for both Branch-and-Bound and cascaded detection (similar results are obtained for all classes). In all cases 'exact' refers to results

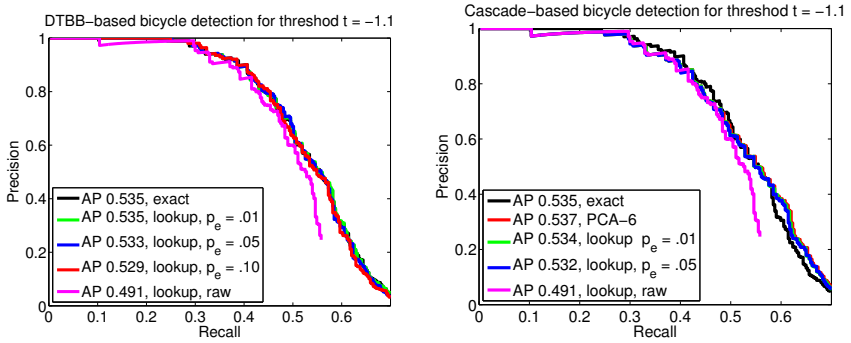


Figure 5.4: Precision-Recall curves for bicycle detection using branch-and-bound (left) and cascaded detection (right) with our lookup-based bounds. Having a small probability of error, p_e, ensures that we get virtually identical results to the exact, convolution-based method.

Exact (non-cascade) Detection				
	GDTs	DTBB	$p_e = 0.05$	$p_e = 0.01$
Part terms	2.35 ± 0.77	1.69 ± 0.18	0.69 ± 0.03	0.69 ± 0.06
$\theta = -0.5$	0.60 ± 0.05	0.21 ± 0.06	0.47 ± 0.11	1.04 ± 0.25
Sum	2.95 ± 0.82	1.90 ± 0.23	1.17 ± 0.12	1.74 ± 0.32
$\theta = -0.7$	0.60 ± 0.05	0.42 ± 0.10	1.00 ± 0.23	2.33 ± 0.65
Sum	2.95 ± 0.82	2.10 ± 0.24	1.70 ± 0.27	3.00 ± 0.71
$\theta = -1.0$	0.60 ± 0.05	1.31 ± 0.31	3.80 ± 0.90	9.40 ± 2.70
Sum	2.95 ± 0.82	3.00 ± 0.42	4.50 ± 1.02	10.01 ± 2.82

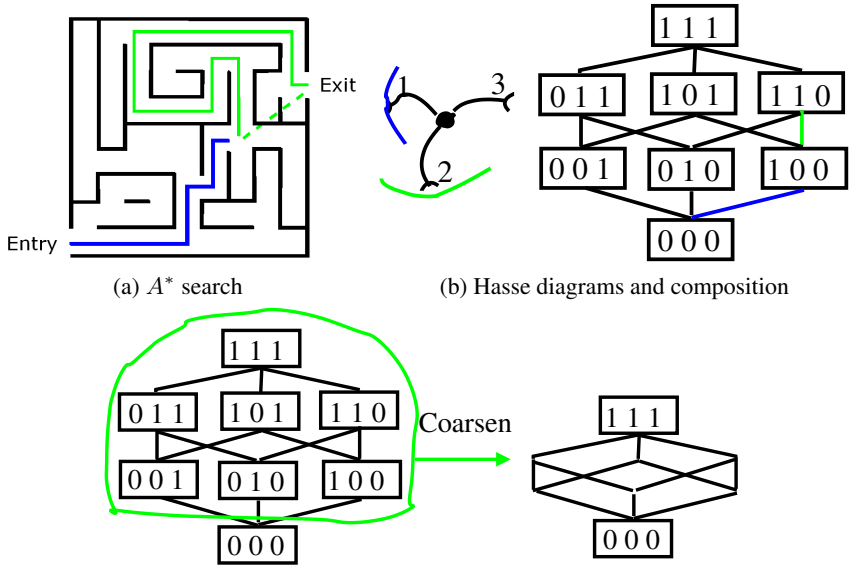
Cascade Detection				
	GDTs	C-DPM	$p_e = 0.05$	$p_e = 0.01$
$\theta = -0.5$	8.95 ± 0.82	0.56 ± 0.07	0.19 ± 0.03	0.23 ± 0.04
$\theta = -0.7$	8.95 ± 0.82	0.72 ± 0.09	0.29 ± 0.04	0.36 ± 0.06
$\theta = -1.0$	8.95 ± 0.82	1.04 ± 0.16	0.51 ± 0.10	1.07 ± 0.29

Table 5.1: Means and standard deviation timings, in seconds, of the considered approaches. GDT stands for distance transforms, BB for Dual Tree Branch-and-Bound, CSC for cascade, and LU-$\{1,5\}$ for lookup-based bounds with $p_e = .01$ and $p_e = .05$ respectively.

obtained with the implementation of [19]. On the left plot we compare the performance of our lookup-based variant of Branch-and-Bound for different values of p_e; we observe that for small values of p_e the performance is identical with that of [19], but with larger values of p_e performance decreases. This validates the need for incorporating uncertainty in lookup-based approximations.

On the right side we compare the performance of the PCA-based cascade of [6] with our lookup-based variant. We observe that performance drops significantly if we use the 'raw' lookup-based estimate of the part scores without the related upper and lower bounds. However, when using bounding intervals to accommodate the 'slack' due to the approximation error, our performance becomes again identical to that of [19].

Coming to computational efficiency, in Table I we provide timings gathered from 1000 images of the PASCAL VOC dataset and averaged over all 20 categories. The first row indicates the time spent to compute part scores by the different methods, and the following rows indicate detection times. We observe that our lookup-based approximations are faster both for DTBB and Cascade Detection for moderate values of the threshold θ; in particular for $\theta = -.7$, or $\theta = -.5$ the lookup-based variant of cascades requires approximately half the time of PCA-based cascade, and $1/30$ of GDT-based detection. We also note that some more time is spent in the second stage when we use the lookup-based bounds, than when we use exact scores. This is the price we pay



(a) A^* search (b) Hasse diagrams and composition

(c) Structure coarsening

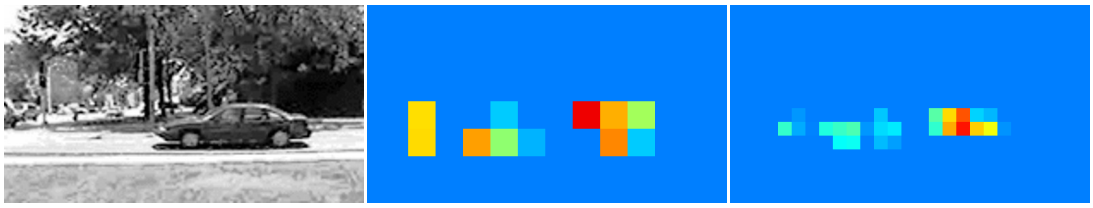
Figure 5.5: From A^* search to A^* parsing: (a) A^* search combines the cost-so-far (dark line) with a heuristic estimate (dashed line) of the cost-to-go (green line). In (b) we view a structure as having bonds that gather their constituents one at a time, and formulate the composition of a structure as climbing to the top of a Hasse diagram. This allows us to construct heuristics for parsing by means of structure coarsening, shown in (c): the task of assembling a structure is 'relaxed' by considering any structure that contains a single part as completed.

for using the optimistic score estimates delivered by the upper bounds, which 'trick' our algorithm into exploring more positions than necessary. This however turns out to still be faster, when considering the overall computation time.

5.2 A^* for hierarchical object models

Having presented efficient algorithms for detection with the star-shaped graphical models of [5] we now turn to detecting objects using the hierarchical, tree-structured models presented in Section 4.2. Detection with such models amounts to building a parse tree, as shown in Fig. 4.7(b) that starts from edge/ridge tokens and finally leads to the whole object.

There is a huge number of candidate parse trees, the vast majority of which will have low scores. In order to control the problem's complexity in [7, 8] we exploit the object's hierarchical representation to develop an hierarchical A^* algorithm. We first use our model to quickly compute score bounds, based on which we construct the heuristics required by A^* - in turn these help us identify a few promising areas, on which more computation is then devoted. This amounts to a coarse-to-fine scheme similar to that of the previous section, but now the coarsening/bounding is also over structures, rather than only over spatial locations.



(a) Input (b) Cost, coarsened problem (c) Cost, fine-level problem

Figure 5.6: Cost function for the coarsened problem (left) and the original problem (right); low costs are red and high costs are blue. *Please see in color.*

Heuristics for parsing

An example that helps describe A^* search is shown in Fig. 5.5(a): an agent wants to exit a maze by following the path of shortest length, say L. Using Dijkstra's algorithm amounts to exploring states with a priority equal to their geodesic distance from the start $C(\nu)$, and will visit all states $\{\nu : C(\nu) \leq L\}$. Instead of prioritizing states based only on their 'cost so far', A^* [20, 21] uses instead a priority equal to $C(\nu) + h(\nu)$, where h is called a *heuristic* and provides an estimate of the 'cost to go'. A heuristic is *admissible* if it provides a lower bound of the cost to go; this is guaranteed to lead to the optimal solution after visiting the -smaller- set of states $\{\nu : C(\nu) + h(\nu) \leq L\}$. Admissible heuristics can be obtained from problem *relaxations*: for our example this could amount to replacing the geodesic distance with an L_1 or L_2 distance.

The counterparts to Dijkstra and A^* in parsing are Knuth's Lightest Derivation (KLD) and Hierarchical A^* Lightest Derivation (HALD) [22], respectively, while an earlier work on A^* for natural language parsing was [23]. KLD prioritizes intermediate structures based on their instantiation cost, while HALD prioritizes intermediate structures based on the sum of their instantiation and heuristic costs. The HALD algorithm was originally applied to contour grouping, where the heuristic costs were obtained by spatial coarsening of the image domain. In our object parsing work we introduce a different way of constructing heuristics which can be understood as a form of 'structure coarsening'.

For this, we work with composition rules where each structure acquires its constituents 'one-at-a-time', a scheme corresponding to the Greibach normal form [24]. This introduces a partial ordering among structures which can be described with a Hasse diagram [25], as shown in Fig. 5.5(b) for a 3-part structure: boxes correspond to structures, for two connected structures the one lying above has more elements, and composing a structure amounts to following a path from the minimum to the maximum element of the graph.

Each transition on this diagram comes at a price, namely the contribution of the acquired structure to the overall score function being optimized. Finding the optimal structure amounts to searching for the shortest path to the diagram's top; the notion of 'structure coarsening' that we introduce amounts to saying that a structure has been fully parsed if one of its constituent is present - the costs for the remaining constituents are replaced by their minimal values. We can interpret this as collapsing the topmost nodes of a Hasse diagram into a single one, as shown in Fig. 5.5(c).

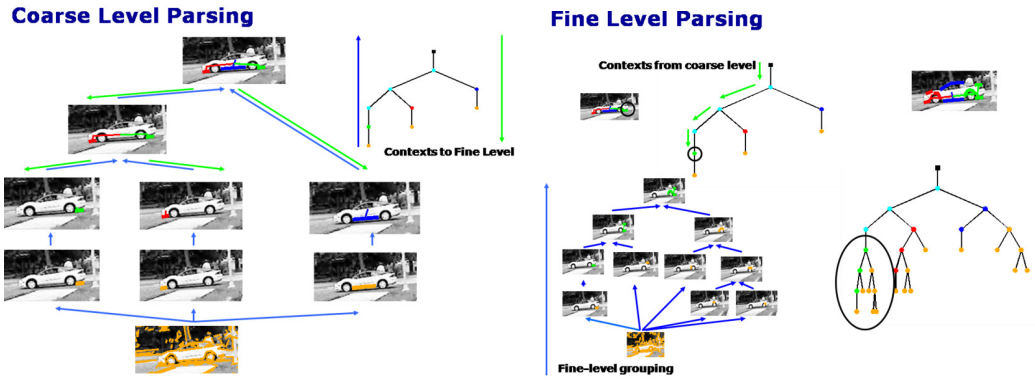


Figure 5.7: A^* Parsing of a car: Initially (left) the car is coarsely parsed using Knuth's Lightest Derivation. The coarsening consists in considering that each car part needs only a single contour. Once a car is thus parsed its parsing is 'rolled back', and generates contexts for parsing at the finer level. This blows-up a single coarse-level node into a full-blown subtree at the fine level. The parse trees indicate the number of low-, mid-, and high- level structures (orange,blue/green/red,black) that are involved in the coarse- and fine-level composition procedure.

Hierarchical object parsing

The coarsening scheme described above provides us with the means of solving the object parsing problem with A^* . Instead of parsing an object entirely from the bottom-up, we can first only roughly detect it by coarsening certain layers of the hierarchy, and then use these results to guide detection at a finer level. The first, 'bottom-up' step can quickly rule out a big portion of the image, and then provide a heuristic function that gives 'top-down' guidance about where more computational resources should be spent.

To illustrate the relationship between the coarse- and fine- level costs, in Fig. 5.6 we show as a heat map the cost function at both levels. We efficiently compute the cost in the middle, which provides us with a lower bound for the cost on the right. We then refine the computation at those locations where the coarse cost falls below a conservative threshold. This ensures that we will not be wasting resources to form an object from the 'bottom-up' at a fine-level if we do not have some 'top-down' evidence from the coarse level score that it is worth doing so.

To describe how the algorithm works we use the illustration in Fig. 5.7; a more technical description is available in [7, 8]. We consider detecting a car structure composed of an engine, cabin and trunk structures, which are in turn composed of multiple contours. Initially we simplify the parsing problem by coarsening the object part level of the hierarchy. For example for the engine structure, we consider that it is complete when we have found one of its contours, e.g. the hood; the same applies to the cabin and trunk parts. In this way we first compute a coarse parse of the object, where each part is composed from one contour. We then backtrack to the intermediate structures formed during coarse detection, and provide them with a heuristic function, equaling a lower bound of their cost-to-go. The availability of a heuristic activates a more detailed parse at the fine level; for example, as shown on the right of Fig. 5.7, the context for the 'back' structure

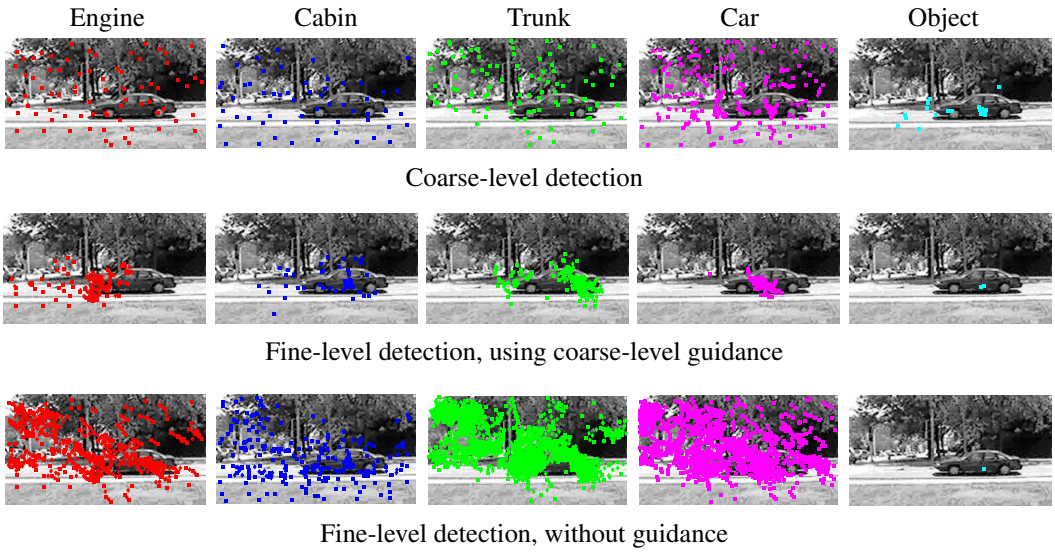


Figure 5.8: The top-two rows demonstrate the Coarse-to-Fine detection scheme, which is contrasted to the plain, Fine-Level detection scheme of the bottom row. At the coarse level a small set of candidate object locations is quickly identified; these locations guide search at the fine level, acting like top-down guidance. Instead, when doing Fine-level Parsing without guidance (bottom row) a detailed parse of the object's parts is performed in several background locations, e.g. around trees. This wastes computation resources on locations which end up being useless.

initiates its fine-level composition, expanding a single node of the coarse parse to a whole subtree.

In Fig. 5.8 we illustrate the computational gain due to this scheme on an image containing a substantial amount of clutter. At the coarse level we coarsen the part level, so we find a few structures per part - we show their centers as dots. We do not coarsen the object level, namely we require all parts to be present, which reduces the candidate objects as shown in the top right corner of Fig. 5.8. Starting from these 'shortlisted' object hypotheses, we backtrack to the lower levels and focus on the image areas likely to contain an object. We now build full-blown object parts with multiple contours, which leads to a single object instantiation being above threshold.

By contrast, as shown in the bottom row of Fig. 5.8 purely bottom-up fine-level detection is 'short sighted'. By trying to form all object parts at full detail from the beginning, it wastes computational resources. This is evident from the large number of individual object parts formed on the background - the final solution is the same, but takes longer to compute.

5.3 Reinforcement learning for facade parsing

The models that we have been working with so far have a fixed number of parts; this is a common assumption in object detection, but does not fully explore the potential of recursive, grammatical representations for vision. In [9, 26] we explore a full-blown parsing task for the problem of facade interpretation: as shown in Fig. 5.9 our goal is to partition a rectified facade into semantic classes

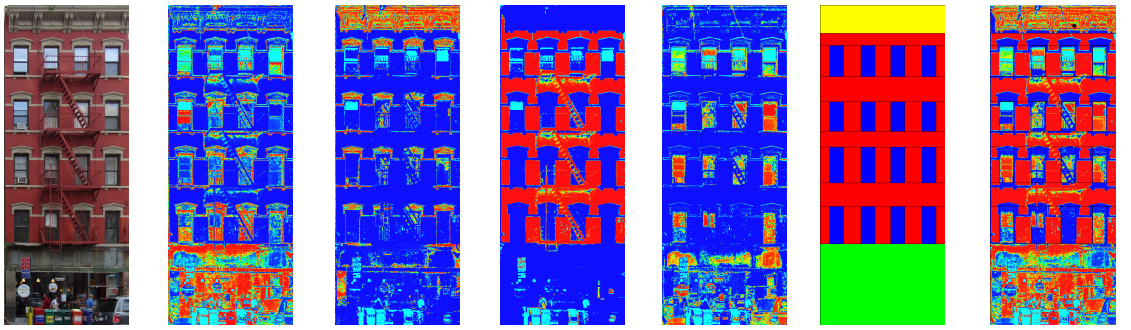


Figure 5.9: Illustration of the objective function: the first four rows are heat maps of the pixel-wise merit functions for each terminal class (shop/roof/wall/window). The penultimate row is a candidate labeling (color-coded), and the last row is a combination of the merits, 'multiplexed' according to c. Our goal is to maximize $\sum_{x,y} m(x, y, c(x, y))$ with respect to $c(x, y)$.

such as 'wall', 'window', 'balcony' or 'roof'. This is a problem of practical importance for the semantic interpretation of large image datasets in commercial urban map applications, but is also theoretically interesting, as it involves fitting models that accommodate structure variation.

We develop our ideas around shape grammars (SG's) [27–29], which were used earlier for building interpretation using either heuristic bottom-up computation [30] or Monte Carlo optimization [31]. Instead, we use Reinforcement Learning (RL) [32] to efficiently solve the parsing problem: we apply established RL techniques such as Hierarchical RL and state aggregation to our problem and develop a novel method to exploit image-based information during optimization, which allows us to use bottom-up guidance while being resilient to potential front-end failures.

This method has delivered state-of-the-art results in a fraction of the time required by [31], and has been validated under diverse imaging conditions; our implementation is available at [33].

Facade parsing and Shape Grammars

We first phrase the facade interpretation task as one of optimizing a merit function, and then turn to constraining the space of solutions in an architecturally meaningful way.

We consider that we have a function $m(x, y, c) \in [0, 1]$ that indicates the score obtained for labelling pixel x, y as having class c - for instance in columns 2-5 of Fig. 5.9 we show the pixel-wise class posteriors of random forest classifiers for different classes. Our goal is to find a labelling $c(x, y)$ that maximizes the cumulative merit $\sum_{x,y} m(x, y, c(x, y))$. This is similar to the semantic segmentation task [34], but in our case the labeling needs to not only be smooth, but also architecturally meaningful.

To phrase this constraint we use the shape grammar (SG) [27–29] shape modeling framework. Adapting SGs to our setting, we call 'shape' a rectangular domain $c(x, y, w, h)$ where c is the type (e.g. 'd' for door, 'w' for wall), (x, y) the position, and (w, h) the width and height. Terminal shapes relate to the image observations, while non-terminals only relate to compositions of termi-

Parent	Children	Split
Axiom(W,H)	Facade(0,0,W,H)	None
Facade(0,Y,W,H)	Floor(0,Y,W,h) Fa-Wall(0,Y+h,W,H-h)	Y:h
Fa-Wall(0,Y,W,H)	Wall(0,Y,W,h) Facade(0,Y+h,W,H-h)	Y:h
Floor(X,Y,W,H)	Wall(X,Y,w,H) Fl-Win(X+w,Y,W-w,H)	X:w
Fl-Win(X,Y,W,H)	Window(X,Y,w,H) Floor(X,Y,W-w,H)	X:w

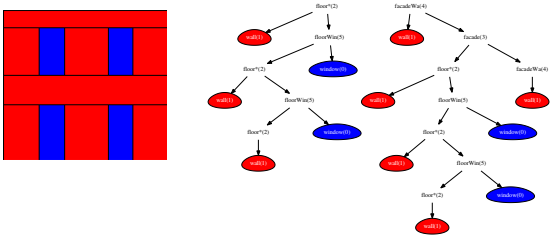


Figure 5.10: A toy 2D split grammar (left) and a shape generated by it (right). Walls and windows are terminals; filling their domain with their color gives the facade on the right.

nals, or other non-terminals. A shape grammar also involves a set of replacement rules to break up a non-terminal into non-terminals and/or terminals. In our 'Binary Split Grammar' (BSG) variant each rule splits a shape along a single dimension and into at most two shapes. For instance a rule can break up a facade as follows:

$$fa(X,Y,W,H) \underset{H:h,fl}{\longrightarrow} fl(X,Y,W,h)fa(X,Y+h,W,H-h),$$

meaning that we take a shape of type fa(cade), and split it along the H(eight) dimension into a fl(oor) of height h and a remainder of type fa(cade); the support of the original facade equals the union of the supports of the symbols on the right. Finally an 'axiom' shape appears only on the left side of a rule; a tree rooted at an axiom with terminals at all leaves is called a 'derivation tree'.

We demonstrate in Fig. 5.10 a small BSG for facades. We denote in color the grammar terminals for walls and windows. The split direction alternates per layer, while the grammar enforces a certain alteration of shapes within each direction; namely the 'Fa-Wall' and 'Fl-Win' shapes indicate that the decomposition is half-done, and require the next shape to be of a complementary type. On the right side we show a shape derived by applying some of the grammar rules, as well as the respective derivation tree; we can thus see our grammar as a generative model for a variety of facades, obtained by using different rule parameters.

Coming to the inverse problem of fitting a SG model to a given image, if we define the 'merit' of terminal $c_A(x,y,w,h)$ as:

$$M(c_A(x,y,w,h)) = \sum_{x'=x}^{x+w} \sum_{y'=y}^{y+h} m(x',y',c_A) \tag{5.15}$$

and express the merit of a non-terminal recursively as the sum of its descendants' merits, the objective laid out in the beginning amounts to the merit of the 'axiom' shape. We thus face a problem of picking the set of grammar rules that maximizes the merit of the axiom; this is a challenging optimization problem, as we have an unknown a priori number of rules, each rule comes with continuous parameters, and the merit is defined recursively for non-terminals. This led us to the Reinforcement Learning-based formulation that is outlined below.

Reinforcement Learning and bottom-up guidance of shape parsing

We view parsing as the work of an agent (or, algorithm) that starts with a tree containing only the axiom shape at its root and iteratively applies rules to nodes of the tree until obtaining a derivation tree. The agent collects 'rewards' at each step and aims at maximizing its cumulative reward.

At any point in time the currently entertained derivation is called the 'state' s of the agent, and the rule being applied is called the 'action', a. We consider that the agent follows a policy $\pi(s,a) = p(a|s)$, indicating the probability with which action a is chosen at state s. The *action-value function* $Q^{\pi}(s,a)$ indicates the anticipated cumulative reward for picking action a at state s, and then following the policy π afterwards; this can be expressed by Bellman's recursion:

$$Q(s,a) = R(s,a) + \sum_{a'} \pi(s',a) Q(s',a'), \tag{5.16}$$

where $R(s,a)$ is the reward obtained for taking action a at s, s' is the state subsequent to taking action a at s, and the summation on the right is marginalizing over the subsequent actions, a'; s' and $R(s,a)$ may also be random, and require marginalization, but we omit this for simplicity.

Reinforcement Learning uses several iterations ('episodes') to improve $\pi(s,a)$ by estimating $\pi(s,a)$ simultaneously with $Q^{\pi}(s,a)$ and gradually finding a policy π that leads to large values of $Q^{\pi}(s,a)$. In particular, an ϵ-greedy policy can be determined from the merit function as follows:

$$\pi(s,a) = (1-\epsilon)\delta(a,a_s^*) + \epsilon U(a), \quad a_s^* = \text{argmax}_a Q(s,a), \tag{5.17}$$

where a_s^* is the apparently best action, based on the currently entertained $Q(s,a)$ and $U(a)$ is a uniform distribution on actions; namely we 'exploit' the gathered knowledge $Q(s,a)$ with probability $1-\epsilon$ and 'explore' new options with probability ϵ. The parameter ϵ is decreased over different runs of the algorithm, gradually compounding the gathered knowledge. As the agent is executing its task we can update $Q(s,a)$ 'on the fly', according to the Q-learning update:

$$\Delta Q(s,a) = \alpha[R(s,a) + \max_{a'} Q(s',a') - Q(s,a)] \tag{5.18}$$

where α is a learning rate. This brings closer a previous estimate of $Q(s,a)$ with the more up-to-date estimate, $R(s,a) + \max_{a'} Q(s',a')$. Repeatedly applying Eq. 5.18 for several runs while slowly decreasing ϵ and α results in an optimal policy [32].

At an intuitive level the advantage of RL compared to Dynamic Programming is that RL does not explore all state-action combinations, but rather samples some promising ones per iteration; and unlike a naive Monte Carlo estimation of $Q(s,a)$ by sampling, the update in Eq. 5.18 exploits the problem structure and modifies $Q(s,a)$ at each step, leading to fast convergence. These comparisons are more thoroughly articulated in [32].

In our work we use RL to tackle several problems of shape grammar parsing, involving the hierarchical definition of the rewards for non-terminals, for which we use Hierarchical RL [35], and the enforcement of symmetry across floors, for which we used state aggregation [36] to tie together the policy functions of different floors. We refer to [9, 26] for a thorough presentation and focus instead on incorporating bottom-up guidance in shape parsing.

According to Eq. 5.17, at any stage our agent will 'explore' an action at random with probability ϵ; this allows it to globally optimize its policy by getting 'unstuck' from early errors. But

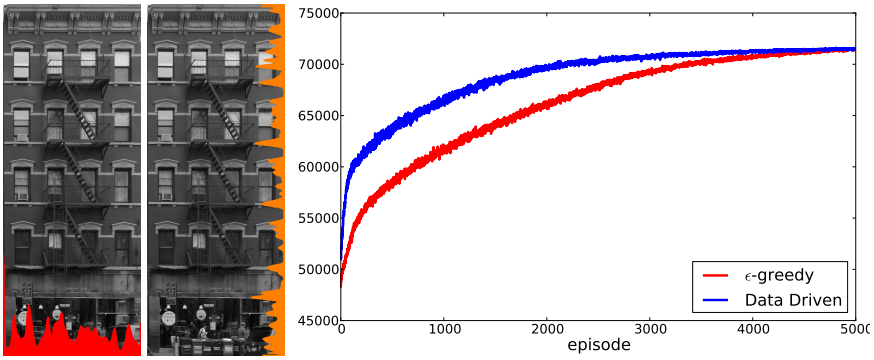


Figure 5.11: Image-based guidance and Q-learning: the image gradient guides action exploration, by proposing to 'break' shapes at strong intensity discontinuities. This accelerates Q-learning over the 'agnostic' ϵ-greedy baseline, as can be seen in the average return-per-episode plots on the right.

instead of having an 'uninformed' uniform distribution on actions, we can instead use image-based cues to suggest good actions. For instance when splitting a floor into windows and walls, taking action a at position x will place the next boundary at $x + a$. Since the appearance of walls and windows is supposed to be different, we can anticipate that the horizontal image gradient will be strong at $x + a$, and use the image gradient to favor those actions a that will lead to such positions.

As illustrated in Fig. 5.11 we develop such a scheme by accumulating horizontal gradients vertically (left) and vertical gradients horizontally (right). These histograms are transformed through a softmax function to construct a proposal distribution that steers our agent's action exploration: instead of the 'agnostic' ϵ-greedy policy in Eq. 5.17, we use a data-driven policy of the form:

$$\pi(s,a) = (1-\epsilon)d_{a^*}(a) + \epsilon P(a;x), \quad P(a;x) = \frac{e^{h(x+a)}}{\sum_{a'} e^{h(x+a')}}, \tag{5.19}$$

where h is the cumulative gradient signal. When operating in 'exploration' mode our agent will now use image guidance to check locations around boundaries more frequently; however we do not compromise the convergence of the algorithm, as $\pi(s,a)$ stays above zero for all actions. Empirically, as we can see on Fig. 5.11 we observe that the image-driven strategy results in a significant speed-up over the plain and 'uninformed' ϵ-greedy search. We anticipate that this scheme can apply to other vision problems in a manner similar to Data-Driven MCMC [37].

Results

For evaluation we use the facade benchmarks in [31] and [9], where ground-truth segmentations into 'window', 'wall', 'balcony', 'door', 'roof', 'sky', and 'shop' regions are provided for 10 and 84 Parisian building facades respectively. All facades follow the Haussmann architecture rules, making it easy to formulate a binary shape grammar (BSG) for the task at hand.

As in [31] we use Random Forests to recover the pixel-wise merit functions and consider the same optimization problem with them - any differences in performance can thus be attributed to the optimization algorithm. As can be seen from the confusion matrices in Fig. 5.12, our method gives

0	**81**	11	3	0	5	0	0	**68**	23	4	0	4	2	0	*window*
+1	5	**84**	7	1	1	0	2	3	**87**	7	0	1	0	1	*wall*
−9	10	26	**63**	0	0	0	1	9	24	**64**	0	1	0	1	*balcony*
+13	0	2	0	**84**	0	0	14	0	1	0	**53**	0	0	46	*door*
+6	10	4	0	0	**86**	0	0	6	3	0	0	**83**	8	0	*roof*
0	2	0	0	0	4	**94**	0	1	0	0	0	3	**96**	0	*sky*
+2	0	1	0	2	0	0	**97**	0	6	1	6	0	0	**88**	*shop*
	Facade benchmark of [31]							Facade benchmark of [9]							

Figure 5.12: Confusion matrices on the datasets of [31] and [9]; the left column compares to [31].

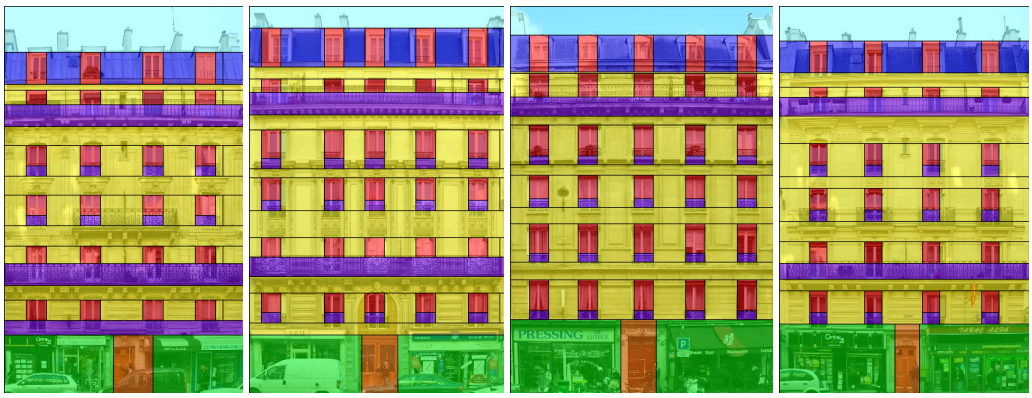


Figure 5.13: Parisian facade parsing with a binary shape grammar for Haussmann architecture.

typically better results on the same benchmarks; however the results of our method are delivered in a fraction of the time: [31] need around 10^6 iterations and 10 minutes to converge, whereas the RL-based approach only needs 2000 iterations and 30 seconds. As shown in Fig. 5.13, apart from some minor geometrical mistakes, the delivered buildings interpretations are mostly plausible.

In the top row of Fig. 5.14 we apply our parsing method to skyscrapers; the difficulty of these examples lies in the -unknown- number of decisions the agent must take to perform a segmentation; we see that our algorithm manages to correctly parse these buildings. To further illustrate the flexibility of our method, in the rightmost example we parse the input facade with a BSG that enforces an alternation of two kinds of floors. In the bottom row of Fig. 5.14 we provide examples of occlusions and lightings that are correctly handled by the proposed framework. We observe that the algorithm 'hallucinates' windows behind the occlusions, due to the model's bias for symmetry.

In Fig. 5.15 and Fig. 5.16 we demonstrate parsing results based on another merit function inspired from Grab-cut [38]. As shown in Fig. 5.15, the user manually selects on the image some examples of terminal elements and we fit a Gaussian Mixture Model (GMM) per terminal type, each made of 3 Gaussian kernels. The GMM-based class posteriors are then used to define the pixel-wise merit functions. The results show that we can easily combine architectural shape priors

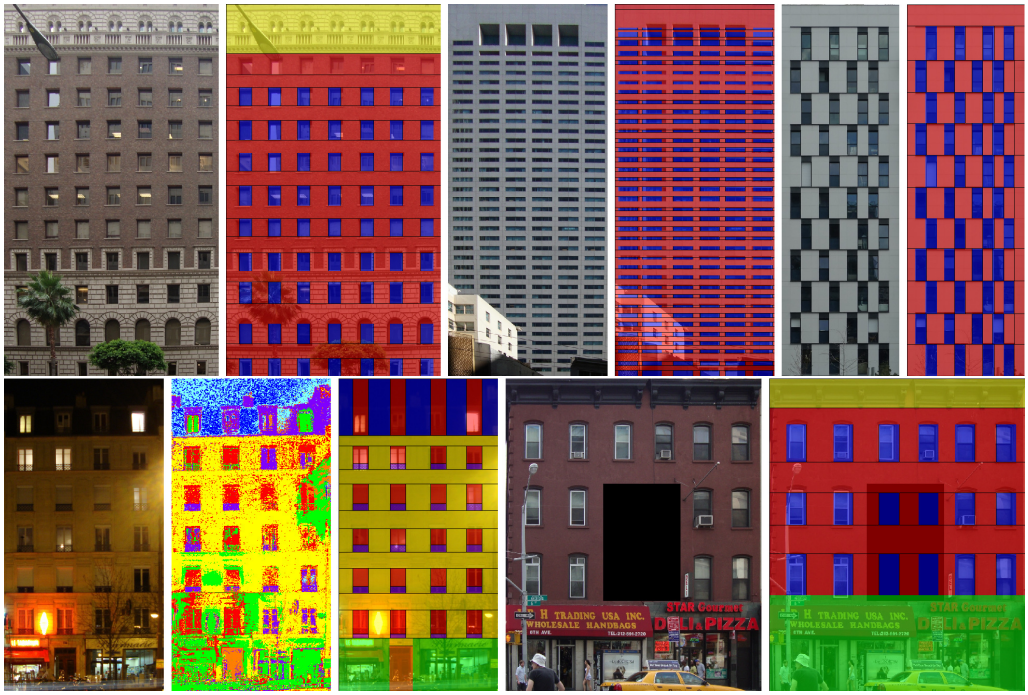


Figure 5.14: The top row demonstrates our algorithm's ability to deal with a large and unknown number of floors, and the bottom row its robustness to poor illumination and to occlusions.

with user annotation, which could prove useful in the development of procedural models for large-scale city datasets.

As an illustration of how this can be done, we note that we can easily construct 3D building models by turning the 2D inferred grammar rules into 3D variants and manually adding a depth value per terminal type. In Fig. 5.17 for instance we apply this approach on some Haussmannian buildings, and render novel 3D views from different viewpoints and illuminations; such results have previously been demonstrated for generic natural images e.g. in [39]; what is a main advantage of our method is the granularity and precision at which this task is solved by employing a strong prior model for the shapes being analyzed.

Figure 5.15: Grab-Cut -based facade parsing: we show, from left to right the original image, the user's brush strokes used to train a GMM classifier a pixel-wise segmentation using the learned GMMs, and the optimal parse delivered by our algorithm.

Figure 5.16: Grab-cut parsing results on buildings with classic architectures from Barcelona and Budapest and a modern building in Paris.

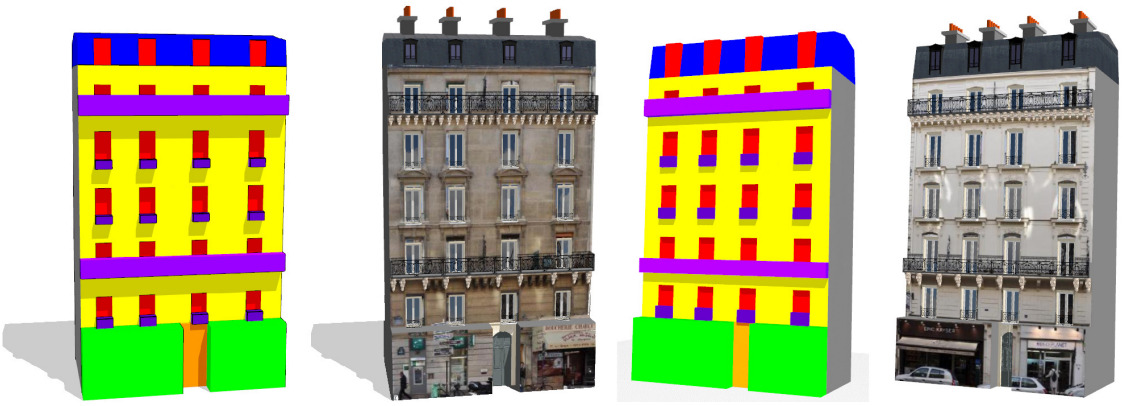


Figure 5.17: Image-based modeling: we turn a 2D rule sequence estimated from an image into a 3D sequence by manually adding depth values, and texture-map the input image to the 3D model.

Bibliography

[1] I. Kokkinos, "Rapid deformable object detection using dual-tree branch-and-bound," in *Proc. NIPS*, 2011.

[2] ——, "Bounding part scores for rapid detection with deformable part models," in *2nd Parts and Attributes Workshop, in conjunction with ECCV 2012*, 2012.

[3] I. Kokkinos, "Rapid Deformable Object Detection using Bounding-based Techniques," INRIA, Tech. Rep. RR-7940, 2012.

[4] P. Felzenszwalb and D. Huttenlocher, "Efficient matching of pictorial structures," in *Proc. CVPR*, 2000.

[5] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. CVPR*, 2008.

[6] P. F. Felzenszwalb, R. B. Girshick, and D. A. McAllester, "Cascade object detection with deformable part models," in *Proc. CVPR*, 2010.

[7] I. Kokkinos and A. Yuille, "HOP: Hierarchical Object Parsing," in *Proc. CVPR*, 2009.

[8] I. Kokkinos and A. Yuille, "Inference and Learning with Hierarchical Shape Models," *Int.l Journal of Computer Vision*, vol. 93, pp. 201–225, 2 2011.

[9] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, L. Van Gool, and N. Paragios, "Shape grammar parsing via reinforcement learning," in *Proc. CVPR*, 2011.

[10] Y. Chen, L. Zhu, C. Lin, A. L. Yuille, and H. Zhang, "Rapid inference on a novel AND/OR graph for object detection, segmentation and parsing," in *Proc. NIPS*, 2007.

[11] V. Ferrari, M. J. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Proc. CVPR*, 2008.

[12] M. Pedersoli, A. Vedaldi, and J. Gonzàlez, "A coarse-to-fine approach for fast deformable object detection," in *Proc. CVPR*, 2011.

[13] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *Proc. CVPR*, 2013.

[14] A. G. Gray and A. W. Moore, "Nonparametric density estimation: toward computational tractability.," in *SIAM International Conference on Data Mining*, 2003.

[15] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," Cornell CS, Tech. Rep., 2004.

[16] P. Felzenszwalb and D. Huttenlocher, "Pictorial Structures for Object Recognition," *Int.l Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[17] A. Vedaldi and A. Zisserman, "Sparse kernel approximations for efficient classification and detection," in *Proc. CVPR*, 2012.

[18] M. Mitzenmacher and E. Upfal, *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[19] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, *Discriminatively trained deformable part models, release 4*, http://people.cs.uchicago.edu/ pff/latent-release4/.

[20] J. Pearl, *Heuristics*. Addison-Wesley, 1984.

[21] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[22] P. F. Felzenszwalb and D. Mcallester, "The Generalized A * Architecture," *Artificial Intelligence*, vol. 29, pp. 153–190, 2007.

[23] D. Klein and C. D. Manning, "A * Parsing : Fast Exact Viterbi Parse Selection," *Science*, 2004.

[24] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 2006.

[25] G. Birkhoff, *Lattice Theory*. AMS, 1967.

[26] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios, "Parsing Facades with Shape Grammars and Reinforcement Learning," *IEEE Trans. PAMI*, vol. 35, no. 7, pp. 1744–1756, 2013.

[27] G. Stiny, "Pictorial and Formal Aspects of Shape and Shape Grammars," PhD thesis, Birkhäuser, 1975.

[28] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky, "Instant architecture," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 669–677, 2003.

[29] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool, "Procedural modeling of buildings," *ACM Transactions on Graphics*, vol. 25, no. 3, p. 614, 2006, ISSN: 07300301. DOI: 10.1145/1141911.1141931.

[30] P. Müller, G. Zeng, P. Wonka, and L. Van Gool, "Image-based procedural modeling of facades," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 85, 2007, ISSN: 0730-0301.

[31] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios, "Segmentation of building facades using procedural shape priors," in *Proc. CVPR*, 2010.

[32] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[33] O. Teboul, *GRAPES - a GRAmmar Parser for shapES*, http://vision.mas.ecp.fr/Personnel/teboul/grapesPage/index.php, 2012.

[34] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Proc. ECCV*, 2006.

[35] T. G. Dietterich, "Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition," *Journal of Artificial Intelligence Research*, vol. 13, pp. 227–303, 2000.

[36] S Singh, T. Jaakkola, and M. Jordan, "Reinforcement Learning with soft state aggregation," in *Proc. NIPS*, 1995.

[37] Z. W. Tu, X Chen, A. Yuille, and S.-C. Zhu, "Image Parsing: Unifying Segmentation, Detection, and Recognition," *Int.l Journal of Computer Vision*, vol. 63, no. 2, pp. 113–140, 2005.

[38] A. Blake, C. Rother, M Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptative GMMRF model," in *European Conference on Computer Vision (ECCV)*, 2004.

[39] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 577–584, 2005.

Chapter 6

Future Research

I looked, and, behold, a new world! There stood before me, visibly incorporate, all that I had before inferred, conjectured, [.] a beautiful harmonious something - for which I had no words; but you, my readers in Spaceland, would call it the surface of the sphere.
– Flatland *Edwin A. Abbott*

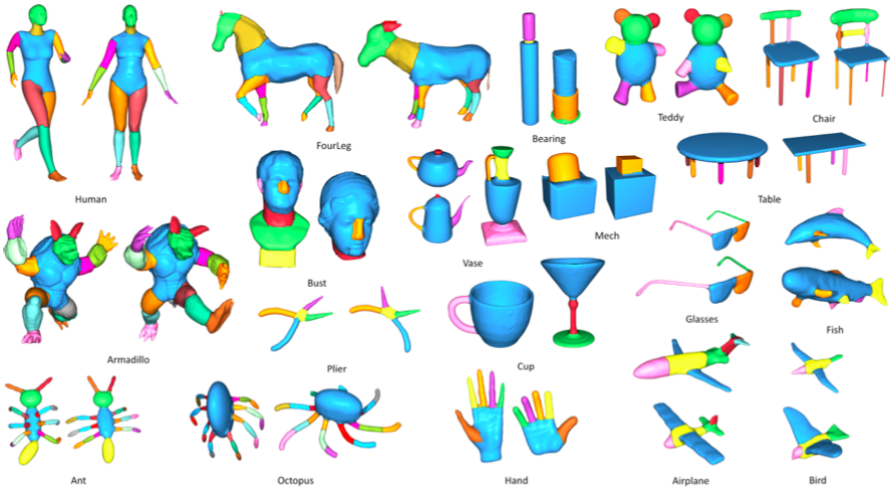
This thesis presents methods to reliably extract, describe, model and detect shape in natural images, by combining techniques from machine learning and optimization. The motivation for these works is the understanding that shape- and more generally, grouping- based representations can go all the way from image features to objects, thereby facilitating a proper coupling of bottom-up and top-down processing for fast detection. What emerges from the developments described in this work is that we can indeed do this, whether in deterministic optimization, such as Branch-and-Bound, or in stochastic optimization, such as Reinforcement Learning. Looking deeper into such connections and means of exploiting them for scalable detection is a primary research priority for the near future, since it is a problem of both theoretical and practical interest.

One aspect which seems central to scaling up this approach to 3D recognition and large-scale object detection is the development of appropriate mid-level representations. As outlined in Section 1.2 this is a problem that has received increased interest lately, and in on-going work we have also integrated such ideas in Branch-and-Bound detection in the 2D case. So far the use of shared parts in 3D has been pursued through voting in 3D category detection [1, 2], while the current 3D extensions of DPMs are mostly view-based [3–5] and do no yet implement part sharing.

We anticipate that questions pertaining to part sharing in 3D will be addressed most successfully by relying on explicit 3D representations. On the one hand depth sensors, such as Microsoft's Kinect, are now cheap enough to bring surface modeling and matching into the mainstream of computer vision - so these advances may be directly exploitable at test time for detection. On the other hand, even if we do not use depth information at test time, having 3D information can simplify the modeling task during training.

One of the most promising advances in this direction is illustrated in the results of [6] shown in Fig. 6.1. The figure on the left illustrates how the correspondence problem for 3D data can exploit information that is lost due to the effects of 2D projection: when seen as 2D images the two shapes differ dramatically, but in 3D the correspondence problem becomes substantially more clear-cut, and therefore more refined notions of consistency can be conceived and enforced. Furthermore, as illustrated in Fig. 6.1(b), addressing the problem of co-segmenting/registering data on a 3D surface database brings us much closer to the construction of shared mid-level parts, in the vein

(a) Consistent surface registration



(b) Surface co-segmentation

Figure 6.1: 3D surface registration and model building, from [6]: (a) shows the 3D correspondences used to establish geometric consistenty among two different shapes, in a manner that allows one-to-many correspondences; (b) shows in color the results of co-segmenting a large dataset of surfaces.

of generalized cylinders/geon/ribbon representations [7–9] pursued in the earliest days of vision. What was missing from research in that era was the 3D information about the scenes - so they were facing a much harder task, with very limited tools.

Currently we have the necessary 3D information at hand from range sensors, we have rigorous tools exist to describe surfaces, laid out in our chapter on invariant descriptors, we know how to train statistical models for deformable objects and we also know how to detect such objects efficiently. In on-going work with collaborators we have started exploring combinations of such aspects, namely (i) the use of surface analysis tools to match surfaces from depth sensors (ii) using branch-and-bound for efficient inference in 3D space and (iii) groupwise-registration to build statistical 3D surface models. In the coming years we intend to pursue a tighter integration of these different directions for scalable 3D object recognition.

Bibliography

[1] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool, "Towards multi-view object class detection," in *Proc. CVPR*, 2006.

[2] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and pose estimation," in *Proc. ICCV*, 2011.

[3] B. Pepik, P. Gehler, M. Stark, and B. Schiele, "3DDPM - 3D deformable part models," in *Proc. ECCV*, 2012.

[4] ——, "Teaching 3D Geometry to deformable part models," in *Proc. CVPR*, 2012.

[5] S. Fidler, S. Dickinson, and R. Urtasun, "3D object detection and viewpoint estimation with a deformable 3D cuboid model," in *Proc. NIPS*, 2012.

[6] Q.-X. Huang, V. Koltun, and L. J. Guibas, "Joint Shape Segmentation with Linear Programming," *ACM Trans. Graph.*, vol. 30, no. 6, p. 125, 2011.

[7] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 200, no. 1140, pp. 269–294, 1978.

[8] I. Biederman, "Recognition-by-components: a theory of human image understanding," *Psychological Review*, no. 2, pp. 115–147, 1987.

[9] G. Russell, R. Brooks, and T. Binford, "The ACRONYM model-based vision system," in *Proc. IJCAI*, 1979.

Chapter 7

Curriculum Vitae

Education

Ph.D., Electrical and Computer Engineering, 2001-2006
National Technical University of Athens.
Ph.D. Dissertation: ‘‘Synergy between Image Segmentation and Object Recognition using Geometrical and Statistical Computer Vision Techniques.”

M. Eng. Diploma, Electrical and Computer Engineering, 1997-2001
National Technical University of Athens.
M. Eng. Thesis: ‘‘Modeling and Prediction of Speech Signals using Chaotic Time-Series Analysis Techniques.”

Appointments

9/2008 - present	Assistant Professor	ECP, Department of Applied Mathematics.
9/2008 - present	Affiliate Researcher	INRIA-Saclay, Galen Group.
6/2006 - 8/2008	Postdoctoral Researcher	UCLA, Department of Statistics.
11/2001- 6/2006	Graduate Research Assistant	NTUA, School of ECE.
3/2002 - 7/2002	Visiting Student	INRIA-Sophia Antipolis, Odyssée Group.

Curriculum Development

Machine Learning for Computer Vision (2008-)
Mathématiques, Vision, Apprentissage (MVA) M2-Master.
8 lectures - 24 teaching hours, attended by 20-30 students annually.
The course covers discriminative techniques (Logistic Regression, Adaboost, Support Vector Machines, Multiple Instance Learning) and generative models (Mixture Models and Expectation-Maximization, Linear Models, Hidden Markov Models, Markov Random Fields) with an emphasis on applications to computer vision.

Introduction to Signal Processing (2009-)
7th semester course, École Centrale Paris.
11 lectures - 22 teaching hours, attended by 35-45 students annually.

The course introduces basic concepts from signals and systems (Frequency-domain analysis of Signals and Systems, Modulation and Gabor filters, Sampling, the Discrete-Time Fourier Trasform, the Z-transform, the Fast Fourier Transform) and elements of random signals (Autoregressive-Moving Average Processes, the Linear Predictive Coding model, Wiener and Kalman filtering).

Introduction to Computer Vision (2009-)
8th semester course, École Centrale Paris.
11 lectures - 22 teaching hours, attended by 20-30 students annually.
The course covers techniques for image analysis (Filterbanks, Scale-Space and Partial Differential Equations), energy minimization (Calclulus of Variations and Curve Evolution, Markov Random Fields), and category modeling (Active Appearance Models, Deformable Part Models, Bag-of-Words models). Applications include image denoising, inpainting, feature detection, texture analysis, image segmentation, motion estimation, object detection and tracking.

PhD Student Supervision

Haithem Boussaid, École Centrale Paris (2010-)
Co-advised with Nikos Paragios.
Topic: Learning deformable models for medical image analysis.

Stavros Tsogkas, École Centrale Paris (2011-)
Topic: Shape-based optimization for object category detection.

Olivier Teboul, École Centrale Paris (2008-2011)
Co-advised with Nikos Paragios.
Topic: Reinforcement learning-based parsing of building facades with shape grammars.

Eduard Trulls, Universitad Polytecnica de Catalunia (2012-)
Advisors: Francesc Moreno and Alberto Sanfeliu.
Topic: Dense segmentation-aware descriptors for matching and recognition.

Michalis Raptis, University of California at Los Angeles (2009-2011)
Advisor: Stefano Soatto.
Topic: Mid-level video models for action recognition and localization.

Master and Intern Student Supervision

Siddhartha Chandra, École Centrale Paris (2011)
Topic: Descriptor matching for RGB-D data.
Co-advised with Pawan Kumar.

Stavros Tsogkas, École Centrale Paris (2011)
Topic: Learning symmetry detection.

Ishan Misra, École Centrale Paris (2012)
Topic: Groupwise shape-from-shading.

Aman Bindal, École Centrale Paris (2009)
Topic: Real-time jingle detection in video streams.

Thesis Committees

Anastasios Roussos, National Technical University of Athens (2010)
Topic: Nonlinear Diffusion in Computer Vision and Statistical Shape Models, with Applications in Image Analysis of Articulators of Voiced and Signed Speech.

Olivier Teboul, École Centrale Paris (2011)
Topic: Shape Grammar Parsing: Application to Image-based Modeling.

Christos Pappas, University of Ioannina (on-going)
Topic: Scene Recognition and Semantic Segmentation.

Research Funding

FP7 ICT-9 Project RECONFIG (2013-2016)
Cognitive, Decentralized Coordination of Heterogeneous Multi-Robot Systems via Reconfigurable Task Planning.
Joint research project with KTH (Sweden), U. Aalto (Finland), NTUA (Greece)
Our goal is to use 3D object understanding and localization as a medium for multi-agent coordination and collaboration.
Funding: 400K Euros for ECP, 2.300K Euros total.

FP7 ICT-9 Project MOBOT (2013-2016)
Intelligent Active MObility Assistance RoBOT integrating Multimodal Sensory Processing, Proactive Autonomy and Adaptive Interaction.
Joint research project with TU Munich, U. Heidelberg (Germany), Accrea (Poland), NTUA-ICCS, ILSP (Greece)
Our goal is to equip robotic walking assistants with 3D pose estimation and action recognition capabilities to enable the proactive assistance of elderly users with walking disabilities.
Funding: 300K Euros for INRIA, 3.100K Euros total.

ANR-JCJC HiCoRe (2010-2014)
HIerarchical COmpositional REpresentations for computer vision.
Joung Researcher Award of the French National Research Foundation
Our goal is to develop computational mechanisms for inference and learning in hierarchical, shape-based object representations.
Funding: 168K Euros for ECP.

Distinctions and Awards

Reviewer award, IEEE Conference on Computer Vision and Pattern Recognition, 2013.

Reviewer award, International Conference on Computer Vision, 2009.

Bodossaki foundation scholarship as a graduate student.

Obtained in 4 years the 5-year NTUA M. Eng. Degree, ranking in the top 2%.

Paris Kanellakis award for highest ranking student in the Computer Science major.

National scholarship foundation awards as an undergraduate.

Academic Service

Associate Editor
Image and Video Computing Journal (2011-).

Journal Reviewer
International Journal of Computer Vision (2009-).
IEEE Transactions on Pattern Analysis and Machine Intelligence (2006-).
IEEE Transactions on Image Processing (2006-).
IEEE Transactions on Systems, Man and Cybernetics, B (2011).
IEEE Transactions on Neural Networks (2010).
Computer Vision and Image Understanding (2008-).
Image and Video Computing Journal (2010).
Computer Speech and Language (2009).
EURASIP Journal of Image and Video Processing (2012).
Machine Vision and Applications (2013).

Program Chair
IEEE Workshop on Perceptual Organization in Computer Vision (POCV), 2012.

Area Chair
IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012.

Program Committee
Int'l. Conf. on Computer Vision (ICCV) 2007, 2009, 2011, 2013.
IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2009, 2010, 2011, 2013.
European Conf. on Computer Vision (ECCV) 2010.
Asian Conf. on Computer Vision (ACCV) 2009, 2010, 2012.
Int'l. Conf. on Artificial Intelligence and Statistics (AISTATS) 2011.
Int'l. Conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR) 2007, 2009, 2011, 2013.
Int'l. Workshop on Vision, Modeling and Visualization, 2013.

ACCV Workshop on Detection and Tracking in Challenging Environments, 2012.
Int'l. Workshop on Stochastic Image Grammars 2009, 2011.
IEEE Workshop on Perceptual Organization in Computer Vision (POCV), 2010.
Int'l Symposium on Visual Computing, 2009, 2010, 2011.
Indian Conference on Vision Graphics and Image Processing (ICVGIP), 2008, 2010.

Grant Reviewer
European Union, ERC awards, 2010.
Swiss National Science Foundation, 2013.

Invited Presentations and Academic Visits

July 2013	UCLA, IPAM summer school on computer vision
June 2013	Stony Brook University
April 2013	USI Lugano, visiting faculty (with Prof. M. Bronstein)
April 2013	Zuse Institute Berlin, Graphics Seminar
July 2012	JHU Summer School on Human Language Technology "*Towards a Detailed Understanding of Visual Scenes*" collaboration project with the University of Oxford, Chicago, and Oulu
June 2012	Carnegie Mellon University
June 2012	École Normale Superieure/Willow Group
June 2012	National Technical University of Athens
July 2011	ETH, Visual Computing Lunch
July 2011	USI Lugano, visiting faculty (with Prof. M. Bronstein)
June 2011	CVPR workshop on Symmetry Detection from Real World Images
January 2011	Oxford University, Visual Geometry Group
June 2010	UCLA, Center for Image and Vision Sciences
Aprin 2009	National Technical University of Athens
September 2008	UCLA, Image Processing Seminar
June 2008	UC Irvine, Artificial Intelligence Seminar
May 2008	École Centrale Paris
April 2008	Berkeley, Computer Vision Group
April 2008	Caltech, Computational Vision Lab
March 2008	Rutgers University, Dept. of Computer Science
March 2008	University of Pennsylvania, GRASP Laboratory
February 2008	Johns Hopkins University, Center for Imaging Science
February 2008	Stony Brook University, Image Analysis Lab
June 2007	Lotus Hill Institute
October 2005	UCLA, Center for Image and Vision Sciences
June 2003	INRIA Sophia-Antipolis, Odyssée Group

Software Releases

Dual-Tree Branch-and-Bound for Deformable Part Models.
http://vision.mas.ecp.fr/Personnel/iasonas/dpms.html
Fractional Programming Grouping.
http://vision.mas.ecp.fr/Personnel/iasonas/contours.html
Dense Segmentation-Aware Descriptors (E. Trulls).
http://www.iri.upc.edu/people/etrulls/#code
Dense Scale-Invariant Descriptors for images and surfaces (with M. Bronstein).
http://vision.mas.ecp.fr/Personnel/iasonas/descriptors.html
Learning-based symmetry detection, code and benchmark (S. Tsogkas).
http://www.centrale-ponts.fr/personnel/tsogkas/code.html
Mid-level representations for action recognition, code and benchmark (M. Raptis).
http://vision.ucla.edu/~raptis/action_parts.html
Facade Parsing with Reinforcement Learning, code and benchmark (O. Teboul et. al.).
http://vision.mas.ecp.fr/Personnel/teboul/grapesPage/index.php
Modulation Features for Texture Analysis (with G. Evangelopoulos).
http://cvsp.cs.ntua.gr/software/texture/
Scale-Invariant Edges and Ridges.
http://vision.mas.ecp.fr/Personnel/iasonas/sketch.html

Personal

Date of Birth: 8^{th} January 1980.
Languages: Greek, English, French, German.
Affiliations: IEEE Member, Technical Chamber of Greece.

Publications

Journal articles

1 O. Teboul, I. Kokkinos, S. Loic, P. Katsourakis and N. Paragios, *"Parsing Facades with Shape Grammars and Reinforcement Learning."*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 35(7), pp. 1744-1756, 2013.

2 I. Kokkinos and A. Yuille, *"Inference and Learning with Hierarchical Shape Models."*, International Journal of Computer Vision , Vol. 92(2), pp. 201-225, 2011.

3 I. Kokkinos and P. Maragos, *"Synergy Between Image Segmentation and Object Recognition Using the Expectation Maximization Algorithm."*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 31(8), pp. 1486-1501, 2009.

4 I. Kokkinos, G. Evangelopoulos and P. Maragos, *"Texture Analysis and Segmentation Using Modulation Features, Generative Models and Weighted Curve Evolution."*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 31(1), pp. 142-157, 2009.

5 I. Kokkinos, R. Deriche, O. Faugeras and P. Maragos, *"Computational Analysis and Learning for a Biologically Motivated Model of Boundary Detection."*, Neurocomputing, Vol. 71(10-12), pp. 1798-1812, 2008.

6 I. Kokkinos and P. Maragos, *"Nonlinear Speech Analysis Using Models for Chaotic Systems."*, IEEE Trans. on Speech and Audio Processing, Vol. 13(6), pp. 1098-1109, 2005.

Double-blind, peer-reviewed conference articles (acceptance rate 20-30%)

7 E. Trulls, I. Kokkinos, A. Sanfeliu and F. Moreno, *"Dense Segmentation-Aware Descriptors"* In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2013.

8 S. Tsogkas and I. Kokkinos, *"Learning-based Symmetry Detection in Natural Images"* In Proc. European Conf. on Computer Vision (ECCV), 2012.

9 I. Kokkinos, M. Bronstein, R. Littman and A. Bronstein *"Intrinsic Shape Context Descriptors for Deformable Shapes"* In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012.

10 M. Raptis, I. Kokkinos, S. Soatto *"Discovering Discriminative Action Parts from Mid-Level Video Representations"* In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012.

11 I. Kokkinos, *"Rapid Deformable Object Detection using Dual Tree Branch and Bound"* In Proc. Neural Information Processing Systems (NIPS), 2011.

12 O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios, *"Shape Grammar Parsing via Reinforcement Learning"* In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011.

13 I. Kokkinos, *"Boundary Detection using F-measure, Filter- and Feature Boost."*, In Proc. European Conference in Computer Vision (ECCV), 2010.

14 I. Kokkinos, *"Highly Accurate Boundary Detection and Grouping."*, In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2010.

15 M. Bronstein and I. Kokkinos, *"Scale-invariant heat kernel signatures for non-rigid shape recognition."*, In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2010.

16 I. Kokkinos and A. Yuille, *"HOP: Hierarchical Object Parsing."*, In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2009.

17 I. Kokkinos and A. Yuille, *"Scale Invariance without Scale Selection."*, In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008.

18 I. Kokkinos and A. Yuille, *"Unsupervised Learning of Object Deformation Models."*, In Proc. IEEE Int'l. Conf. on Computer Vision (ICCV), 2007.

19 I. Kokkinos, P. Maragos and A. Yuille, *"Bottom-Up and Top-Down Object Detection Using Primal Sketch Features and Graphical Models."*, In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2006.

20 I. Kokkinos and P. Maragos, *"An Expectation Maximization Approach to the Synergy Between Image Segmentation and Object Categorization."*, In Proc. IEEE Int'l. Conf. on Computer Vision (ICCV), 2005.

21 I. Kokkinos, R. Deriche, P. Maragos and O. Faugeras, *"A Biologically Motivated and Computationally Tractable Model of Low- and Mid- Level Vision Tasks."*, In Proc. European Conference on Computer Vision (ECCV), 2004.

Double-blind, peer-reviewed conference and workshop articles

22 H. Boussaid, I. Kokkinos, and N. Paragios *"Rapid Mode Estimation for 3D MRI Brain Tumor Segmentation "*, Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), 2013.

23 I. Kokkinos, *"Bounding Part Scores for Rapid Detection with Deformable Part Models "*, Proc. Workshop on Parts and Attributes, in conjunction with ECCV, 2012.

24 H. Boussaid, S.Kadoury, I. Kokkinos, J.-Y. Lazennec, G. Zheng, N. Paragios, *"3D Model-based Reconstruction of the Proximal Femur from Low-dose Biplanar X-Ray Images"*, Proc. British Machine Vision Conference (BMVC), 2011.

25 A. M. Bronstein, M. M. Bronstein, B. Bustos, U. Castellani, M. Crisani, B. Falcidieno, L. J. Guibas, I. Kokkinos, V. Murino, M. Ovsjanikov, G. Patan, I. Sipiran, M. Spagnuolo, J. Sun, *"SHREC 2010: robust feature detection and description benchmark."*, Proc. EUROGRAPHICS Workshop on 3D Object Retrieval (3DOR), 2010.

26 A. M. Bronstein, M. M. Bronstein, U. Castellani, B. Falcidieno, A. Fusiello, A. Godil, L. J. Guibas, I. Kokkinos, Z. Lian, M. Ovsjanikov, G. Patan, M. Spagnuolo, R. Toldo, *"SHREC 2010: robust large-scale shape retrieval benchmark."*, Proc. EUROGRAPHICS Workshop on 3D Object Retrieval (3DOR), 2010.

27 I. Kokkinos and A. Yuille, *"Inference and Learning with Hierarchical Compositional Models."*, In Proc. 1st Int'l. Workshop on Stochastic Image Grammars, in conjunction with CVPR 2009.

28 I. Kokkinos and P. Maragos, *"A Detection-Theoretic Approach to Texture and Edge Discrimination."*, In Proc. 4th Int'l. Workshop on Texture Analysis and Synthesis, in conjunction with ICCV 2005.

29 G. Evangelopoulos, I. Kokkinos and P. Maragos, *"Advances in Variational Image Segmentation using AM-FM models: Regularized Demodulation and Probabilistic Cue Integration."*, In Proc. 3rd IEEE Variational and Level-Set Methods (VLSM) Workshop, in conjunction with ICCV 2005.

30 I. Kokkinos, G. Evangelopoulos and P. Maragos, *"Advances in Texture Analysis: Energy Dominant Component & Multiple Hypothesis Testing."*, In Proc. IEEE Int'l. Conf. on Image Processing (ICIP), 2004.

31 I. Kokkinos, G. Evangelopoulos and P. Maragos, *"Modulation-Feature based Textured Image Segmenation Using Curve Evolution."*, In Proc. IEEE Int'l. Conf. on Image Processing (ICIP), 2004.

32 V. Pitsikalis, I. Kokkinos and P. Maragos, *"Nonlinear Analysis of Speech Signals: Generalized Dimensions and Lyapunov Exponents."*, In Proc. European Conference on Speech Communication and Technology (EUROSPEECH), 2003.

33 P. Maragos, A. Dimakis and I. Kokkinos. *"Some Advances in Nonlinear Speech Modeling Using Modulations Fractals and Chaos."* In Proc. IEEE Int'l. Conf. on Digital Signal Processing, 2002.

Theses and reports

34 I. Kokkinos, M. Bronstein and A. Yuille. *Dense Scale-Invariant Descriptors for Images and Surfaces*, INRIA Research Report RR-7914, 2012.

35 I. Kokkinos. *Rapid Deformable Object Detection using Bounding-based Techniques*, INRIA Research Report RR-7940, 2012.

36 I.Kokkinos, R.Deriche, Olivier Faugeras and P.Maragos, *Towards Bridging the Gap Between Biological and Computational Segmentation*, INRIA Research Report RR-6317, 2007.

37 I. Kokkinos. *Synergy between Image Segmentation and Object Recognition using Geometrical and Statistical Computer Vision Techniques*, Ph.D. Thesis, School of Electrical and Computer Engineering, National Technical University of Athens, 2006.

38 I. Kokkinos. *Nonlinear Speech Processing Using Models for Chaotic Systems*, Diploma Thesis, School of Electrical and Computer Engineering, National Technical University of Athens, 2001.