Title
Optimal Visual Representation Engineering and Learning for Computer Vision

Author
Dong, Jingming

Publication Date
2017

UNIVERSITY OF CALIFORNIA

Los Angeles

Optimal Visual Representation
Engineering and Learning for Computer Vision

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Jingming Dong

2017

ABSTRACT OF THE DISSERTATION

Optimal Visual Representation
Engineering and Learning for Computer Vision

by

Jingming Dong
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2017
Professor Stefano Soatto, Chair

Estimating the optimal representation from sensor data has been one of the most challenging problems in computer vision research. Given a particular task, an optimal representation should contain the right information for answering queries related to the task. To be specific, such a representation should be a sufficient statistics of the data that is invariant to nuisance factors irrelevant to the task yet affecting the data. Among all the sufficient statistics, we desire the minimal that costs the least in terms of complexity. In terms of invariance, we want to achieve the maximal so that nuisance will not affect the inference at test time.

In the first part of the dissertation, we show that it is possible to build such an optimal local descriptor that is a minimal sufficient statistic of the data and is maximally invariant to certain nuisance variables in the problem of establishing feature correspondence. Given only one single image, such nuisance group is quite restricted as a single view does not afford the ability to distinguish the intrinsic properties of the scene from the extrinsics. This restriction is lifted once multiple views of the same underlying scene become available. A theoretical framework is proposed to compute an optimal multiple-view local representation with viewpoint change-induced domain deformation marginalized. In the second part, we investigate the nuisance management ability of deep neural networks in the context of image classification and show that an explicit sampling-based marginalization technique can improve its performance significantly. This is in line with the principle developed in the previous part.

Finally, we build a real-time system to estimate a visual-inertial-semantic representation of the 3D scene from both imaging and inertial measurements. Evidence from the imaging and inertial measurements are causally aggregated into the final estimate in a Bayesian filtering framework. The geometric and semantic properties of the scene do not depend on the pose and motion of the camera, and are persistent over time.

The dissertation of Jingming Dong is approved.

Joseph DiStefano III

Wei Wang

Ying Nian Wu

Stefano Soatto, Committee Chair

University of California, Los Angeles

2017

To my parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to first thank my advisor Professor Stefano Soatto for giving me the opportunity to work with him at UCLA Vision Lab. His wisdom in science has inspired me from the very beginning of my Ph. D. journey. During these years, his guidance was invaluable. The discussions between us were full of creativity and insights. His support during my study and stay in Los Angeles was unparalleled. He is not only my academic advisor, but a role model in many aspects.

Many thanks go to the members of my doctoral committee, including Professor Joseph DiStefano III, Professor Wei Wang and Professor Ying Nian Wu, for their valuable comments, suggestions and support.

I am grateful to have the opportunity to work with many talented researchers at UCLA Vision Lab. Special thanks go to Brian Taylor and Jason Meltzer, with whom I worked during my visit to the lab when I was still an undergraduate. This first experience in formal research increased my determination to start my Ph. D. at UCLA. I also would like to give special thanks to former lab members Alper Ayvaci, Jonathan Balzer and Vasiliy Karasev for their support for my research and guidance on pursuing my academic goals. Among current lab members, I would like to give special thanks to Nikolaos Karianakis, my close collaborator for these years. Shoulder to shoulder, we supported each other and overcame many difficulties over the past few years. I also would like to thank Xiaohan Fei. Together we made possible some of my most challenging works.

I also have the pleasure to work with other former and current members of the lab: Ganesh Sundaramoorthi, Taehee Lee, Chaohui Wang, Avinash Ravichandran, Damek Davis, Joshua Hernandez, Georgios Georgiadis, Virginia Estellers, Konstantine Tsotsos, Pratik Chaudhari, Yanchao Yang and Alessandro Achille. Moments we shared together will be forever memorable.

Last but not least, I would like to thank my parents, Xiujuan and Zhangxiong for their unconditional support during my years of study abroad. Their patience and support were extraordinary during all these years.

VITA

2009 Visiting Student, Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong.

2007-2011 B. Sc., Computer Science, Fudan University, Shanghai.

2014 Teaching Assistant, Computer Science, University of California, Los Angeles.

2016 Research Scientist Intern, NVIDIA.

2011-2017 Graduate Student Researcher, Computer Science, University of California, Los Angeles.

PUBLICATIONS

S. Soatto, J. Dong. “Visual Correspondence, the Lambert-Ambient Shape Space and the Systematic Design of Feature Descriptors.” *Registration and Recognition in Images and Videos*, Pages 63-93, 2014.

J. Dong and S. Soatto. “Domain-size Pooling in Local Descriptors: DSP-SIFT.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

J. Dong, N. Karianakis, D. Davis, J. Hernandez, J. Balzer and S. Soatto. “Multi-view Feature Engineering and Learning.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

S. Soatto, J. Dong and N. Karianakis. “Visual Scene Representations: Scaling and Occlusion in Convolutional Architectures.” *The International Conference on Learning Representations (ICLR) Workshop*, 2015.

N. Karianakis, J. Dong and S. Soatto. “An Empirical Evaluation of Current Convolutional Architectures’ Ability to Manage Nuisance Location and Scale Variability.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

J. Dong, I. Frosio and J. Kautz. “Learning Adaptive Parameter Tuning for Image Processing.” *ArXiv preprint arXiv:1610.09414*, 2016.

J. Dong, X. Fei and S. Soatto. “Visual Inertial Semantic Scene Representation for 3D Object Detection.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

CHAPTER 1

Introduction

Perceiving, interacting and creating the world around us display the power of human intelligence with which computer scientists want to endow machines. In the pursuit of such machine intelligence, the first question to answer is how the world is represented? Representation is the fundamental problem of machine perception, and addressing it successfully makes possible the subsequent interactions and creations. In a visual perception system, we are interested in answering questions about the *scene* which consists of *objects* each having its own *intrinsic* properties such as *shape, material* and *reflectance* and *extrinsic* properties dependent on *vantage point, illumination* and *incidence* relationship. The scene could be further complicated by dynamics and interactions between objects and the perception could be made even harder by imprecise measurement of sensors and other nuisances that are neglected in the modeling process. The questions we want to answer for perception can be the identity of the objects, their categorization, whereabouts and other high-level semantic concepts.

In this dissertation, we focus on designing and learning the optimal representation of the scene that is *invariant* to nuisance variabilities while retaining *intrinsic* properties of the former. Depending on the *tasks* or *questions* we want to answer about the scene, the definition of nuisance and information may change. For local representations in the task of correspondence or recognition, we focus on nuisances that are induced by viewpoint change and illumination change. We interpret the optimal representation as a likelihood function conditioned on the scene with nuisances marginalized. For image classification, the intrinsic property of the scene we are interested in is the semantic category the (dominant) object belongs to, regardless of it position, orientation, scale, aspect ratio and class-dependent deformations in the image. These nuisance factors should be marginalized so that the prediction of the image label is invariant to such extrinsic properties. When the task changes, for instance, from image classification to object detection, the aforementioned nuisance factors become the information that we would like to retrieve, *i.e.,* both the object semantic identity and its whereabout are required to answer the question. For 3D object detection in space, we are interested in recovering the scene properties including geometric and semantic ones which are not dependent on the viewpoint and motion of the camera, as well as time given that objects of interest are static.

1.1 Dissertation Outline

In Chapter 2, we focus on the representation at *low*-level, or local descriptor for a small neighborhood of an image region (patch). We start with the most widely adopted local feature representation – SIFT (Scale-invariant Feature Transform) descriptor, analyze its invariant properties and introduce a new operator called domain-size pooling (or DSP for short) which has deep roots in the classical sampling theory and is complementary to the spatial (translational) and rotation pooling regularly performed in SIFT and its variants. The resulting descriptor DSP-SIFT outperforms the original descriptor by a large margin in the task of correspondence in which one wants to associate points in two or more images to the same point in the physical world. In Chapter 3, we further investigate the same idea in various (local) low level descriptors, mid-level representations and high-level architectures for different tasks such as correspondence, classification and detection. Most of the existing descriptors are computed from one single image. However, as we will point out in Chapter 4, a single image does not afford the ability to distinguish intrinsic variabilities from extrinsic ones. By using multiple images, one is able to construct an approximation of the optimal local representation for the scene that is invariant to arbitrary domain deformation induced by viewpoint change. From the theoretical and empirical evidence gathered from the construction of the local level representations, we apply the similar idea to the problem of image classification in Chapter 5. We compare the explicit way of marginalizing nuisance factors to the implicit approach inherent in the certain forms of the widely-used deep neural networks, and show the effectiveness of explicit marginalization. Empirical results and discussions on the current deep architectures' ability to handle nuisance variability are provided. Chapter 6 focuses on representations in 3D space. Given imaging and inertial sensors, we build a visual-inertial-semantic scene representation for the 3D scene of interest given evidence gathered from 2D images. Camera poses are marginalized in the underlying navigation subsystem, and the intrinsic properties of the objects (*e.g.,* pose, shape and semantic identity) are causally updated in a Bayesian framework. Chapter 7 provides a summary of the findings in previous chapters.

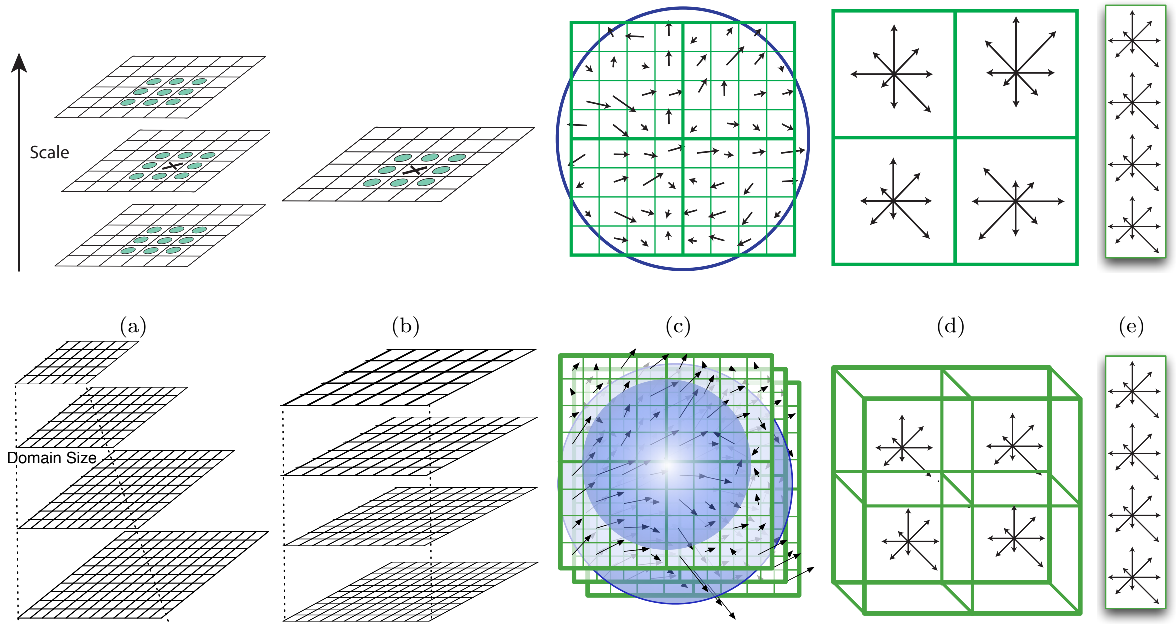
CHAPTER 2

Domain-Size Pooling in Local Descriptors: DSP-SIFT

2.1 Introduction

In this chapter, we focus on building a local descriptor for a local image patch (region) for the task of establishing correspondence in the context of wide-baseline matching. In wide-baseline matching, we are asked to find corresponding (feature) points in multiple images that are the projections of the same point in space. In the traditional feature matching pipeline, local representations are computed at a set of local frames around the feature points returned by a *detector*. A detector is a sampling mechanism which returns a set of local reference frames within the image plane with certain types of *nuisance* factors eliminated. We deal with the most common transformations such as planar similarity (translation, rotation and scaling) and illumination change in this chapter and the next, and move on to investigate more general deformations in Chapter 4. By centering the local frame at the feature point, planar translation is by construction eliminated. The same normalization (or canonization) can be applied to other nuisances such as rotation and scale by selecting a local frame with its orientation and scale *co-variant* with the transformation. A regular sampling detector returns a densely sampled frames independently of the underlying data. One can also use an adaptive sampling detector which returns a subset of frames by computing a score function whose response values are data-dependent. A representation or a *descriptor* is a deterministic function of the data computed on the local image patch rectified according to the detector. The descriptors are designed to be invariant and robust (or insensitive) to the residual nuisances that are not removed by the detector.

Local image descriptors, such as SIFT [Low04] and its variants [DT05, BTG06, CTC09],



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

Figure 2.1: In SIFT (top, recreated according to [Low04]) isolated scales are selected (a) and the descriptor constructed from the image at the selected scale (b) by computing gradient orientations (c) and pooling them in spatial neighborhoods (d) yielding histograms that are concatenated and normalized to form the descriptor (e). In DSP-SIFT (bottom), pooling occurs across different domain sizes (a): Patches of different sizes are re-scaled (b), gradient orientation computed (c) and pooled across locations *and* scales (d), and concatenated yielding a descriptor (e) of the same dimension of ordinary SIFT.

are designed to reduce variability due to illumination and vantage point while retaining discriminative power. This facilitates finding correspondence between different views of the same underlying scene. In a wide-baseline matching task on the Oxford benchmark [MS05, MTS04], nearest-neighbor SIFT descriptors achieve a mean average precision (mAP) of 27.50%, a 71.85% improvement over direct comparison of normalized grayscale values. Other datasets yield similar results [MP07]. In this chapter, we show that a simple modification of SIFT, obtained by pooling gradient orientations across different domain sizes ("scales"), in addition to spatial locations, improves it by a considerable margin, also outperforming the neural network descriptors learned from a large training dataset. We call the resulting descriptor "domain-size pooled" SIFT, or DSP-SIFT.

Pooling across different domain sizes is implemented in few lines of code, can be applied to any histogram-based method (Sect. 2.3), and yields a descriptor of the same size that outperforms the original essentially uniformly (Fig. 2.4). Yet combining histograms of images of different sizes is counterintuitive and seemingly at odds with the teachings of scale-space theory and the resulting established practice of *scale selection* [Lin98] (Sect. 2.1.1). It is, however, rooted in classical sampling theory and anti-aliasing. Sect. 2.2 describes *what* we do, Sect. 2.3 *how* we do it, and Sect. 2.5 *why* we do it. Sect. 2.4 validates our method empirically.

2.1.1 Related work

A single, un-normalized cell of the "scale-invariant feature transform" SIFT [Low04] and its variants [BTG06, CTC09, DT05] can be written compactly as a formula [DKD15, VF10]:

$$h_{\texttt{SIFT}}(\theta|I,\hat{\sigma})[x] = \int \mathcal{N}_\epsilon\left(\theta - \angle\nabla I(y)\right)\mathcal{N}_{\hat{\sigma}}(y-x)d\mu(y) \tag{2.1}$$

where I is the image restricted to a square domain, centered at a location $x \in \Lambda(\hat{\sigma})$ with size $\hat{\sigma}$ in the lattice Λ determined by the response to a difference-of-Gaussian (DoG) operator across all locations and scales (SIFT *detector*). Here $d\mu(y) \doteq \|\nabla I(y)\|dy$, θ is the independent variable, ranging from 0 to 2π, corresponding to an orientation histogram bin of size ϵ, and $\hat{\sigma}$ is the *spatial pooling scale*. The kernel $\mathcal{N}_\epsilon$ is bilinear of size ϵ and $\mathcal{N}_{\hat{\sigma}}$ separable-bilinear of size $\hat{\sigma}$ [VF10], although they could be replaced by a Gaussian with standard deviation $\hat{\sigma}$ and an *angular Gaussian* with dispersion parameter ϵ. The SIFT descriptor is the concatenation of 16 cells (2.1) computed at locations $x \in \{x_1, x_2, \dots, x_{16}\}$ on a 4×4 lattice Λ, and normalized.

The spatial pooling scale $\hat{\sigma}$ and the size of the image domain where the SIFT descriptor is computed $\Lambda = \Lambda(\hat{\sigma})$ are *tied* to the photometric characteristics of the image, since $\hat{\sigma}$ is derived from the response of a DoG operator on the (single) image.[1] Such a response depends on the *reflectance* properties of the scene and *optical characteristics* and *resolution* of the sensor,

[1] Approaches based on "dense SIFT" forgo the detector and instead compute descriptors on a regular sampling of locations and scales (Fig. 2.12). However, no existing dense SIFT method performs domain-size pooling.

neither of which is related to the size and shape of co-visible (corresponding) regions. Instead, how large a portion of a scene is visible in each corresponding image(s) depends on the *shape* of the scene, the *pose* of the two cameras, and the resulting visibility (*occlusion*) relations. Therefore, we propose to *untie* the size of the domain where the descriptor is computed ("scale") from photometric characteristics of the image, departing from the teachings of scale selection (Fig. 2.9). Instead, we use basic principles of classical sampling theory and *anti-aliasing* to achieve robustness to domain size changes due to occlusions (Sect. 2.5).

Pooling is commonly understood as the combination of responses of feature detectors or descriptors at nearby locations, aimed at transforming the joint feature representation into a more usable one that preserves important information (intrinsic variability) while discarding irrelevant detail (nuisance variability) [BPL10, JHD12]. However, precisely how pooling trades off these two conflicting aims is unclear and mostly addressed empirically in end-to-end comparisons with numerous confounding factors. Exceptions include [BPL10], where intrinsic and nuisance variability are combined and abstracted into the variance and distance between the means of scalar random variables in a binary classification task. For more general settings, the goals of reducing nuisance variability while preserving intrinsic variability is elusive as a *single image* does not afford the ability to separate the two [DKD15].

An alternate interpretation of pooling as anti-aliasing [SC14] clearly highlights its effects on intrinsic and nuisance variability: Because one cannot know what portion of an object or scene will be visible in a test image, a scale-space ("semi-orbit") of domain sizes ("receptive fields") should be marginalized or searched over ("max-out"). Neither can be computed in closed-form, so the semi-orbit has to be sampled. To reduce complexity, only a small number of samples should be retained, resulting in undersampling and aliasing phenomena that can be mitigated by anti-aliasing, with quantifiable effects on the sensitivity to nuisance variability. For the case of histogram-based descriptors, anti-aliasing planar translations consists of spatial pooling, routinely performed by most descriptors. Anti-aliasing visibility results in *domain-size aggregation*, which no current descriptor practices. This interpretation also offers a way to quantify the effects of pooling on discriminative (reconstruction) power directly, using classical results from sampling theory, rather than indirectly through an end-to-end classification experiment that may contain other confounding factors.

Domain-size pooling can be applied to a number of different descriptors or convolutional architectures. We illustrate its effects on the most popular, SIFT, in this chapter and validate its applications to other descriptors and architectures in the next chapter. However, we point out that proper marginalization requires the availability of multiple images of the same scene, and therefore cannot be performed in a single image (Chapter 4). While most local image descriptors are computed from a single image, exceptions include [DKD15, LS11b]. Of course, multiple images can be "hallucinated" from one, but the resulting pooling operation can only achieve invariance to modeled transformations.

In neural network architectures, there is evidence that abstracting spatial pooling hierarchically, *i.e.,* aggregating nearby responses in feature maps, is beneficial [BPL10]. This process could be extended by aggregating across different neighborhood sizes in feature space. To the best of our knowledge, the only architecture that performs some kind of pooling across scales is [SOP07], although the justification provided in [BRP09] only concerns translation within each scale. The same goes for [BM11], where pooling (low-pass filtering) is only performed within each scale, and not across scales. Other works learn the regions for spatial pooling, for instance [JHD12, SVZ14b], but still restrict pooling to within-scale, similar to [LeC12], rather than across scales as we advocate.

We distinguish *multi-scale methods* that concatenate descriptors computed *independently at each scale*, from *cross-scale pooling*, where statistics of the image at different scales are combined directly in the descriptor. Examples of the former include [HMZ12], where ordinary SIFT descriptors computed on domains of different size are assumed to belong to a linear subspace, and [SVZ14b], where Fisher vectors are computed for multiple sizes and aspect ratios and spatial pooling occurs within each level. Also bag-of-word (BoW) methods [SZ03], as mid-level representations, aggregate different low level descriptors by counting their frequency after discretization. Typically, vector quantization or other clustering technique is used, each descriptor is associated with a cluster center ("word"), and the frequency of each word is recorded in lieu of the descriptors themselves. This can be done for domain size, by computing different descriptors at the same location, for different domain sizes, and then counting frequencies relative to a dictionary learned from a large training dataset (Sect. 2.4.4).

Aggregation across time, which may include changes of domain size, is advocated in [HLB11], but in the absence of formulas it is unclear how this approach relates to our work. In [FCN12], weights are shared across scales, which is not equivalent to pooling, but still establishes some dependencies across scales. MTD [LS11a] appears to be the first instance of pooling across scales, although the aggregation is global in scale-space with consequent loss of discriminative power. Most recently, [GWG14] advocates the same but in practice space-pooled VLAD descriptors obtained at different scales are simply concatenated. Also [BM01] can be thought of as a form of pooling, but the resulting descriptor only captures the mean of the resulting distribution. In addition, [TH14] exploits the possibility of estimating the proper scales for nearby features via scale propagation but still no pooling is performed across scales.

2.2 Domain-Size Pooling

If SIFT is written as (2.1), then DSP-SIFT is given by

$$h_{\texttt{DSP}}(\theta|I)[x] = \int h_{\texttt{SIFT}}(\theta|I,\sigma)[x]\mathcal{E}_s(\sigma)d\sigma \quad x \in \Lambda \tag{2.2}$$

where $s > 0$ is the size-pooling scale and $\mathcal{E}$ is an exponential or other unilateral density function. The process is visualized in Fig. 2.1. Unlike SIFT, that is computed on a scale-selected lattice $\Lambda(\hat{\sigma})$, DSP-SIFT is computed on a *regularly sampled* lattice Λ. Computed on a different lattice, the above can be considered as a recipe for DSP-HOG [DT05]. Computed on a tree, it can be used to extend deformable-parts models (DPM) [FMR08] to DSP-DPM. Replacing $h_{\texttt{SIFT}}$ with other histogram-based descriptor "X" (for instance, SURF [BTG06]), the above yields DSP-X. Applied to a hidden layer of a convolutional network, it yields a DSP-CNN, or DSP-Deep-Fisher-Network [SVZ13]. These variants are explored in the next chapter. The details of the implementation are in Sect. 2.3.

While the implementation of DSP is straightforward, its justification is less so. We report the summary in Sect. 2.5. In Sect. 2.4 we compare DSP-SIFT to alternate approaches. Motivated by the experiments of [MTS04, MP07] that compare local descriptors, we choose SIFT as a paragon and compare it to DSP-SIFT on the standard benchmark [MTS04]. Motivated by [FDB14] that compares SIFT to both supervised and unsupervised CNNs trained on ImageNet and Flickr respectively on the same benchmark [MTS04], we submit DSP-SIFT to the same protocol. We also run the test on the new synthetic dataset introduced by [FDB14], that yields the same qualitative assessment.

Clearly, domain-size pooling of under-sampled semi-orbits cannot outperform fine sampling, so if we were to retain all the scale samples instead of aggregating them, performance would further improve. However, computing and matching a large collection of SIFT descriptors across different scales would incur significantly increased computational and storage costs. To contain the latter, [HMZ12] assumes that descriptors at different scales populate a linear subspace and fit a high-dimensional hyperplane. The resulting Scale-less SIFT (SLS) outperforms ordinary SIFT as shown in Fig. 2.6. However, the linear subspace assumption breaks when considering large scale changes, so SLS is outperformed by DSP-SIFT despite the considerable difference in (memory and time) complexity.

2.3 Implementation and Parameters

Following other evaluation protocols, we use *Maximally Stable Extremal Regions* (MSER) [MCM02] to detect candidate regions, affine-normalize, re-scale and align them to the dominant orientation. For a detected scale $\hat{\sigma}$, DSP-SIFT samples $N_{\hat{\sigma}}$ scales within a neighborhood $(\lambda_1 \hat{\sigma}, \lambda_2 \hat{\sigma})$ around it. For each scale-sampled patch, a single-scale un-normalized SIFT descriptor (2.1) is computed on the SIFT scale-space octave corresponding to the sampled scale σ. By choosing $\mathcal{E}_s$ to be a uniform density, these raw histograms of gradient orientations at different scales are accumulated and normalized[2] (2.2). Fig. 2.2(a) shows the mean average precision (defined in Sect. 2.4.2) for different domain size pooling ranges. Improvements

[2] We follow the practice of SIFT [Low04] to normalize, clamp and re-normalize the histograms, with the clamping threshold set to 0.067 empirically.

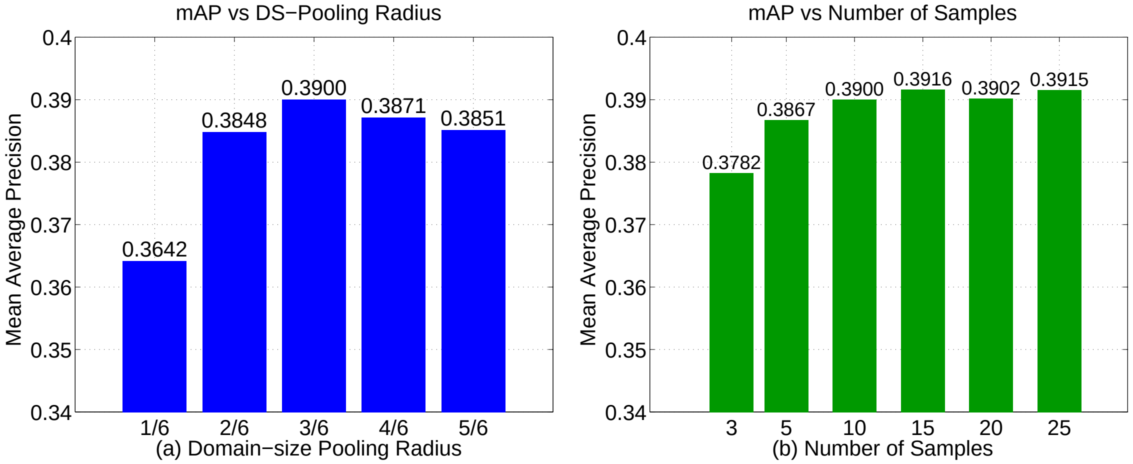


Figure 2.2: Mean Average Precision for different parameters. (a) shows that mAP changes with the radius s of DS pooling. The best mAP is achieved at $\hat{s} = \hat{\sigma}/2$; (b) shows mAP as a function of the number of samples used within the best range $(\hat{\sigma} - \hat{s}, \hat{\sigma} + \hat{s})$.

are observed as soon as more than one scale is used, with diminishing return: Performance decreases with domain size pooling radius exceeding $\hat{\sigma}/2$. Fig. 2.2(b) shows the effect of the number of size samples used to construct DSP-SIFT. Although the more samples the merrier, three size samples are sufficient to outperform ordinary SIFT, and improvement beyond 10 samples is minimal. Additional samples do not further increase the mean average precision, but incur more computational cost. In the evaluation in Sect. 2.4, we use $\lambda_1 = 1/6, \lambda_2 = 4/3$ and $N_{\hat{\sigma}} = 15$. These parameters are empirically selected on the Oxford dataset [MS05, MTS04].

2.4 Validation

As a baseline, the RAW-PATCH descriptor (named following [FDB14]) is the unit-norm grayscale intensity of the affine-rectified and resized patch of a fixed size (91 × 91).

The standard SIFT, which is widely accepted as a paragon [MS05, MP07], is computed using the VLFeat library [VF10]. Both SIFT and DSP-SIFT are computed on the SIFT scale-space corresponding to the detected scales. Instead of mapping all patches to an arbitrarily user-defined size, we use the area of each selected and rectified MSER region to determine the octave level in the scale-space where SIFT (as well as DSP-SIFT) is to be computed.

Scale-less SIFT (SLS) is computed using the source code provided by the authors [HMZ12]: For each selected and rectified patch, the standard SIFT descriptors are computed at 20 scales from a scale range of $(0.5, 12)$, and the standard PCA subspace dimension is set to 8, yielding a final descriptor of dimension 8256 after a subspace-to-vector mapping.

To compare DSP-SIFT to a convolutional neural network, we use the top-performer in [FDB14], an unsupervised model pre-trained on 16000 natural images undergoing 150 transformations each (total 2.4M). The responses at the intermediate layers 3 (CNN-L3) and 4 (CNN-L4) are used for comparison, following [FDB14]. Since the network requires input patches of fixed size, we tested and report the results on both 69×69 (PS69) and 91×91 (PS91) as in [FDB14].

Although no direct comparison with Multiscale Template Descriptors (MTD) [LS11a] is performed, SLS can be considered as dominating it since it uses all scales without collapsing them into a single histogram. The derivation in Sect. 2.5 suggests, and empirical evidence in Fig. 2.2(a) confirms, that aggregating the histogram across *all* scales significantly reduces discriminative power. Sect. 2.4.4 compares DSP-SIFT to a BoW which pools SIFT descriptors computed at different sizes at the same location.

2.4.1 Datasets

The Oxford dataset [MS05, MTS04] comprises 40 pairs of images of mostly planar scenes seen under different pose, distance, blurring, compression and lighting. They are organized into 8 categories undergoing increasing magnitude of transformations. While routinely used to evaluate descriptors, this dataset has limitations in terms of size and restriction to mostly planar scenes, modest scale changes, and no occlusions. Fischer *et al.* [FDB14] recently introduced a dataset of 400 pairs of images with more extreme transformations including zooming, blurring, lighting change, rotation, perspective and nonlinear transformations.

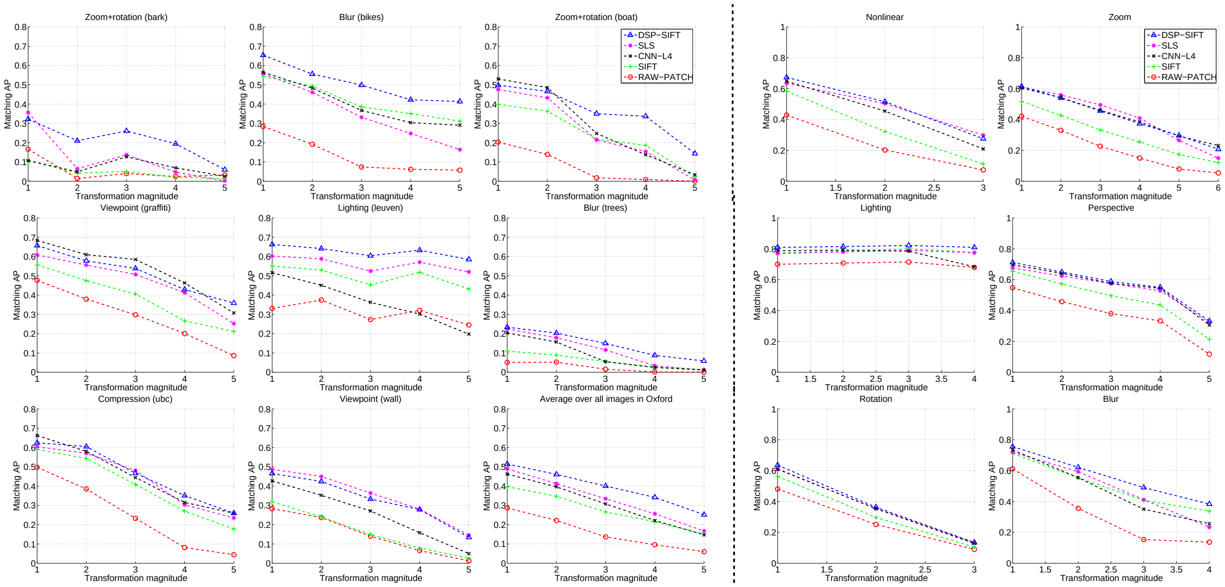


Figure 2.3: Average Precision for different magnitude of transformations. The left 9 panels show (AP) for increasing magnitude of the 8 transformations in the Oxford dataset [MS05]. The mean AP over all pairs with corresponding amount of transformation are shown in the middle of the third row. The right 6 panels show the same for Fischer's dataset [FDB14].

2.4.2 Metrics

Following [MS05], we use precision-recall (PR) curves to evaluate descriptors. A *match* between two descriptors is called if their Euclidean distance is less than a threshold τ_d. It is then labeled as a *true positive* if the area of intersection over union (IoU) of their corresponding MSER-detected regions is larger than 50%. Both datasets provide ground truth mapping between images, so the overlapping is computed by warping the first MSER region into the second image and then computing the overlap with the second MSER region. *Recall* is the fraction of true positives over the total number of correspondences. *Precision* is the percentage of true matches within the total number of matches. By varying the distance threshold τ_d, a PR curve can be generated and *average precision* (AP, *a.k.a area under the curve*, AUC) can be estimated. The average of APs provides the *mean average precision* (mAP) scores used for comparison.

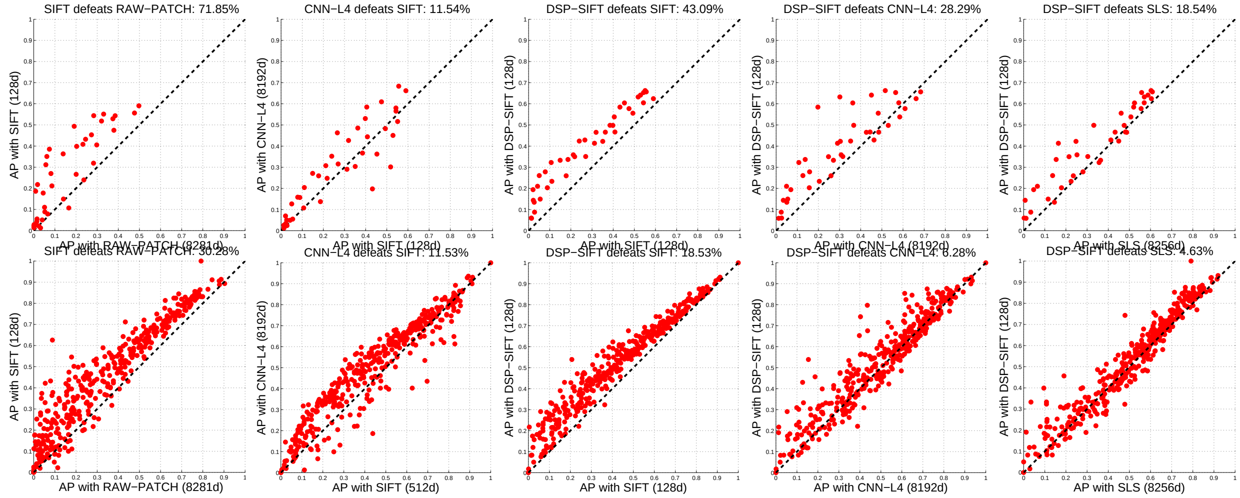


Figure 2.4: Head-to-head comparisons. Similarly to [FDB14], each point represents one pair of images in the Oxford (top) and Fischer (bottom) datasets. The coordinates indicate average precision for each of the two methods under comparison. SIFT is superior to RAW–PATCH, but is outperformed by DSP-SIFT and CNN-L4. The right two columns show that DSP-SIFT is better than SLS and CNN-L4 despite the difference in dimensions (shown in the axes). The relative performance improvement of the winner is shown in the title of each panel.

2.4.3 Comparison

Fig. 2.3 shows the behavior of each descriptor for varying degree of severity of each transformation. DSP-SIFT consistently outperforms other methods when there are large scale changes (zoom). It is also more robust to other transformations such as blur, lighting and compression in the Oxford dataset [MTS04], and to nonlinear, perspective, lighting, blur and rotation in Fischer's [FDB14]. DSP-SIFT is not at the top of the list of all compared descriptors in viewpoint change cases, although "viewpoint" is a misnomer as MSER-based rectification accounts for most of the viewpoint variability, and the residual variability is mostly due to interpolation and rectification artifacts. The fact that DSP-SIFT outperforms CNN in nearly all cases in Fischer's dataset is surprising, considering that the neural network is trained by augmenting the dataset using similar types of transformations.

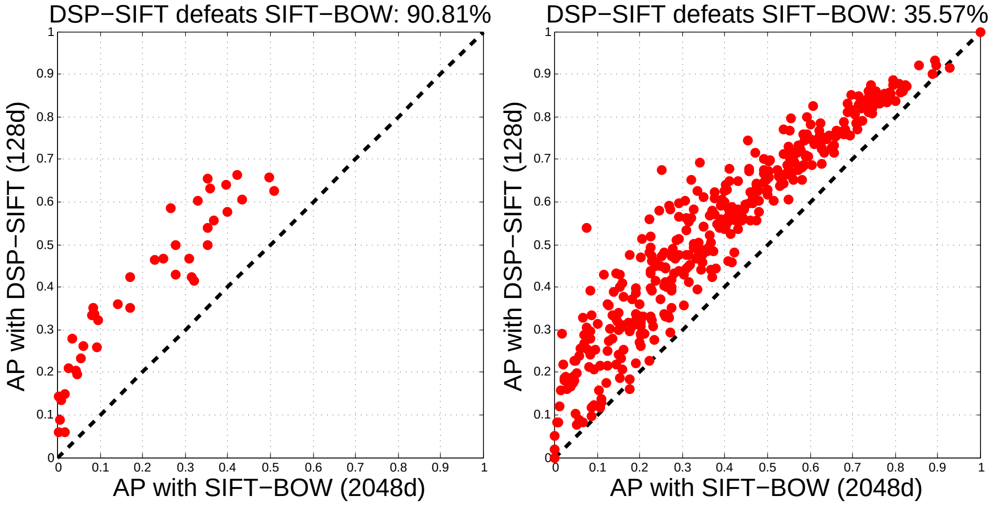


Figure 2.5: DSP-SIFT vs. SIFT-BOW. Similarly to Fig. 2.4, each point represents one pair of images in the Oxford (left) and Fischer (right) datasets. The coordinates indicate average precision for each of the two methods under comparison. The relative performance improvement of the winner is shown in the title of each panel. DSP-SIFT outperforms SIFT-BOW by a wide margin on both datasets.

Fig. 2.4 shows head-to-head comparisons between these methods, in the same format of [FDB14]. DSP-SIFT outperforms SIFT by 43.09% and 18.54% on Oxford and Fischer respectively. Only on two out of 400 pairs of images in Fischer dataset does domain-size pooling negatively affect the performance of SIFT, but the decrease is rather small. DSP-SIFT improves SIFT on every pair of images in the Oxford dataset. The improvement of DSP-SIFT comes without increase in dimension. In comparison, CNN-L4 achieves 11.54% and 11.53% improvements over SIFT by increasing dimension 64-fold. On both datasets, DSP-SIFT also consistently outperforms CNN-L4 and SLS despite its lower dimension.

2.4.4 Comparison with Bag-of-Words

To compare DSP-SIFT to BoW we computed SIFT at 15 scales on concentric regions with dictionary sizes ranging from 512 to 2048, trained on over 100K SIFT descriptors computed on samples from ILSVRC-2013 [DDS09]. To make the comparison fair, the same 15 scales

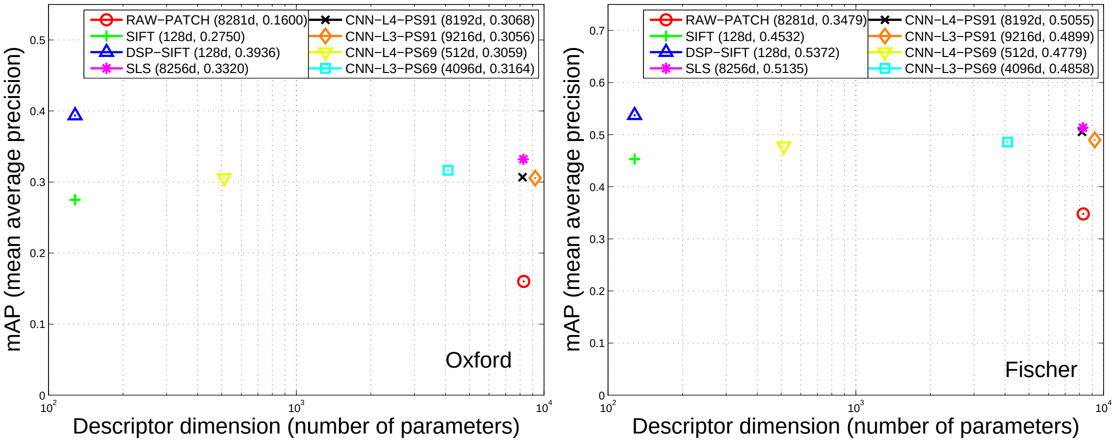


Figure 2.6: Complexity-Performance Tradeoff. The abscissa is the descriptor dimension shown in log-scale, the ordinate shows the mean average precision.

are used to compute DSP-SIFT. By doing so, the only difference between these two methods is *how* to pool across scales rather than *what* or *where* to pool. In SIFT-BOW, pooling is performed by encoding SIFTs from nearby scales using the quantized visual dictionary, while DSP-SIFT combines the histograms of gradient orientations across scales directly. To compute similarity between SIFT-BOWs, we tested both the intersection kernel and ℓ_1 norm, and achieved a best performance with the latter at 20.62% mAP on Oxford and 39.63% on Fischer. Fig. 2.5 shows the direct comparison between DSP-SIFT and SIFT-BOW with the former being a clear winner.

2.4.5 Complexity and Performance Tradeoff

Fig. 2.6 shows the complexity (descriptor dimension) and performance (mAP) tradeoff. Table 2.1 summarizes the results. In Fig. 2.6, an "ideal" descriptor would achieve mAP = 1 by using the smallest possible number of bits and land at the top-left corner of the graph. DSP-SIFT has the same lowest complexity as SIFT and is the best in mAP among all the descriptors. Looking horizontally in the graph, DSP-SIFT outperforms all the other methods at a fraction of complexity. SLS achieves the second best performance but at the cost of a 64-fold increase in dimension. In general, the performance of CNN descriptors is worse

Method	Dim.	mAP	
		Oxford	Fischer
SIFT	**128**	.2750	.4532
DSP-SIFT	**128**	**.3936**	**.5372**
CNN-L4-PS69	512	.3059	.4779
SIFT-BOW	2048	.2062	.3963
CNN-L3-PS69	4096	.3164	.4858
CNN-L4-PS91	8192	.3068	.5055
SLS	8256	.3320	.5135
RAW-PATCH	8281	.1600	.3479
CNN-L3-PS91	9216	.3056	.4899

Table 2.1: Summary of complexity (dimension) and performance (mAP) for all descriptors sorted in order of increasing complexity. The lowest complexities and the best performances are highlighted in bold. We also report mAP for CNN descriptors computed on 69 × 69 patches as in [FDB14]. The fourth row shows comparison with a bag-of-words of SIFT descriptors computed at the same location but different domain sizes, described in detail in Sect. 2.4.4.

than DSP-SIFT but, interestingly, their mAPs do not change significantly if the network responses are computed on a resampled patch of size 69 × 69 to obtain lower dimensional descriptors.

2.4.6 Comparison with SIFT on Larger Domain Sizes

Descriptors computed on larger domain sizes are usually more discriminative, up to the point where the domain straddles occluding boundaries (Fig. 2.7). When using a detector, the size of the domain is usually chosen to be a factor of the detected scale, which affects performance in a way that depends on the dataset and the incidence of occlusions. In our experiments, this parameter (dilation factor) is set at 3, following [MS05], and we note that DSP-SIFT is less sensitive than ordinary SIFT to this parameter. Since DSP-SIFT aggregates domains of various sizes (smaller and larger) around the nominal size, it is important to ascertain whether the improvement in DSP-SIFT comes from size pooling, or simply from including larger domains. To this end, we compare DSP-SIFT by pooling domain sizes from 1/6th through 4/3rd of the scale determined by the detector, to a single-size descriptor computed at the largest size (SIFT-L). This establishes that the increase in performance of DSP-SIFT over ordinary SIFT comes from pooling across domain sizes, not just by picking larger domain sizes. In the example in Fig. 2.8, the largest domain size yields an even worse performance than the detection scale (Fig. 2.8(b)). In a more complex scene where the test images exhibit occlusion, this will be even more pronounced as there is a tradeoff between discriminative power (calling for a larger size) and the probability of straddling an occlusion (calling for a smaller size).

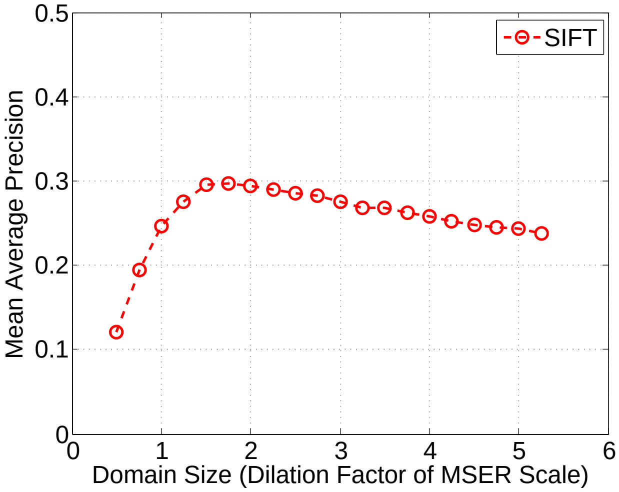


Figure 2.7: The discriminative power of a descriptor (*e.g.,* mAP of SIFT) increases with the size of the domain, but so does the probability of straddling an occlusion and the approximation error of the imaging model implicit in the detector/descriptor. This effect, which also depends on the base size, is most pronounced when occlusions are present, but is present even on the Oxford dataset, shown above.

2.5 Derivation

In this section we describe the trace of the derivation of DSP-SIFT, which is reported in Appendix A. Crucial to the derivation is the interpretation of a descriptor as a likelihood

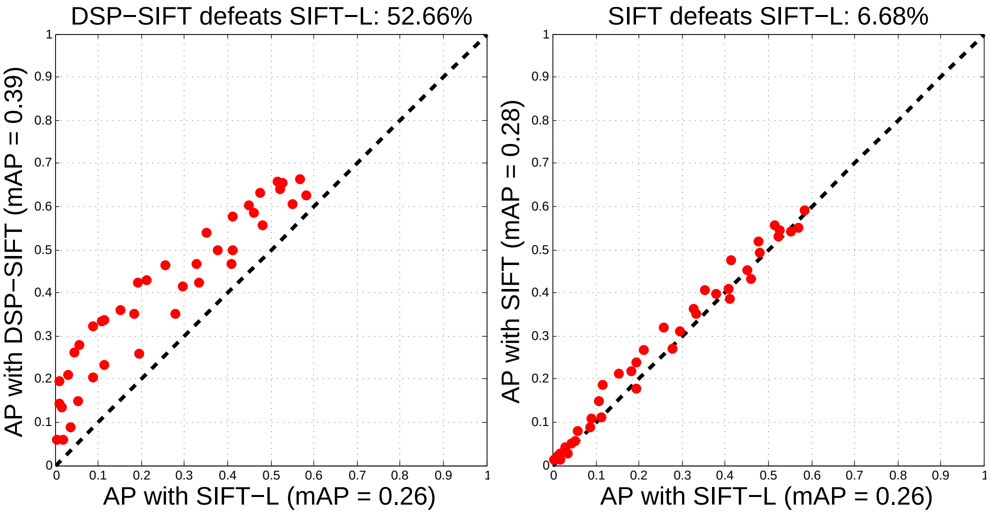


Figure 2.8: DSP-SIFT vs. SIFT-L. Similarly to Fig. 2.4, each point represents one pair of images in the Oxford dataset. The coordinates indicate average precision for each of the two methods under comparison. The relative performance improvement of the winner is shown in the title of each panel. 2.8(a) shows that DSP-SIFT outperforms SIFT computed at the largest domain size. This shows that the improvement of DSP-SIFT comes from the pooling across domain sizes rather than choosing a larger domain size. 2.8(b) shows that choosing a larger domain size actually decreases the performance on the Oxford dataset.

function [SC14].

1. The likelihood function of the scene given images is a minimal sufficient statistic of the latter for the purpose of answering questions on the former [Bah54]. Invariance to nuisance transformations induced by (semi-)group actions on the data can be achieved by representing orbits, which are maximal invariants [Sha98]. The planar translation-scale group can be used as a crude first-order approximation of the action of the translation group in space (viewpoint changes) including scale change-inducing translations along the optical axis. This draconian assumption is implicit in most single-view descriptors.

2. Comparing (semi-)orbits entails a continuous search (non-convex optimization) that has to be discretized for implementation purposes. The orbits can be sampled adaptively, through the use of a co-variant detector and the associated invariant descriptor, or regularly – as

customary in classical sampling theory.

3. In adaptive sampling, the *detector* should exhibit high sensitivity to nuisance transformations (*e.g.,* small changes in scale should cause a large change in the response to the detector, thus providing accurate scale localization) and the *descriptor* should exhibit small sensitivity (so small errors in scale localization cause a small change in the descriptor). Unfortunately, for the case of SIFT (DoG detector and gradient orientation histogram descriptor), the converse is true.

4. Because correspondence entails search over samples of each orbit, time complexity increases with the number of samples. Undersampling introduces structural artifacts, or "aliases," corresponding to topological changes in the response of the detector. These can be reduced by "anti-aliasing," an averaging operation. For the case of (approximations of) the likelihood function, such as SIFT and its variants, anti-aliasing corresponds to *pooling.* While spatial pooling is common practice, and reduces sensitivity to translation parallel to the image plane, scale pooling – which would provide insensitivity to translation orthogonal to the image plane – and domain-size pooling – which would provide insensitivity to small changes of visibility, are not. This motivates the introduction of DSP-SIFT, and the rich theory on sampling and anti-aliasing could provide guidelines on what and how to pool, as well as bounds on the loss of discriminative power coming from undersampling and anti-aliasing operations.

2.5.1 Generative and Discriminative Power

According to the above derivation, one would expect that images can be sampled from SIFT or DSP-SIFT, as interpreted as a likelihood function. Fig. 2.11 shows that this is indeed the case. We compute descriptors at the detected SIFT locations and scales, and then draw samples of gradient orientations from the histograms. Local image patches are then reconstructed from their gradient. With these patches put back to the detected locations, the object is clearly visible (Fig. 2.11(b)). Given multiple views of the same object, *e.g.,* from Moreels' dataset [MP07], one can also learn a likelihood of configurations in addition to the

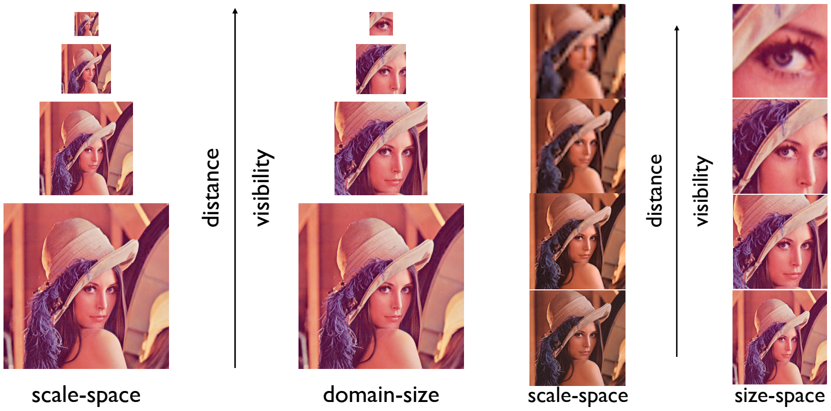


Figure 2.9: Scale-space vs. Size-space. Scale-space refers to a continuum of images obtained by smoothing and downsampling a base image. It is relevant to searching for correspondence when the distance to the scene changes. Size-space refers to a scale-space obtained by maintaining the same scale of the base image, but considering subsets of it of variable size. It is relevant to searching for correspondence in the presence of occlusions, so the size (and shape) of co-visible domains are not known.

appearance. For instance, a simple model can be a Gaussian over the locations of tracked features over multiple views. Fig. 2.11(c) and (d) show the instances sampled from both geometric and photometric likelihood. These sampled images resemble what have been visualized by others in the context of CNNs learned from millions of images of the same object category [MV15].

Fig. 2.10 shows the images reconstructed from regularly sampled DSP-SIFT. We follow the procedure of [MV15] with dense DSP-SIFT replacing SIFT. It is remarkable to notice that the reconstructed images are sharper than one would have expected by thinking that pooling across different domains will lead to a blurry reconstruction. These qualitative results suggest that the loss in discriminative power is very limited after domain-size pooling.

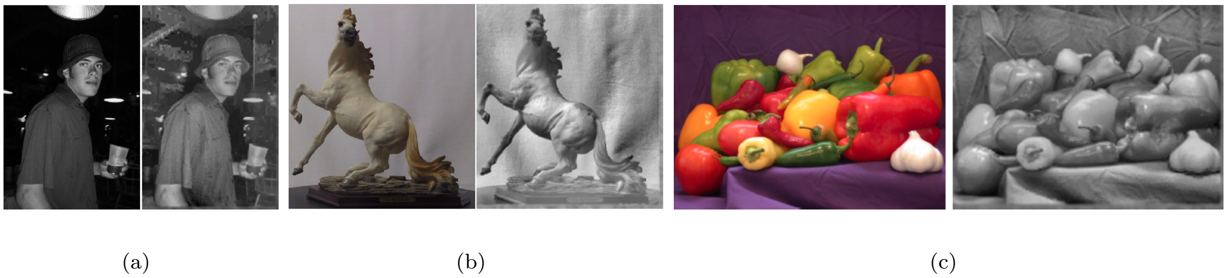


Figure 2.10: Images reconstructed from dense DSP-SIFT. Each pair shows the original and the reconstructed images from dense DSP-SIFT.

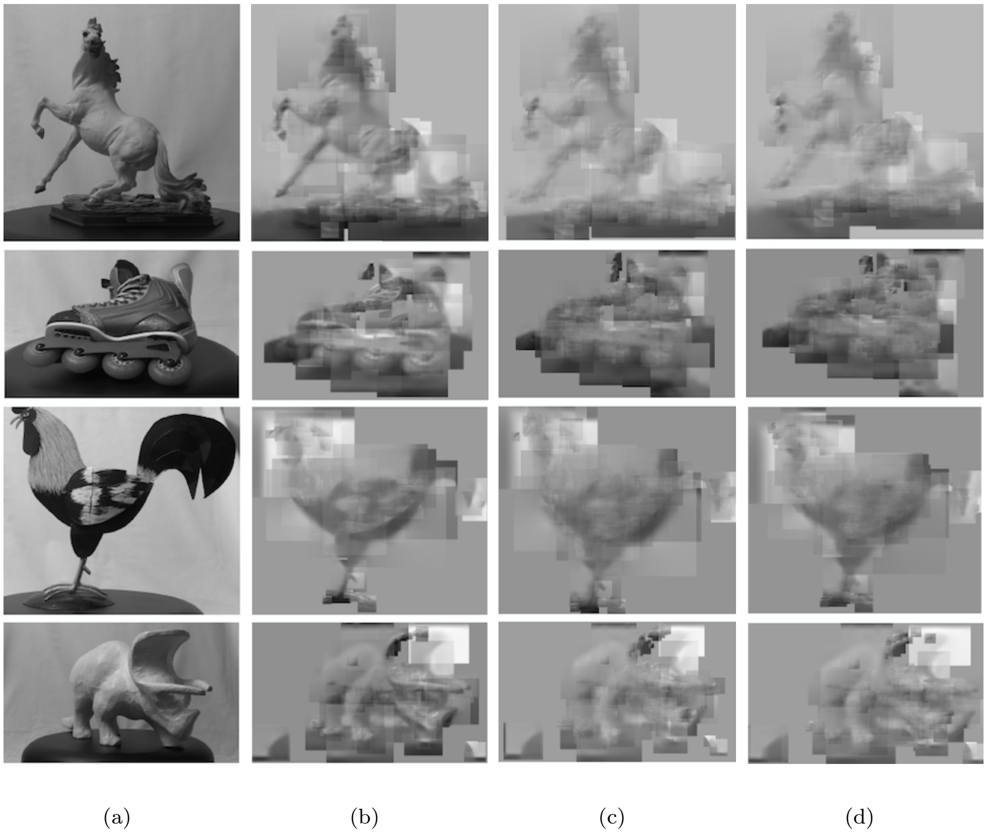


Figure 2.11: Images sampled from a learned descriptor-configuration model. (a) Original image. (b) Image sampled from a single view. (c)-(d) show two instances sampled from templates learned from multiple views.

2.6 Discussion

Image matching under changes of viewpoint, illumination and partial occlusions is framed as a hypothesis testing problem, which results in a non-convex optimization over continuous nuisance parameters. The need for efficient test-time performance has spawned an industry of engineered descriptors, which are computed locally so the effects of occlusions can be reduced to a binary classification (co-visible, or not). The best known is SIFT, which has been shown to work well in a number of independent empirical assessments [MS05, MP07], that however come with little analysis on *why* it works, or indications on how to improve it. We have made a step in that direction, by showing that SIFT can be derived from sampling considerations, where spatial binning and pooling are the result of anti-aliasing operations. However, SIFT and its variants only perform such operations for planar translations, whereas our interpretation calls for anti-aliasing domain-size as well. Doing so can be accomplished in few lines of code and yields significant performance improvements. Such improvements even place the resulting DSP-SIFT descriptor above a convolutional neural network (CNN), that had been recently reported as a top performer in the Oxford image matching benchmark [FDB14]. Of course, we are not advocating replacing large neural networks with local descriptors. Indeed, there are interesting relations between DSP-SIFT and convolutional architectures, explored in [SC14, SDK14].

Domain-size pooling, and regular sampling of scale “unhinged” from the spatial frequencies of the signal is divorced from scale selection principles, rooted in scale-space theory, wavelets and harmonic analysis. There, the goal is to reconstruct a signal, with the focus on photometric nuisances (additive noise). In our case, the size of the domain where images correspond depends on the three-dimensional shape of the underlying scene, and visibility (occlusion) relations, and has little to do with the spatial frequencies or “appearance” of the scene. Thus, we do away with the linking of domain size and spatial frequency (“uncertainty principle”, Fig. 2.12).

DSP can be easily extended to other descriptors, such as HOG, SURF, CHOG, including those supported on structured domains such as DPMs [FMR08], and to network architectures such as convolutional neural networks and scattering networks [BM11], opening the door to multiple extensions of the present work. In addition, a number of interesting open theoretical questions can now be addressed using the tools of classical sampling theory, given the novel interpretation of SIFT and its variants introduced in this chapter.

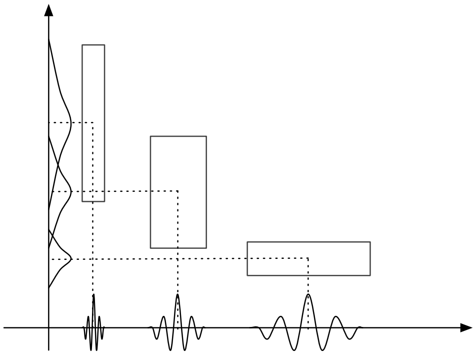


Figure 2.12: The "uncertainty principle" links the size of the domain of a filter (ordinate) to its spatial frequency (abscissa): As the data is analyzed for the purpose of compression, regions with high spatial frequency must be modeled at small scale, while regions with smaller spatial frequency can be encoded at large scale. When the task is correspondence, however, the size of the co-visible domain is independent of the spatial frequency of the scene within. While approaches using "dense SIFT" forgo the detector and compute descriptors at regularly sampled locations and scales, they perform spatial pooling by virtue of the descriptor, but fail to perform pooling across scales, as we propose.

CHAPTER 3

Sampling and Pooling in Local, Mid-Level and High-Level Representations

3.1 Introduction

In Chapter 2, detector/descriptor ensembles are interpreted as *sampled representations*, whereby the detector selects elements of a group of transformations to which we desire invariance (translation, in the simplest case), and the descriptor is a function defined on that group, that has additional desirable properties, for instance being insensitive to other nuisance variability such as contrast transformations, while maintaining discriminative power.

Extensions of classical sampling theory [SZ05] then suggest that one should *anti-alias* the descriptors, *i.e.,* average them against local group transformations. What should be stored at the samples is *not* the value of the function (descriptor) at that sample, but rather a local average of group transformations around that sample, a process known as *anti-aliasing.* The results in Chapter 2 indicate that this may be indeed beneficial.

However, even considering the same detector, different descriptors have different averaging (pooling) mechanisms, so the benefits anti-aliasing gives to SIFT may not extend to other descriptors. Considering the magnitude of such benefits as reported in Chapter 2, however, it is important for us to test whether anti-aliasing with respect to an undesired group of transformations improves performance in other low- and mid-level vision methods. Accordingly, our goal is to test the effects of anti-aliasing on a variety of low- and mid-level vision methods, including other histogram-based descriptors such as HOG [DT05], SURF [BTG06], DAISY [TLF10], binarized descriptors such as BRIEF [CLS10], mid-level data structures such as DPMs [FMR08] and BoWs [KWB14], and global hierarchical descriptors such as Convolutional Neural Networks [KSH12, SZ15] and Scattering Transform [BM11].

We test our hypothesis empirically in Sect. 3.6 by using common benchmarks for wide-baseline matching such as the Oxford [MS05, MTS04] and PASCAL VOC [EGW10], as well as more recently introduced benchmarks such as Fischer's [FDB14] and Balzer's [BS13]. Concerning the chosen group, we limit ourselves to the isotropic translation-scale semi-group, as done in Chapter 2. Extension to anisotropic scales (changes of aspect ratio) and more general finite-dimensional Lie groups such as similarity, affine, projective are conceptually straightforward but beyond our scope in this dissertation. To explore the role of confounding factors in the evaluation, we also test descriptors relative to different choices of *base size*, *i.e.,* the area in pixels to which all detected regions are mapped in order to compute the descriptor. This tests quantization and interpolation artifacts that were not explored in Chapter 2.

The conclusion of our investigation is that the benefits from anti-aliasing vary by descriptor, but are present in varying degree in each of them. The biggest aggregate benefit is observed in histogram-based descriptors, where anti-aliasing can be interpreted as marginalization of local deformations, and the smallest is observed in binarized descriptors, where the metric of the embedding space does not lend itself to straightforward averaging and concepts such as anti-aliasing do not extend in a straightforward manner.

3.2 Related Work

Many local feature detectors and descriptors have been proposed in the past two decades. The most well-known is SIFT [Low04] which detects keypoints as local extrema of Difference of Gaussian response, which is an approximate of Laplacian of Gaussian, once normalized is invariant to scale changes [Lin98]. A SIFT descriptor is computed at the same level where the detector fires. Other descriptors vary by how they build a scale-space and how they compute different descriptors in the octave levels [BTG06, LCS11], but there is a common trait among all these local descriptors developed so far. They are all computed from a cropped region centered at the detected spatial location with a fixed *domain size* related to the detected scale. The detector is designed to be sensitive to scale changes, thus having high localization accuracy, which in turn grants the descriptor the property of "scale-invariant" even if the descriptor *per se* is not invariant to scale changes. The residual deformation induced by the viewpoint change is handled by the insensitivity of the descriptor to such nuisance.

One way to handle the residual deformation is through *spatial pooling* which is widely used in low level descriptor design and also popular among convolutional neural networks. Spatial pooling aggregates statistics from nearby locations to achieve invariance (or "insensitivity") to translational group nuisance. However, as to be discussed in Sect. 3.3, the assumption behind the philosophy of combining scale selection by the detector with residual nuisance handling in descriptor breaks when there exist occlusions or the viewpoint change induced self-occlusions. In hindsight, this is a natural consequence of such detector/descriptor ensembles because a detector, by construction, is incapable of handling the residual deformations, otherwise there is no need for a descriptor.

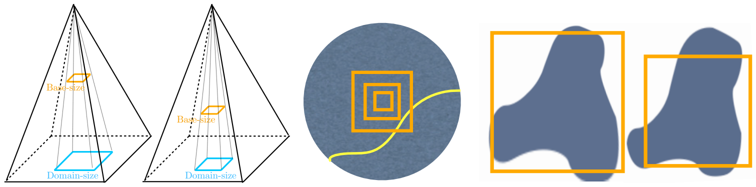
Two lines of approaches have been developed to handle scale selection failure. Multi-scale methods include [SVZ14b] which computes Fisher vectors at multiple scales and aspect ratios and spatial pooling of these features happens within each level. [HMZ12] assumes that descriptors, *e.g.,* SIFT, computed from all scales live in a low dimensional space and a Principal Component Analysis (PCA) representation encodes descriptors at all scales. On the other end of the spectrum, [DS15] proposes pooling SIFT computed from nearby scales around detection to achieve anti-aliasing of domain sizes. Also, Bag of Words (BoW) [KWB14, SZ03] can be considered as pooling descriptors computed at different locations and scales. By enforcing descriptors to be computed from nearby locations or scales, BoW can be thought of a way of spatial or scale pooling. In network architectures, [PML11] proposes averaging responses of filters of neighboring sizes and [KSJ14] performs cross scale pooling by filtering resized images or intermediate responses and averaging responses. Cross scale pooling has also been used in the direct image-to-image matching algorithm [YLS15] where local patches are averaged around nearby points in the affine space which includes both translational, scale and aspect ratio changes.

3.3 Scales, Domain Sizes and Base Sizes

"Scale" has been used interchangeably in many contexts [Lin98, KD84]. In automatic scale selection [Low04], scale refers to the standard deviation of the Difference of Gaussian (DoG) used to detect blob structures in the image. Once a scale is detected, a descriptor is computed from a neighborhood around the detected location. The size of the neighborhood, also known as *domain*, where the descriptor is computed is usually a deterministic function (typically a multiple) of the detected scale. To deal with scale change, descriptors are not computed at the native image resolution, but on the octave in scale-space where it is detected. The neighborhood is downsampled from the native resolution to the octave level. We call the size on the octave *base size*, which is also a function of the DoG detected scale. Therefore in scale selection, domain size and base size are tied with the photometric property of the image irrespective of the viewpoint and co-visibility relation between different views.

In terms of discriminative power, one wants to choose a larger domain size where descriptors are computed. On the other hand, the possibility of straddling occlusion boundaries and therefore include mixtures of unrelated objects in the descriptor calls for the domain size to be small ("locality of descriptors"). At test time, which portions of the transformed image are co-visible with the base image is unknown, which requires domain size selection independent of the detection scale (Fig. 3.1(b)). Matching descriptors requires searching over size-space, which carries a high cost. This can be partly mitigated by under-sampling, and accordingly anti-aliasing the domain size (DS), a technique developed in the previous chapter. Domain-size pooling around a detected scale corresponds to anti-aliasing the descriptors around the (adaptively) sampled size determined by the detectors.

Even if there is no occlusion, computing descriptors on cropped image patches associated with the detection scale can cause self-occlusion, or obscuration, by the cropped window, illustrated in Fig. 3.1(c). The DoG isotropic filter returns an extreme value with respect to location and scale that co-varies with the translational component of the vantage point, but with poor scale localization accuracy. If even a small percentage of the patch is outside the domain where the descriptor is computed, regions from other parts of the image intrude in



(a) (b) (c)

Figure 3.1: Scale, Domain-size and Base-size. 3.1(b), domain-size has to be searched over at test time when occlusion is present. 3.1(c), self-occlusion caused by viewpoint change in the *cropped* image patch. 3.1(a), patches with different domain sizes are matched to the same base sizes where descriptors are to be computed.

the domain, thus making the resulting descriptors highly sensitive to errors in scale selection. Self-occlusions prevail in almost every local descriptor as long as they are computed from a local region cropped from the image. This issue is addressed by domain-size pooling.

To achieve invariance to scale, descriptors of different domain sizes have to be computed after first downsampling the patch to the same base size (Fig. 3.1(a)). This is critical for descriptors computed by finite difference, *e.g.,* histogram-based descriptors. Ideally one should compute statistics using the metric unit defined on the "scene" not on the image (pixel). Alternatively, both patches can be resampled to the base size where descriptors are computed. Too small of a base size leads to a loss in discriminative power because of the loss in details, while too large a base size forces every feature with a domain size less than it to be computed at the native image resolution, thus losing scale invariance. Base size is the tradeoff between discriminative power and the degree of scale invariance. In Sect. 3.6, we compare domain-size pooled descriptors with the single-scale descriptors at different domain sizes and base sizes to provide insight in the effects of DS pooling in various low- and mid-level descriptors.

3.4 Pooling Domains

Domain-size pooling, as described in Chapter 2, can be decomposed into two major steps – a pooling step followed by a normalization step. A more detailed recipe to apply DSP to a single scale descriptors around a detected scale is i) patches with different domain sizes are sampled within a local neighborhood of the detected scales, ii) patches are re-sized to the base size, iii) single-scale descriptors are computed from each sample individually and averaged with a base measure kernel and iv) properly normalized. In terms of a histogram based descriptor, DSP corresponds to marginalization over scales (Sect. 3.4.1). Sect. 3.4.2 and 3.4.3 extends the idea further to binary and network descriptors.

3.4.1 Histogram-based Descriptors

Histograms of gradient orientation are widely used in the design of local descriptors [BTG06, DT05, Low04, TLF10]. Typically image patches are divided into cells or grids on a rectangle or a circular lattice, and a histogram of gradient orientation is built in each cell as

$$h(\theta|I_\sigma) = \int \kappa_\epsilon(\theta - \angle\nabla I_\sigma(x))\kappa_\sigma(y-x)\|\nabla I_\sigma(x)\|dy \tag{3.1}$$

a formula reported by [VF10], where θ is a free variable of gradient orientation over a unit circle, x is the detected feature location, and $I_\sigma(x)$ is the image in the scale space corresponding to the detected scale $\sigma(x)$. The kernel κ_ϵ controls the quantization of the orientations around the unit circle and kernel κ_σ is the spatial anti-aliasing which averages nearby pixel locations. Various descriptors vary by what kernels to use and how histograms are combined. For example, HOG [DT05] and SIFT [Low04] uses bilinear function in both orientations and spatial locations, but differs in how they are normalized. SIFT normalizes the whole descriptors by ℓ_2 norm, while HOG combines nearby cells into blocks and normalizes each cell by the norm of the enclosing blocks. Most recently, we proposed pooling domains in SIFT [DS15] simply by averaging

$$h(\theta|I) = \int h(\theta|I_\sigma)\mathcal{E}(\sigma)d\sigma \tag{3.2}$$

over the scale (semi-)group with some prior $\mathcal{E}$. This can be extended to any other histogram-based descriptors, in principle. However, due to the various form of combining histograms from different cells, normalization methods (globally, block-wise, per-cell) and thresholding ("clamping"), it is not clear how much improvement can be expected from descriptors other than SIFT. To this end, we empirically evaluate these variants in Sect. 3.6. Also, among all these histogram-based descriptors, DAISY computes each cell at different scales according to the deviation of the cell center to the patch center. Cells that are far away from the center are smoothed with a Gaussian kernel of larger standard deviation. This is equivalent to computing histograms on a larger domain size but downsampled to a smaller base size similar to the center cells. In this sense, DAISY can be considered a way of pooling gradient orientations across different domain sizes, with a spatially-varying prior $\mathcal{E}$. But it is different from DSP in that the pooling comes from different locations (cell centers) and the resulting histograms are concatenated rather than aggregated (averaged).

3.4.2 Binary Descriptors

In this section, we show how to extend domain-size pooling to binary descriptors by interpreting each bit of the descriptor as a probability conditioned on the domain size. Then similar marginalization as in Eq. (3.2) can be applied directly. A binary descriptor is a 0-1 string constructed by comparing the intensity ordering between pairs of pixels $\{(x_i, y_i)\}_{i=1}^N$ within the patch. Computed on a single domain size, the i-th bit of the binary descriptor can be thought of a Bernoulli probability

$$p(I_\sigma(x_i) > I_\sigma(y_i)|\sigma) \tag{3.3}$$

which takes value 1 if $I_\sigma(x_i) > I_\sigma(y_i)$ or 0 otherwise. The probability is conditioned on the domain-size ("scale") σ. By marginalizing against a base measure $\mathcal{E}(\sigma)$ on the domain-size, a domain-size pooled version of Eq. (3.3) can be computed as

$$\int p(I_\sigma(x_i) > I_\sigma(y_i)|\sigma)\mathcal{E}(\sigma)d\sigma \tag{3.4}$$

Note that the above can be applied to most binary descriptors such as [AOV17, CLS10, LCS11, RRK11] to obtain DSP-BRIEF, DSP-ORB, DSP-BRISK and DSP-FREAK. For instance, the simplest binary descriptor is BRIEF [CLS10] which randomly samples a set of pairs of pixels within the patch and compares the intensity order of two pixels. Pooling over domain-sizes produces DSP-BRIEF which improves the matching performance of the standard BRIEF uniformly on different datasets as shown in Sect. 3.6. However, note that

DSP-BRIEF (also other DSP binary descriptors) is no longer binary. To binarize it, we threshold each bit at 0.5 and call the binarized descriptor DSP-BRIEF-BIN. But after the binarization, the interpretation of Eq. (3.4) no longer stands. In Sect. 3.6, we test both DSP versions against the original.

3.4.3 Convolutional Network Architectures

Convolutional network architectures convolve an image (or patch) with a fixed filter bank. These filters are either hand-picked (engineered) or trained (learned) from a vast amount of data. Regardless of how these filters are obtained, the domain size (receptive field) of the filters are pre-defined. These receptive fields are typically small (*e.g.,* 11×11 in the first layer of the convolutional neural network in [KSH12]) to lower the risk of straddling occlusion boundaries. One can expect that anti-aliasing by domain-size pooling in the individual filter response can further improve the performance of these networks. On the other hand, these convolutional networks are usually trained in an end-to-end fashion and there are conjectures that by stacking multiple convolutional and pooling layers, invariance (or "insensitivity") to planar similarity, as well as visibility, can be achieved. We defer the investigation of the nuisance management ability of the end-to-end learned neural networks until Chapter 5 where DSP has also been applied to CNNs to yield DSP-CNN. In this chapter, in order to remove the bias in filter learning in evaluating the effects of domain-size pooling, we consider a special convolutional network architecture – the Scattering Transform [BM11] which convolves a patch with a Gabor filter bank at different rotations and dilations, takes the modulus of the responses, and applies an averaging operator to yield the scattering coefficients. This is repeated to produce coefficients at different layers and the final descriptor is the concatenation of these coefficients. Domain-size pooling can be applied to each intermediate layers of the network by convolving the patch with various sized filters and averaged, which is a costly operation. Alternatively, one can apply DSP to the image plane as in SIFT and other local descriptors by sampling nearby domain sizes of the image, propagating each size-sampled patch through the entire network and averaging the responses. This is motivated by the fact that, as shown by [BM11], most of the energy in the scattering coefficients is exhausted after the second layer, and therefore there would be diminishing return in manipulating the convolutional structure beyond that. Since the responses are normalized at each layer of the network, there is no post-normalization. We call the resulting descriptor DSP-SC.

3.5 Evaluation Criteria

3.5.1 Datasets

Following Chapter 2, we use Oxford [MTS04] and Fischer's [FDB14] datasets for descriptor evaluation. Empirical results show that descriptors achieve consistent performance on both Oxford and Fischer datasets [DS15, FDB14]. Datasets with more complicated scenes include [BS13, MP07] with the latter has more complicated pose relations between cameras and objects, thus more (self-)occlusion phenomena. [BS13] has 15 synthetic objects from MeshLab, wrapped with random images as textures. A training video is provided for each object and 5 testing views are captured from different viewpoints uncovered by the training trajectory. Both DSP and single scale descriptors under evaluation are computed on a single image, so we only use the first frame of the training video as the base image. We use all three datasets in our evaluation, and as the results shown in Sect. 3.6, occlusion play a critical role in the behavior of descriptor performance.

3.5.2 Metrics

Following Chapter 2, we use *Precision-Recall Curve* (PR-Curve) to evaluate the performance of descriptors. *Precision* is the fraction of *true matches* among all declared *matches*. A *match* is claimed between two descriptors if they are nearest neighbors in terms of descriptor distance and the distance is smaller than a threshold τ_d. A match is labeled as *true match* if two features correspond according to the ground truth. *Recall* is the the ratio of the number of true matches over the total number of correspondences. By varying the distance threshold τ_d, a PR-Curve can be drawn and the *Average Precision* (AP) can be computed by the area under PR-Curve. For the entire dataset, an average of AP, *Mean Average Precision* (mAP) is reported and used to compare domain-size pooled and single size descriptors.

3.6 Comparison and Discussion

We evaluate descriptors by pairs – DSP-X *vs.* X where X includes histogram-based HOG (both UoCTTI [FMR08] and DT [DT05] variants), SURF [BTG06], DAISY [TLF10], binary representative BRIEF, and a convolutional architecture scattering network (SC). We use MSER [MCM02] to select regions from both base and transformed images in Oxford and Fischer's [FDB14] datasets and SIFT [Low04] detector on Balzer's [BS13]. Alternate detectors are also tested and results are presented in Fig. 3.6, 3.7.

3.6.1 Effect of Domain Size

Fig. 3.2 shows Mean Average Precision as a function of domain size (as a dilation of the detected scale). In general, domain-size pooling improves HOG-TTI, HOG-DT, SURF and BRIEF, but not DAISY on Oxford dataset. This is because DAISY computes histograms at different scales at different locations and concatenates them into the same descriptor. This grants DAISY a certain degree of robustness to the scale localization inaccuracy of the detection. However, when there are more occlusions, *e.g.,* on Balzer dataset, DSP-DAISY improves DAISY by a large margin when domain size is large. An improvement is also observed in the larger Fischer dataset shown in Fig. 3.5. When there are more self-occlusions presented in the dataset due to the more complex-shaped objects, there is a tradeoff between discriminative power (towards larger size) and being local to avoid occlusion (towards smaller size). In the bottom row of Fig. 3.2, all the curves have a turning point at a certain dilation factor of the detected scale. This is not observed in Oxford dataset because it does not contain occlusions. The improvements of the DSP versions are much more obvious when the domain size is large which leads to a higher risk of hitting the occlusion boundaries (most obvious in the right half of the curves). Fig. 3.2 also shows that the benefit of DSP is from *pooling* across nearby domain-sizes, not just using a larger domain, as it is clear from the curve that the mAP of a DSP descriptor with a small domain size can be much higher than its plain version computed at a (*single*) larger domain. Despite the large improvement over BRIEF, DSP-BRIEF is no longer a binary descriptor. We also tested DSP-BRIEF-BIN

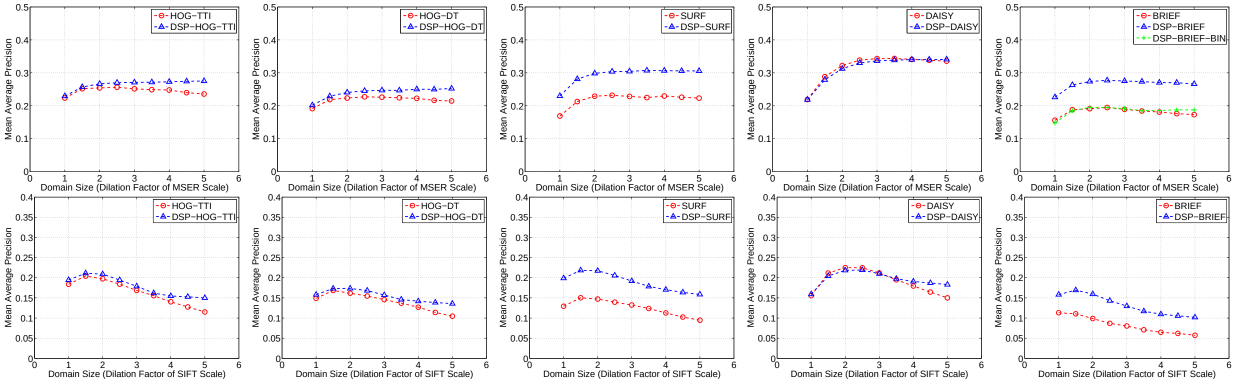


Figure 3.2: Mean Average Precision vs. Domain-Size. Top: Oxford dataset with MSER detector. Bottom: Balzer dataset with SIFT detector.

which is thresholded at 0.5 to turn DSP-BRIEF into binary. However, this makes DSP-BRIEF-BIN almost the same as BRIEF at a single scale. This is because binarization can be thought of scale-selection (selecting a scale from nearby scales, and test intensity ordering there), so the binarized descriptor has the performance similar to BRIEF which tests ordering at the detected scale. Therefore, we forgo further testing of DSP-BRIEF-BIN.

3.6.2 Effect of Base Size

Fig. 3.3 shows the performance of each descriptor as a function of base size. Base size affects the degree of improvement of DSP-X over X. Both DSP-HOGs perform best on a small base size (21×21). This explains why small base sizes are commonly used in dense HOG computation to describe the whole image. A small base size captures details of the image at each location. As discussed in Sect 3.3, due to scale changes, a small base size is preferred so that larger patches can be downsampled to the base size and descriptors computed there. Patches with sizes smaller than the base size have to be computed at the native image resolution. Upsampling to large sizes causes aliasing introduced in interpolation. This is most obvious when there are large scale changes in the image, but it is already observable here (*e.g.,* both HOGs). Also, it is interesting to see that SURF and BRIEF are less sensitive to the selection of base size, and therefore the improvement of the DSP version is consistent over different base sizes.

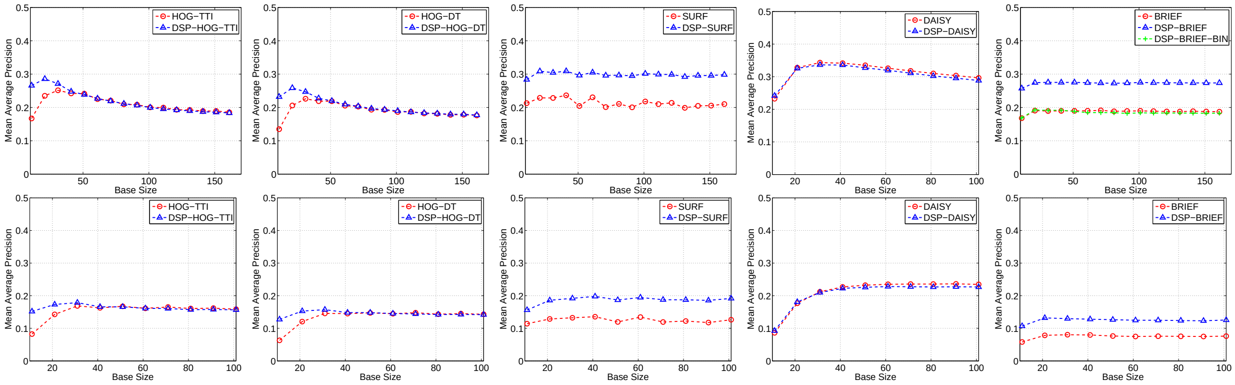


Figure 3.3: Mean Average Precision vs. Base-Size. Top: Oxford dataset with MSER detector. Bottom: Balzer dataset with SIFT detector.

3.6.3 Effect of DSP Radius and Sampling

Fig. 3.4 shows the performance of each DSP descriptor with varying domain-size pooling radius shown as a fraction of the domain size. The domain size and base size used for each descriptor are selected from Fig. 3.2 and 3.3. The domain-size is 3× the detected MSER scale, and the base size is 21×21 for DSP-HOG-TTI and DSP-HOG-DT and 31×31 for other descriptors shown. The maximum performance is highlighted with the marker. In general, the best pooling radius is between 1/3rd and 2/3rds of the domain size with specific values varying for different descriptors. Obviously the curves also show that pooling over all scales decrease the performance. In the right panel, we further fix the DS pooling radius and test the effect of the number of size sampled. It is observed that even 3 samples used, the performance is boosted from that of the corresponding single scale descriptor. It is not surprising that the more samples, the better the performance. But increase after 10 samples are marginal, so there is a tradeoff between computational complexity and performance. Figs. 3.5 shows head-to-head comparison between DSP-X and X with the best parameters obtained on Oxford and tested on Fischer's dataset.

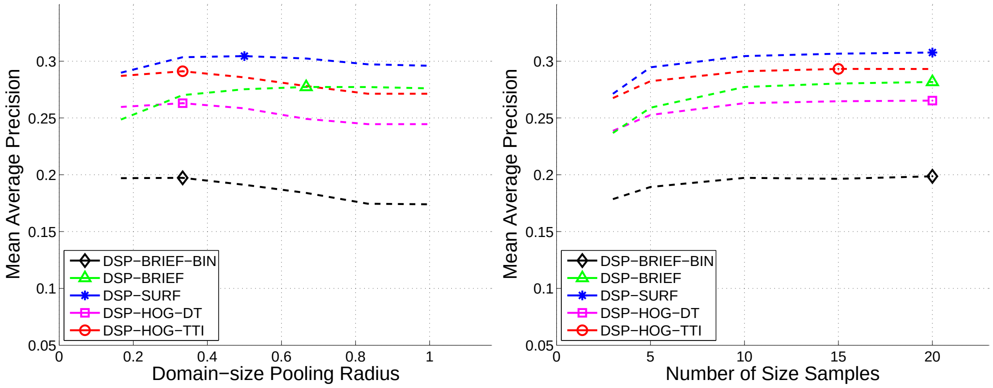


Figure 3.4: Domain-size Pooling Radius and the Number of Size Samples (Oxford). The parameters that achieve the best performance are highlighted with markers.

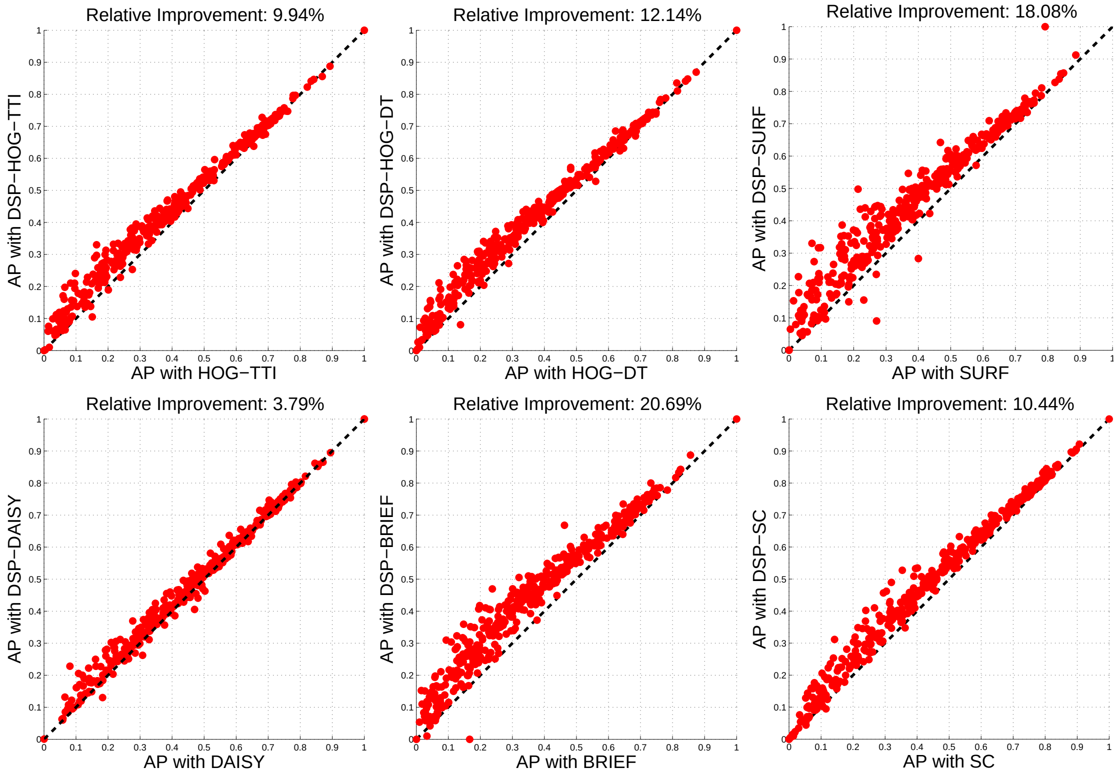
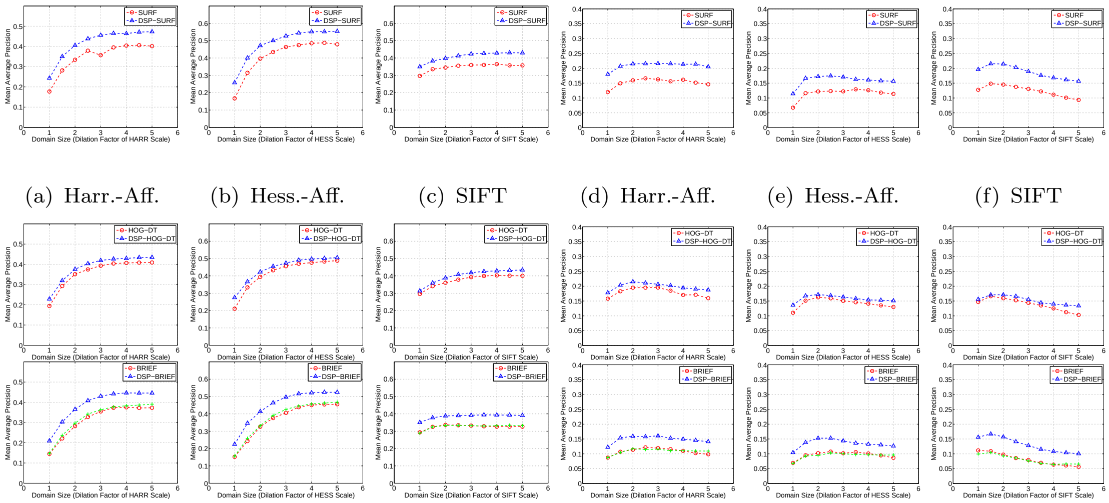


Figure 3.5: Head-to-head comparisons. Each point in the plot shows one pair of images from Fischer dataset. The horizontal axis shows the Average Precision of the single-scale descriptor (X), and the vertical axis shows that of DSP-X. The relative improvement of DSP-X over X is shown in the title of each panel.

3.6.4 Effect of Detector

Fig. 3.6 and 3.7 shows the performance of several DSP descriptors with different feature detectors. The choice of detector affects the performance of both DSP-X and X. However, regardless of the detector, the margin between DSP-X and X remains similar. The anti-aliasing effect of DSP is effective as long as there exist occlusions (including self-occlusions caused by cropping) in image (sub)regions and undersampling over the domain-size space in scale selection or detection.

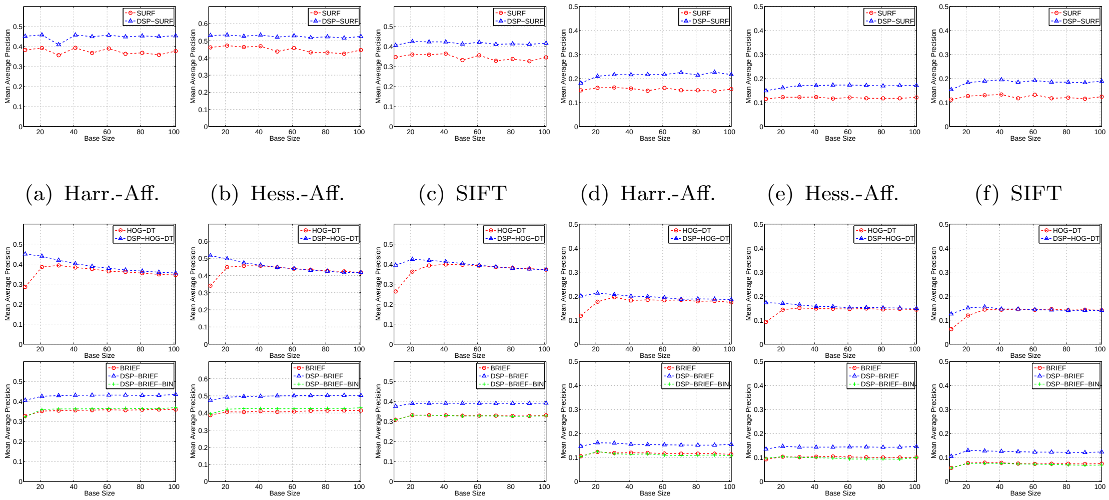


(a) Harr.-Aff. (b) Hess.-Aff. (c) SIFT (d) Harr.-Aff. (e) Hess.-Aff. (f) SIFT

Figure 3.6: Effect of different detectors. Each curve is mAP as a function of Domain-size. Left 3 columns: Oxford, Right 3 columns: Balzer's dataset. Top row: DSP-SURF, Middle: DSP-HOG-DT and Bottom: DSP-BRIEF.

3.7 Mid-level Descriptors

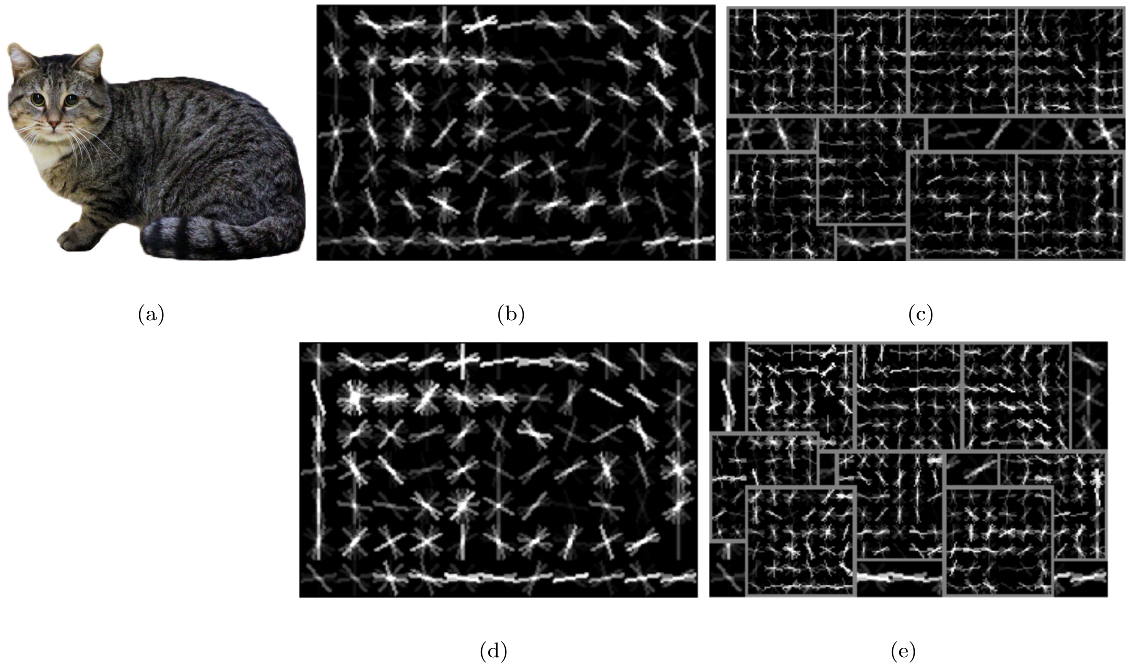
Descriptors are not only used for correspondence, but also heavily used as the first building block of many object detection model. For instance, HOG-TTI has been proposed and used in deformable-part models (DPMs) [FMR08]. Although there are so many components in the end-to-end detection system that they become confounders in evaluating the benefits of using better descriptors, nevertheless, we plug in DSP-HOG-TTI into the same DPM architecture, thus the name DSP-DPM for simplicity, and test object recognition performance on PASCAL



(a) Harr.-Aff. (b) Hess.-Aff. (c) SIFT (d) Harr.-Aff. (e) Hess.-Aff. (f) SIFT

Figure 3.7: Effect of different detectors. Each curve is mAP as a function of Base-size. Left 3 columns: Oxford, Right 3 columns: Balzer's dataset. Top row: DSP-SURF, Middle: DSP-HOG-DT and Bottom: DSP-BRIEF.

VOC 2007 detection challenge [EGW10]. We sampled 10 domain sizes ranging from 0.5σ to 1.5σ where σ is the original size used for HOG-TTI computation. By average pooling of the HOG-TTIs computed from each domain sizes, we obtain a dense DSP-HOG-TTI response for the whole image. They are used to train the deformable model for each object in the challenge. The results are reported in Table 3.1. DSP-DPM wins 16 out of 20 object categories in terms of mean average precision. Among the winning categories, we found they are mostly classes of animals whose configurations are more variable and thus more likely to incur occlusions. In other cases when objects are less "deformable", the performance of the two DPMs is close. Fig. 3.8 shows two models for the Cat class learned by DPM and DSP-DPM respectively. The benefits of DSP can be visually appreciated where the boundary of the cat is enhanced in the root template, and details are more regular in the part template, compared to the noisier appearance in the standard DPM. At test time, the original DPM samples scale very densely (10 levels between two octaves), which is observed to be critical to achieve a good performance [FMR08]. When the scale-orbit is (regularly) densely sampled, the effect of anti-aliasing of DSP decreases as expected. Nevertheless, the improvement of DSP-DPM over the standard version is nontrivial. On the other hand, at test time, the original DPM samples scale very densely (10 levels between two octaves), which is observed to be critical to achieve a good performance [FMR08]. When the scale space is (regularly) densely sampled, the effect of anti-aliasing of DSP decrease as expected.



(a) (b) (c)

(d) (e)

Figure 3.8: Learned templates from DPM and DSP-DPM. The top row shows the "root" and the "part" templates learned by the standard DPM. The bottom row shows the same learned by DSP-DPM. The geometry of the cat (*e.g.,* head) is more visible in the templates learned via domain-size pooling, compared to the original DPM.

To add one more mid-level representation example, we extend DSP to Bag-of-Words model for classification tested on four datasets: Flickr Material Dataset (FMD) [SRA09], Caltech 101 (CAL) [FFP04], MIT Scene (SCN) [QT09] and PASCAL VOC 2007 classification challenge (VOC) [EGW10]. DSP-SIFT is used in DSP-BoW and standard SIFT in BoW. We fix the DSP radius and the number of size samples for all four datasets. Table 3.2 shows that DSP improves plain BoW on CAL, SCN and VOC. The categories in FMD are materials, most of which are textures. DSP does not improve BoW when the same parameters are used as in the other three datasets which consist of common objects. However, if we reduce the pooling radius by half, the average precision of DSP-BoW rises to .3383.

	Aero.	Bike	Bird	Boat	Bottle	Car	Cat	Chair	Cow	Bus	Table	Dog	Horse	Motor	Person	Sheep	Sofa	Train	TV	Plant	mAP
DPM	.3221	.5814	.0835	.1212	.2931	**.5733**	.2221	.2101	.2447	.5241	**.2803**	.1215	.6078	.4604	.4020	.1745	**.3248**	.4243	**.4470**	.1246	.3271
DSP-DPM	**.3526**	**.5951**	**.1070**	**.1377**	**.3022**	.5679	**.2668**	**.2281**	**.3042**	**.5246**	.2760	**.1329**	**.6149**	**.4634**	**.4127**	**.2005**	.3193	**.4538**	.4318	**.1345**	**.3413**

Table 3.1: PASCAL VOC 2007 Detection Challenge.

	FMD	CAL	SCN	VOC
BoW	**.3021**	.2991	.1428	.1906
DSP-BoW	.3012	**.3731**	**.1664**	**.2147**

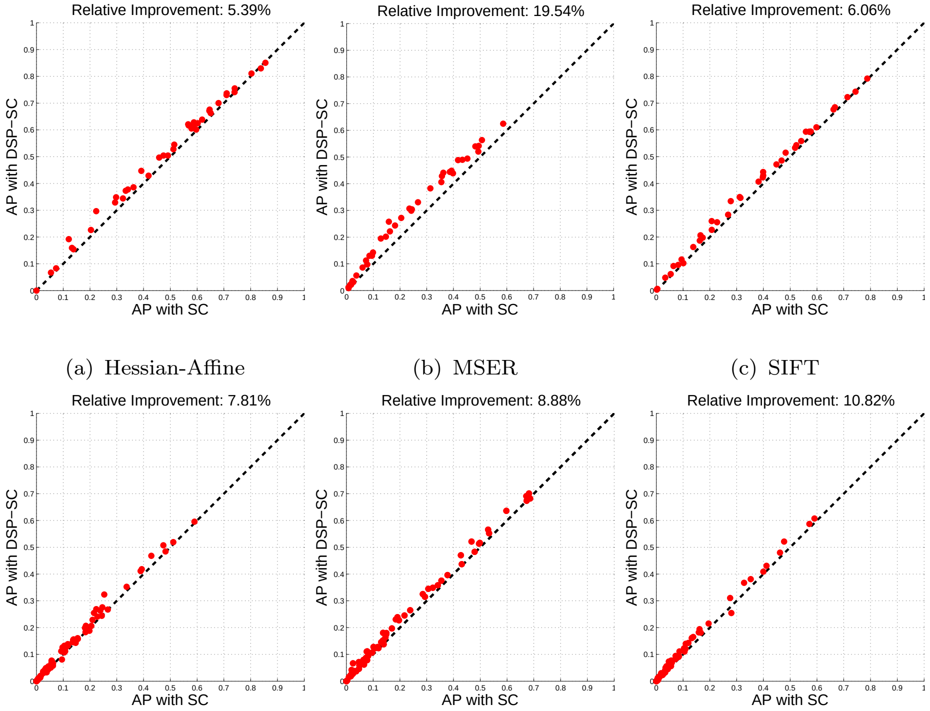
Table 3.2: Average Precision for Classification.

3.7.1 DSP-Scattering Transform

In this section, we compare domain-size pooled scattering network with the plain version of [BM11]. It has been established by the authors of [BM11] that the first level of the scattering transform is equivalent to a continuous extension of a histogram descriptor a' la SIFT. Therefore, pooling in the first level of the SC is equivalent to pooling gradient histograms in regions of the image domain. However, the second layer averages frequency components, so the extension of the interpretation is less straightforward. In our experiments, we limit the pooling to the first layer, and show that already it induces an increase in performance (Fig. 3.9) regardless of the detectors used to extract the image regions.

3.8 Conclusion

We have conducted an empirical exploration of the effects of pooling different domains in the construction of low-level descriptors, both histogram-based and binary, mid-level descriptors, and global architectures. We have found that the effects of pooling domain sizes, even when



(a) Hessian-Affine (b) MSER (c) SIFT

Figure 3.9: DSP-Scattering Transform (DSP-SC) vs. SC. Top (Left to Right): Oxford dataset with Hessian-Affine, MSER and SIFT detector. Bottom: Same for Balzer dataset. The results of Harris-Affine is similar to Hessian-Affine.

using simplistic pooling schemes such as uniform sampling with respect to uniform weights, are measurable improvements in performance. The degree of improvement depends on the particular method, and is largest for histogram-based descriptors, and smallest for binary descriptors. Pictorial structures that already densely sample domain sizes exhibit modest improvements that are probably not sufficient to justify their modification. Otherwise, for local descriptors, the algorithmic and computational changes are minimal and therefore warrant the adoption of domain-size pooling.

CHAPTER 4

Beyond Single-View Descriptors

4.1 Introduction

For visual data, a "feature descriptor" is a function of images designed to be "insensitive" to nuisance variability and yet "discriminative" with respect to intrinsic properties of the scene or object of interest. Nuisance variability may be due to changes of viewpoint and illumination, and intrinsic properties include three-dimensional shape and material properties of the scene, or object-specific deformations. The best-known local descriptors are SIFT [Low04], HOG [DT05] and their variants [BTG06], which we refer to collectively as HoG (histogram of gradient) in this chapter. For an image region centered at a point, they are histograms of the orientation of its gradient in that region, variously normalized.

On the other hand, representation learning via neural networks [LHB04] constructs functions that are insensitive to nuisance variability by training a convolutional architecture supported on the entire image domain. There have been several studies of the empirical performance of local feature descriptors, including their comparison [MTS04], and their generative abilities [VKM13, SVZ14b]. However, efforts to elucidate their relationships have only recently begun to appear [BM11, BRP09].

But what is an ideal representation? In terms of being "discriminative" of the intrinsic properties of the scene, such as its shape and reflectance, one could do no better than a (minimal) sufficient statistic, for instance the likelihood function [SC14]. In terms of being "insensitive" to nuisance factors, such as viewpoint and illumination, one could do no better than a (maximal) invariant to their action on the data. So, an ideal representation would be a *minimal sufficient* statistic that is *maximally invariant* to nuisance factors [SC14]. Does such a representation exist? If so, can it be computed? If not, can it be approximated? Can existing descriptors be related to it? If so, under what conditions? If not, how can we construct better approximations of an ideal representation? In this chapter, we aim to begin addressing these questions. Naturally, their answer depends on a *model* of image formation as well as *assumptions* on the underlying scene.

4.1.1 Related Work

There are many engineered descriptors of *one* image [Low04, DT05, BTG06, TLF10], that differ on where and how the local histograms are aggregated and normalized, with many implementation details affecting performance [CLV11]. Some entail learning [WB07, LLF05] to minimize classification (correspondence) error. Relatively few local descriptors aggregate multiple views: [DNO07] combines spatial (averaged SIFT) and temporal statistics; [GB05] performs feature selection from trajectories of key points. Deformable parts models [FMR08] are also learned from multiple views to capture intrinsic variability.

One could also learn away nuisance variability through a neural network architecture [LHB04, RHB07]. This approach has been steadily improving performance in large-scale pattern recognition [DDS09], but not in correspondence, where it is outperformed by engineered descriptors, even some built using a single image [DS15]. Rather than performing direct comparison between different descriptors, we instantiate an *ideal local representation* relative to a simple image-formation (Lambert-Ambient, or LA) model, and relate various descriptors to it.

4.1.2 Summary

To quantify how “discriminative” a descriptor is, we characterize its dependency on intrinsic properties of the scene, namely shape S and reflectance[1] ρ. To quantify how “insensitive” it is, we describe its dependency on nuisance factors such as viewpoint and illumination.

[1] In the LA model $S \subset \mathbb{R}^3$ is a multiply-connected piecewise smooth surface in Euclidean space, and $\rho : S \to \mathbb{R}^+$ is a positive-valued scalar function called “albedo.” As we model illumination via contrast transformations of the albedo, we interpret ρ modulo contrast changes as the *reflectance* of the surface S.

In [MSK03] the LA model is described as the simplest to capture the phenomenology of image formation for the purpose of correspondence. Local illumination changes are modeled, to first-order approximation, as monotonic continuous transformations of the range of the image, also known as *contrast transformations.* They form a group[2], and under certain conditions [SDK14] the gradient orientation is a maximal invariant. So we can eliminate first-order dependency on illumination by replacing the image[3] I with its gradient orientation $\theta(x) = \angle\nabla I(x) \doteq \nabla I(x)/\|\nabla I(x)\|$, at locations x where $\nabla I(x) \neq 0$. For a local neighborhood $\mathcal{B} \subset \mathbb{R}^2$, the *likelihood function,* computed at a location $x \in \mathcal{B}$ and conditioned on a given shape S and reflectance ρ, is a minimal sufficient statistic [SC14], and can be thought of as a probability density on θ, $p_{\mathcal{B}}(\theta|\rho, S)$ with marginals[4] $p_x(\theta|\rho, S)$. If there are additional groups G acting on the scene (for instance changes of spatial position and orientation, $G = SE(3)$) they can be marginalized, thus obtaining a density

$$p_{x,G}(\theta|\rho, S). \tag{4.1}$$

The marginalized likelihood is a maximal contrast-invariant that is also G-invariant. With respect to this *ideal representation*, our goals are to: (i) Instantiate the formal notation above using the LA model and derive an expression for (4.1) suitable for computation (Sec. 4.2.1). (ii) Show that HoG approximates an ideal descriptor when the scene is planar and the viewer is constrained to translating parallel to it (Sec. 4.2.1). (iii) Derive a sampling approximation of (4.1), which we call MV-HoG, where the scene (S, ρ) is replaced with a collection of images of it, captured from multiple viewpoints $\{I_t\}_{t=1}^T$ (Sec. 4.3.1). (iv) Derive a point-estimate based approximation of (4.1), which we call R-HoG, where the scene (S, ρ) is replaced with a point estimate $(\hat{S}, \hat{\rho})$ reconstructed from a finite sample $\{I_t\}_{t=1}^T$, possibly using structured illumination (Sec. 4.3.2).

[2] If strictly monotonic, lest they form a monoid.

[3] Here $I : D \subset \mathbb{R}^2 \to \mathbb{R}^+$; $x \mapsto I(x)$ is a gray-scale image, $x \in D$ is a point on the plane. In practice, I takes a finite number of values on a quantized domain, extended to the entire plane by zero-padding.

[4] If we knew the viewpoint, under the assumptions of the LA Model, the conditional density would be spatially independent, (4.10); otherwise, marginalizing viewpoint introduces spatial dependency, so the product of the marginals is only an approximation, (4.12).

4.2 Engineered Features Revisited

A "cell" of the HOG/SIFT descriptor[5] h of an image I in a region centered at a pixel x is a histogram of the orientation of its gradient, θ, around x. If the histogram is not normalized, we call it uHoG (un-normalized HoG) and indicate it with

$$h_x(\theta|I) \qquad\qquad \texttt{uHoG}. \tag{4.2}$$

Given one image I, this un-normalized histogram returns a positive number for each orientation θ, related to the number of pixels around x where the image gradient orientation is close to θ. Variants of HoG differ in where they compute and how they aggregate and normalize such histograms. For instance, SIFT [DT05] evaluates the histogram above on a 4×4 grid $\mathcal{B} = \{x_i,\ i = 1, \dots, 16\}$, and concatenates the result into a vector $[h_{x_1}, \dots, h_{x_{16}}]$, that is then normalized, clamped, and re-normalized. Discrete bins are computed using a bilinear interpolation kernel κ_ϵ with $\epsilon = 2\pi/\#\ \texttt{bins}$, and a linear spatial weighting kernel κ_σ with σ the area of each cell in the 4×4 grid, further weighted by the magnitude of the image gradient $\|\nabla I\|$. If we extend the sum to the continuum, we can write the histogram in each cell as [VF10, DKD15]

$$h_x(\theta|I) = \int \kappa_\epsilon\big(\theta - \angle\nabla I(y)\big)\kappa_\sigma(x-y)\|\nabla I(y)\|dy \tag{4.3}$$

where the argument of the orientation kernel is intended modulo 2π. Alternatively, histograms can be normalized independently at each location x:

$$\bar{h}_x(\theta|I) = \frac{h_x(\theta|I)}{\int_{\mathbb{S}^1} h_x(\theta|I)d\theta}, \qquad h = [h_{x_1}, h_{x_2}, \dots, h_{x_i}, \dots]. \tag{4.4}$$

Note that in HoG, described above, the nuisance group G is absent. We introduce it next.

4.2.1 Ideal descriptor of one view and its HoG

As a preliminary step to computing the minimal sufficient invariant statistic (4.1), and to understand its relation to single-view descriptors, consider a special case obtained by

[5] Here $\theta \in \mathbb{S}^1$ is an angle (the free variable) and $h : D \times \mathbb{S}^1 \to \mathbb{R}^+; (x, \theta) \mapsto h_x(\theta)$ for a fixed image I.

assuming that the scene is a plane parallel to the image plane, with albedo equal to the image irradiance. Then, conditioning on the image I, we have $p_{x,G}(\theta|I)$, which we wish to relate to uHoG (4.2).

To guarantee contrast-invariance, one could replace the intensity $I(x) \in \mathbb{R}^+$ with the curvature of the iso-contours [AGL93], or with its dual, the orientation of the gradient, $\angle \nabla I(x) \in \mathbb{S}^1$ where $\nabla I(x) \neq 0$. Let (G, P) be a probability space, with G a group and P a probability distribution on the group, and suppose that to each $g \in G$ we can associate a "transformed" image I_g. For each pixel $x \in \mathbb{R}^2$ where $\nabla I_g(x) \neq 0$, we can then define a (marginal) probability density function over θ, for instance:

$$p_{x,G}(\theta|I, g) \doteq \mathcal{N}_\varepsilon\left(\theta - \angle \nabla I_g(x)\right) \tag{4.5}$$

where the difference is intended in $\mathbb{S}^1$, and correspondingly $\mathcal{N}_\epsilon$ denotes an angular Gaussian [Wat83]. Kernels κ other than Gaussian can also be considered without significant changes to the arguments that follow. Using P, we can marginalize[6] this distribution to eliminate its dependency on $g \in G$:

$$p_{x,G}(\theta|I) \doteq \int_G p_{x,G}(\theta|I, g) dP(g). \tag{4.6}$$

To understand the relationship with uHoG, we restrict G to be the group of planar translations, $G = \mathbb{R}^2$, and choose a particular measure for $\mathbb{R}^2$, $d\mu(v|I) \doteq \|\nabla I_v(x)\| dv$ where, if $v \in G$, $I_v(x) = I(x + v)$ is the transformed image. We then marginalize with respect to the (un-normalized) distribution $dP(v) = \mathcal{N}_\sigma(v) d\mu(v|I_v)$. This corresponds to assuming that the scene is *flat*, parallel to the image-plane (fronto-parallel) and constrained to translate parallel to it. The likelihood function is given by $p_{x,G}(\theta|I, v) = \mathcal{N}_\varepsilon(\theta - \angle \nabla I_v(x))$. Integrating

[6] The integral is well defined by Fubini's theorem; $p_{x,G}(\theta|I, g)$ is a measurable function of g and bounded so the marginalization converges. Thus, we can integrate over θ and exchange the integrals. But while marginalization guarantees invariance to $g \in G$, it does not yield a maximal invariant, which is instead described in [SC14].

against $dP(v)$, we obtain

$$\begin{aligned} h_x(\theta|I) &= \int_G p_{x,G}(\theta|I,v)dP(v) = \\ &= \int_{\mathbb{R}^2} \mathcal{N}_\varepsilon(\theta - \angle\nabla I_v(x))\mathcal{N}_\sigma(v)d\mu(v|I_v) \\ &= \int_{\mathbb{R}^2} \mathcal{N}_\varepsilon(\theta - \angle\nabla I(y))\mathcal{N}_\sigma(y-x)\|\nabla I(y)\|dy, \end{aligned} \tag{4.7}$$

which is one cell of uHoG (4.3) once we restrict to the discrete lattice and replace the Gaussian kernels with (bi-)linear ones. The full descriptor is just the concatenation of a number of cells, suitably normalized; for the case of a single cell,

$$p_{x,G}(\theta|I) = \frac{h_{x,G}(\theta|I)}{\int h_{x,G}(\theta|I)d\theta} \tag{4.8}$$

which leads us to conclude that HOG/SIFT approximates the ideal representation at a point under the assumption that the scene is flat and fronto-parallel, undergoing purely translational motion parallel to the image plane.

4.3 Ideal Descriptor Approximations

To move one step closer to the ideal representation, and to relax the stringent assumptions implicit in HOG/SIFT, suppose for now that we have complete knowledge of the underlying scene (S, ρ). A pinhole camera projects each point on the scene to the image plane via[7] $\pi : S \to D \subset \mathbb{R}^2$ and its associated inverse $\pi_S^{-1} : D \to S$, where $\pi_S^{-1}(x)$ is the point of the first intersection of the pre-image (a line) of x with the scene S. Under the assumptions of the LA model, there exists an open subset $G_0 \subseteq SE(3)$ with compact closure and – after a suitable change of reference frame – containing the identity, such that each $g \in G_0$, with the action

$$I_g(x) = \rho \circ g \circ \pi_S^{-1}(x) \tag{4.9}$$

can be associated with a domain diffeomorphism $w_g : \mathbb{R}^2 \to \mathbb{R}^2$, with $I_g(x) = I(w_g(x))$. Here "$\circ$" denotes function composition. When emphasizing the dependency of w_g on shape,

[7] π incorporates the projection by dividing the coordinates of a point in S by the third component and applying a planar affine transformation depending on the intrinsic calibration of the camera [MSK03].

we indicate it with $w_g(x|S)$. Let P be a probability measure on G_0, *e.g.,* the normalized restriction of the Haar measure on $SE(3)$ to G_0, which is no longer a group, but a subset of G, where the probability of actions outside G_0 is assigned to zero. Then the marginalized descriptor, for a *known* scene, is given by

$$\begin{aligned} p_{x,G_0}(\theta|\rho, S) &= \int_{G_0} \mathcal{N}_\varepsilon(\theta - \angle\nabla\rho \circ g \circ \pi_S^{-1}(x))dP_{G_0}(g) \\ &= \int_{G_0} \mathcal{N}_\varepsilon(\theta - \angle\nabla I(w_g(x|S)))dP_{G_0}(g). \end{aligned} \tag{4.10}$$

The first approximation step is to reduce the dimensionality of $G_0 \subset SE(3) = SO(3) \times \mathbb{R}^3$ to simplify marginalization. This can be done locally around a point $\pi_S^{-1}(x)$ through the use of a *co-variant detector,* a function of the image that returns multiple isolated elements of subsets of G_0 that co-vary with g. For instance, a translation-scale detector [Low04] returns isolated locations on the image plane, x_i, and their corresponding scales σ_i, which can be used to define a local reference frame centered at x_i with unit σ_i. To first approximation, as we qualify in the next paragraph, these co-vary with the translation component of G_0: A spatial translation parallel to the image plane induces a planar translation of x_i, and a spatial translation orthogonal to the image plane induces a change of scale σ_i. Thus, locally around $\pi_S^{-1}(x_i)$, we can annihilate the effects of spatial translation simply by *canonizing* the location-scale group, *i.e.,* imposing $x_i = 0, \sigma_i = 1$, by applying the inverse transformation of that determined by the co-variant detector. This procedure can be applied to any planar group transformation, including the entire group of diffeomorphisms [SPV09]. In particular, planar rotation can be canonized using the direction of gravity as a reference [JS11], leaving only "out-of-plane" rotations to be marginalized in (4.10).

In reality, spatial translations do not co-vary with planar translation-scale transformations, for the former induces (shape-dependent) deformations of the image domain (4.9) in addition to non-invertible transformations due to *occlusions*, which are absent in the latter. Such shape-dependent image variability is lost in any descriptor computed from a single image: Any finite-dimensional planar group-covariant detector co-varies with spatial translations only when the scene is flat and the neighborhood of size σ_i centered in x_i, $\mathcal{B}_{\sigma_i}(x_i)$, does not straddle occluding boundaries. Fortunately, we are not constrained to building descriptors using a single image; instead, we can capture residual deformations after canonization by marginalizing with respect to out-of-plane rotations in $SO(3)$. In addition, we can also marginalize small residual changes in translation v and scale σ using some prior $P_{\mathcal{N}_\sigma} \times P_{\mathcal{E}_s}$, where[8] $dP_{\mathcal{N}_\sigma}(v) = \mathcal{N}_\sigma(v)d\mu(v)$ and $dP_{\mathcal{N}_s}(\sigma) = \mathcal{E}_s(\sigma)d\sigma$ with $\mathcal{E}$ a unilateral density (*e.g.,* exponential) to ensure $\sigma > 0$. Thus, our un-normalized conditional distribution becomes:

$$h_{x,G}(\theta|\rho, S) = \int_{G_0} \mathcal{N}_\varepsilon(\theta - \angle\nabla I_g(x))dP_{G_0}(g) \simeq \tag{4.11}$$
$$\int \mathcal{N}_\varepsilon(\theta - \angle\nabla I(w_g(y)))dP_{SO(3)}(g)\mathcal{N}_\sigma(y - x))\mathcal{E}_s(\sigma)d\mu(y)d\sigma.$$

If out-of-plane rotations are neglected, or if the scene is planar, one image is sufficient to construct an idea descriptor, which then reduces to DSP-SIFT, introduced in Chapter 2 and [DS15]. To obtain the ideal descriptor of *a region* $\mathcal{B}$, we must consider the joint distribution of all pixels within: $h_{x_1,\dots,x_k,G}(\theta_1, \dots, \theta_k|\rho, S)$. Aggregating histograms in high dimensions is challenging but the joint distribution can be approximated by a collection of one-dimensional marginals. The simplest approximation is to neglect spatial correlations altogether: From (4.10),

$$\begin{aligned} & p_{x_1,\dots,x_k,G_0}(\theta_1, \dots, \theta_k|\rho, S) = \\ = \;& \int_{G_0} \prod_{i=1}^{k} \mathcal{N}_\varepsilon(\theta_i - \angle\nabla I(w_g(x_i|S)))dP_{G_0}(g) \\ \simeq \;& \prod_{i=1}^{k} h_{x_i,G}(\theta_i|\rho, S). \end{aligned} \tag{4.12}$$

As already pointed out[4], under the assumptions of the LA model, if the vantage point $g \in SE(3)$ was known, then the conditional density above would indeed factorize into the product of marginals computed independently at each pixel. However, marginalizing viewpoint introduces spatial dependencies, so the above is just an approximation.[9] Even this

[8] It should be noted that this approximation step does not reduce the generality of the approach: In practice, one would have to discretize the group G_0 anyway in order to perform the marginalization in (4.10), and co-variant detectors are just an adaptive discretization mechanism. A trivial detector is one that returns regular samples of the group, for instance a discretization of planar translations and scales as customary in "dense SIFT." Indeed, this discretization is necessary also to compactify the translational component of G_0, that otherwise would have to be marginalized with respect to an improper measure.

[9] Coarse as it seems, this is nevertheless the approximation implicit in most single-view descriptors, that

approximation, however, requires knowledge of the scene (S, ρ) to be computed. We now address how to cope with absence of such knowledge.

4.3.1 Sampling approximation: MV-HoG

If we do not have complete knowledge of the scene, (S, ρ), but we have a collection of images of it $\{I_t\}_{t=1}^T$, we can approximate (4.11) by Monte-Carlo sampling, after noticing that $I_t(x) = \rho \circ g_t \circ \pi_S^{-1}(x) = I(w_{g_t}(x))$ with $\{w_{g_t} | t = 1, \cdots, T\}$ and $g_t \sim P_{G_0}$ with the restriction G_0 determined by visibility. Under sufficient excitation conditions on the sample $\{I_t\}_{t=1}^T$, asymptotically for $T \to \infty$, we can approximate the integral with

$$h_{x,G}(\theta | \{I_t\}_{t=1}^T) \doteq \frac{1}{T} \sum_{t=1}^{T} \int_{\mathbb{R}^2} \mathcal{N}_\varepsilon(\theta - \angle \nabla I_t(y)) \mathcal{N}_\sigma(y - x) d\mu(y).$$

Scale σ can also be marginalized as in (4.11). Sufficient excitation conditions mean that the orbit in $SE(3)$ is sampled along all directions (in the Lie Algebra), which is a tall order, as it requires every surface element to be seen from all vantage points, at all distances, while g_t remains in G_0. This requirement can be mitigated by restricting the marginalization to $SO(3)$ or even to just out-of-plane rotations, using (4.11) in conjunction with a co-variant detector or other sampling mechanism.

Alternatively, we can use whatever data is available to reconstruct a model (a point estimate) of the scene, which can then be used to render synthetic samples from the orbits of $SE(3)$.

4.3.2 Point-estimate approximation: R-HoG

Samples $\{I_t\}$ can be used to compute an approximation of ρ, S, for instance in the sense of maximum-likelihood, with suitable regularization [FK96, GBS15]

$$\hat{\rho}, \hat{S} \;=\; \arg\max_{\rho, S, g_t} p(\{I_t\} | \rho, S) \,+\, \lambda R(S) \qquad \texttt{subject to} \quad I_t \;=\; \rho \circ g_t \circ \pi_S^{-1} \,+\, n_t \quad (4.13)$$

consider the concatenation of (independently aggregated, scalar) histograms. Some single-view descriptors attempt to recapture some of the lost spatial correlations by joint (re)-normalization [DT05].

where $R(S)$ is, for instance, surface area $\int_S dA$, n_t is white and Gaussian, and λ is a scalar multiplier, and then compute (4.10) restricted to out-of-plane rotations:

$$h_{x,G}(\theta|\hat{\rho}, \hat{S}) = \int_{SO(3)} \mathcal{N}_\varepsilon(\theta - \angle\nabla\hat{\rho} \circ g \circ \pi_{\hat{S}}^{-1}(x))dP_{SO(3)}(g) \tag{4.14}$$

or its spatially regularized version:

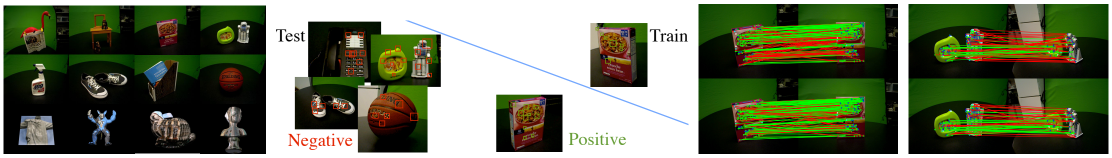
$$h_{x,G}(\theta|\hat{\rho}, \hat{S}) = \int_{SO(3)\times\mathbb{R}^2} \mathcal{N}_\varepsilon(\theta - \angle\nabla\hat{\rho} \circ g \circ \pi_{\hat{S}}^{-1}(y))dP_{SO(3)}(g)\mathcal{N}_\sigma(y - x)d\mu(y)$$

or its scale-marginalized version as in (4.11). Convergence and unbiasedness of the maximum-likelihood estimator ensures convergence of R-HoG to (4.11). Note that it is possible for the reconstruction to be significantly different from S and yet R-HoG be similar to the ideal descriptor, so long as the re-projections $\hat{\rho}\circ g\circ\pi_{\hat{S}}^{-1}(x)$ are compatible with $w_{g_t}(x|S)$. This can happen, for instance, when $\hat{S}$ differs from S in regions where ρ is constant. Also note that, in theory, two views with non-trivial baseline are sufficient to reconstruct an approximation of $\hat{S}$ and $\hat{\rho}$, locally in the co-visible region. Therefore, R-HoG is preferable when T is small and the sample I_t is unlikely to be sufficiently exciting. Normalized versions of each descriptor are obtained as

$$p(\theta|X) = \frac{h_{x,G}(\theta|X)}{\int h_{x,G}(\theta|X)d\theta}, \tag{4.15}$$

where $X = I$ for HOG, $X = \{I_t\}$ for MV-HoG, $X = \{\hat{\rho}, \hat{S}\}$ for R-HoG, and $X = \{\rho, S\}$ for the ideal descriptor that marginalizes the nuisance assuming a known scene.

While MV-HoG had a stringent sampling requirement, R-HoG has its own challenges, in that obtaining a reliable, dense reconstruction of a scene and its photometry can be a tall order. However, an estimate of the surface is only needed locally, where smooth surfaces can be approximated with parametric models of low order. Also, calibrated reconstruction is not necessary, so a projective reconstruction can be obtained through solving systems of linear equations [MSK03]. Alternatively, a structured model can be inferred through factorization methods such as principal component analysis or sparse coding, whereby S is represented by the coefficients of a linear combination of a collection of "basis elements" $\{S_i\}$.



(a) Sample objects (b) Test samples (c) Cereal (d) Robot

Figure 4.1: Dataset, Test Samples and Qualitative Match Visualization. (a): Samples from the real and synthetic object dataset. (b): Positive test samples from the object; negative samples are ten-fold more numerous. (c), (d) show correct (green) and wrong (red) matches claimed by SV-SIFT (Top) and MV-HoG (Bottom). The latter yields many more correct matches, similar to R-HoG.

4.4 Dataset and Ground Truth

Since our focus here is to leverage on *multiple views* to build better descriptors, which can then be matched to single-images in wide-baseline tests, to perform comparisons we need a dataset where *multiple* training images (of the same scene) are available, whereas correspondence testing can be performed on single images.

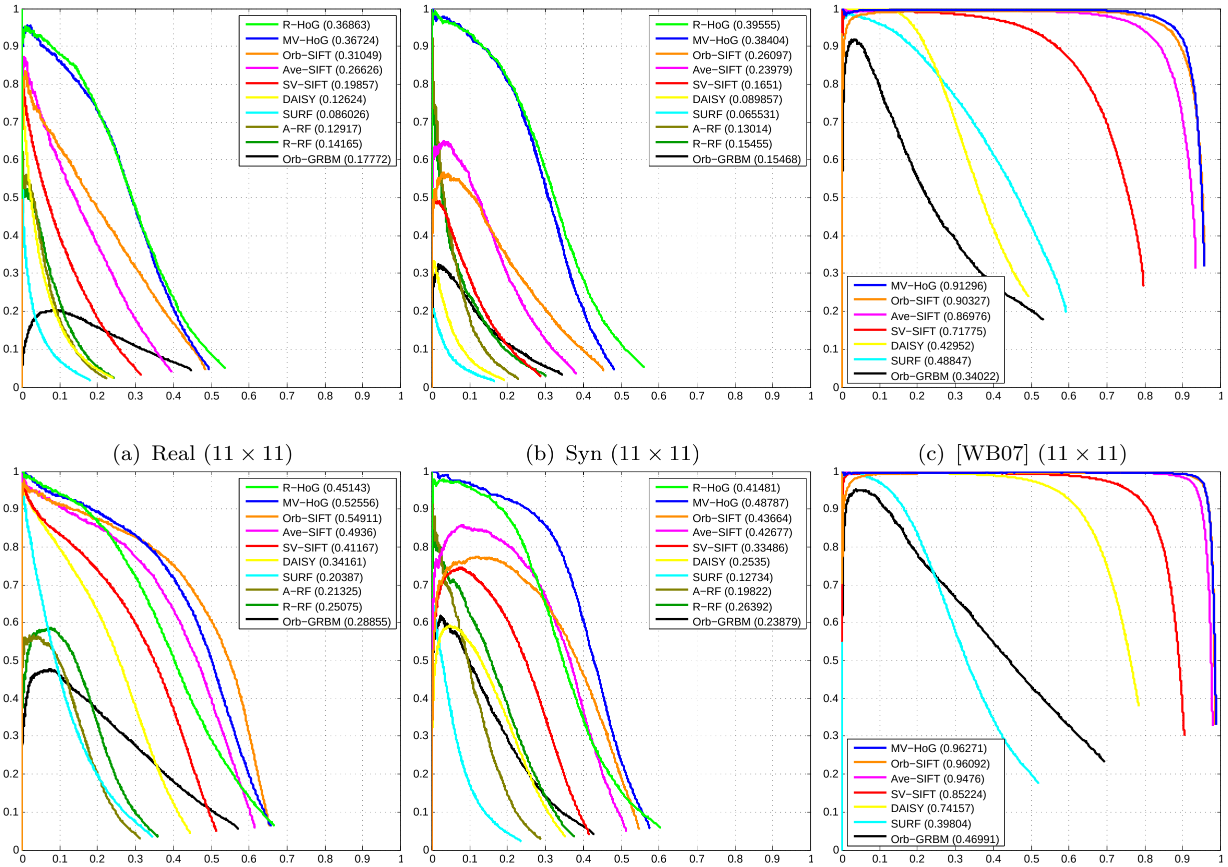
Many datasets are available to test image-to-image matching, *e.g.,* [MTS04], where both training and test sets are individual images, each of a different scene. Testing our approach on such datasets would require forgoing marginalization of out-of-plane rotation, thus reducing our approach to DSP-SIFT, which has been tested on [MTS04] by [DS15].

Fewer datasets are available for testing multi-view descriptors [MP07, WB07]. The latter contains three scenes: Trevi, Half Dome and Notre Dame and provides pixel-level correspondence by back-projecting 3D reconstructed keypoints onto images, which can be used for evaluation. To enable the comparison, we extract a subset containing only features having more than 10 samples. We randomly hold out 5 samples for testing and use the rest for descriptor aggregation. Negative samples are randomly selected from the other scenes.

Almost perfect results are obtained on [WB07] (Fig. 4.2), thus limiting the value of the dataset; we have therefore constructed a new dataset, similar in spirit to [MP07], but with a *separate* test set and dense ground truth for validation, using a combination of 31 real and 15 synthetic objects. The latter are generated by texture-mapping random images onto surface models available in MeshLab. The former are household objects of the kind seen in Fig. 4.1. Some with significant texture variability, others with little; some with complex shape and topology, others simple. In each case, a sequence of (training) images per object is obtained by moving around the objects in a closed trajectory. For real objects, a 400-frame-trajectory circumnavigates them to reveal most visible surfaces; for synthetic ones, 100 frames span a smaller orbit.

Ground Truth: We compare descriptors built from the (training) video and test single frames, by first selecting test images where a sufficient co-visible area is present. To establish ground truth, we reconstruct a dense model of each (real) object using an RGB-D (structured light) range sensor with YAS [BPS14]. The reconstructed surface enables dense correspondence between co-visible regions in different images by back-projection. This is further validated with standard tools from multiple-view geometry by epipolar RANSAC. Occlusions are determined using the range map. Further implementation details are described in [DKD15].

Detection and Tracking: We use FAST [RD06] as a mechanism to (conservatively) eliminate regions that are expected to have non-discriminative descriptors, but this step could be forgone. Scale changes are handled in a discrete scale-space, *i.e.,* images are downsampled by half up to 4 times and FAST is computed at each level. Short-baseline correspondence is established with standard MLK [LK81]. A sequence of image locations is returned by the tracker for each region, which is then sampled in a rectangular neighborhood at the scale of the detector. We report experiments on two window sizes, 11×11 and 21×21, illustrative of a range of experiments conducted. The sequence of such windows is then used to compute the descriptors.



(a) Real (11 × 11) (b) Syn (11 × 11) (c) [WB07] (11 × 11)

(d) Real (21 × 21) (e) Syn (21 × 21) (f) [WB07] (21 × 21)

Figure 4.2: Precision-Recall Curves. Precisions (ordinate) over recall rates (abscissa) with F1-scores in the legends.

4.5 Evaluation and Comparison

We briefly describe the descriptors and classifiers involved in the evaluation and refer to [DKD15] for the implementation details, parameter selections and training procedures.

Single-View Descriptors: We use SIFT from [VF10] as baseline (SV-SIFT), computed on each patch at each frame as determined by the detector and tracker. We also compare single-view descriptor representatives DAISY [TLF10] and SURF-128 [BTG06] computed on the individual images.

Multiple-View Descriptors: MV-HoG is implemented according to Sect. 4.3.1 using the tracks returned by the MLK tracker. We also tested Random Forest [LLF05] as an alternative way of utilizing multiple samples. We present to the RFs the training samples, and refer to this as A-RF. Deformable parts models would be too slow to test on our dataset, so we forgo that comparison.

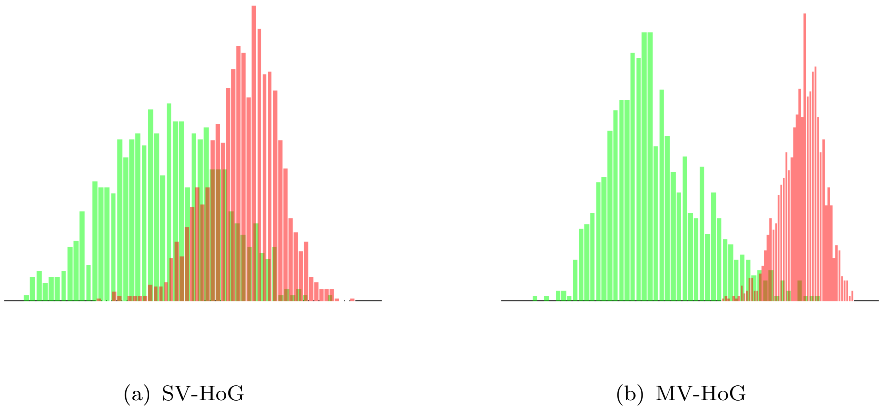
Reconstructive Descriptors: To compute an approximation of R-HoG in Sect. 4.3.2, we compute dense 3-D reconstructions both from some tracked sequences and using a structured-light sensor. Where visual reconstruction was successful, performance was similar, but dense reconstruction was laborious and the quality was not consistent across samples, so to make the evaluation independent of reconstruction methods, we report the results using a structured light sensor only. We use the keyframe where features are first extracted, and sample a viewing hemisphere with 576 vantage points. The R-HoG is built upon these synthesized samples. As in the multiple view case, we also feed synthesized patches to the Random Forest (R-RF).

Classifier and Strategies: Given a descriptor database, the simplest method to match a test query is via *nearest neighbor* (NN) search. We compare five combinations using the same NN search method: (i) single view SV-SIFT, SURF and DAISY – computed on a random image from the training sequence, (ii) Ave-SIFT [DNO07] – averaged SIFT of all frames, (iii) Orb-SIFT – all of the SV-SIFTs stored to represent the orbit which includes the best possible exemplar for each feature [GB05], (iv) MV-HoG and (v) R-HoG.

Network Architecture: We also compare our methods with a simple network architecture in the form of a gated restricted Boltzmann machine (G-RBM) [Mem13, TFL10, SMH11], employed by the authors in correspondence tasks similar to those considered in this chapter. We use the same matching strategy as Orb-SIFT, so we call the network Orb-GRBM. Details of the G-RBMs are in [DKD15].

4.5.1 Metrics

We use precision-recall curves (PR-curves) to quantitatively evaluate the descriptors proposed and compare them to existing methods. For each query patch, nearest neighbor search returns a predicted label and its associated distance. By changing a distance thresh τ_d, a precision-recall curve can be generated. Precision and recall are defined as $p = \frac{\#\text{true matches}}{\#\text{false matches}+\#\text{true matches}}$, $r = \frac{\#\text{true matches}}{\#\text{positive samples}}$. The *positive samples* are the test queries that


(a) SV-HoG

(b) MV-HoG

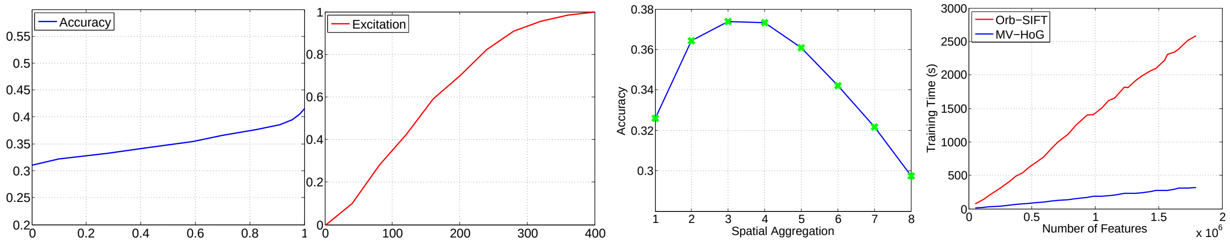
Figure 4.3: Distance Distribution. The horizontal axis indicates the distance between two descriptors in increasing order from left to right. The distribution of distances between corresponding features are shown in green and that of mismatches in red. The error (overlapping area) in 4.3(b) is considerably smaller than 4.3(a). This leads to a lower risk of misclassification in MV-HoG.

have correspondences in the training databases as opposed to the *negative* samples which are never seen in training. The *matches* are the queries that pass the distance threshold test. A match is considered to be a *true match* if the predicted label is correct according to the ground truth. As only one predicted label is obtained for each query, r could remain < 1 once any predicted label is wrong. We report the F1-score $\left(\frac{2pr}{p+r}\right)$ for each PR curve. Similarly, random forests (A-RF and R-RF) return an averaged probability as a confidence score for the predicted label. A precision-recall curve can be generated by changing a belief threshold τ_p.

4.5.2 Empirical Results

Qualitative results are shown in Fig. 4.1. In Fig. 4.2, PR curves are shown for all the datasets on two different patch sizes. R-HoG and MV-HoG are comparable on 11×11 patches and outperform other methods. On 21×21 patches, the 3D-reconstruction generates artifacts in the view-set generation, so the performance of R-HoG decreases below that of MV-HoG in both the real and synthetic datasets. It should not be surprising that Orb-SIFT performs the best among all the other methods, as it entails exhaustive search over the orbit of transformed views. However, its precision drops sharply when the number of negatives is large, as it inherits the vulnerability of SV-SIFT to outliers. Also, MV-HoG is consistently better than Ave-SIFT across all datasets. Note that both involve averaging histograms, but Ave-SIFT averages *normalized* descriptors computed in each frame, and then re-normalized, whereas MV-HoG aggregates gradient orientation over time, and only normalizes the descriptor at the end, using the same procedure and clamping threshold as Ave-SIFT. This shows that temporal aggregation improves performance compared to simply averaging single-view descriptors computed independently.

Fig. 4.3 shows the distance distributions between descriptors of corresponding and non-corresponding patches. SV-HoG is computed from a random single sample from each track, and MV-HoG is aggregated over the whole track. The overlapping area between the two distributions indicates the probability of making a classification error in descriptor matching. The distributions in Fig. 4.3(b) have much less overlapping area than that in Fig. 4.3(a). It shows that the discriminative power of the descriptor is improved by aggregating over multiple views.



(a) Sufficient excitation

(b) Spatial σ

(c) Time complexity

Figure 4.4: (a) Sufficient excitation. Left: Accuracy (maximum recall) as a function of a proxy of sufficient excitation (see text). Right: Excitation as a function of the number of frames. All results are averaged over multiple runs using frames $i, \ldots, i+k-1$ where i is selected at random. (b) F1-score varies with spatial aggregation parameter σ. (c) Time complexity as a function of the number of features with FLANN precision at 0.7. Higher precision will further increase computational load.

4.5.3 Support Region, Spatial Aggregation, Sample Sufficiency and Complexity

The size of the domain where descriptors are computed impacts performance (Fig. 4.2): the larger, the better, so long as the domain remains co-visible (*i.e.,* $g_t \in G_0$). Fig. 4.4(b) shows the effect of the spatial parameter σ in MV-HoG (Sect. 4.3.1). A slight spatial aggregation enhances robustness until σ reaches a critical value, beyond which discriminative power drops. Multiple view descriptors perform scene-dependent blurring, and therefore remain more discriminative, as long as sufficient excitation conditions are met. Clearly, if a sequence of identical patches is given (video with no motion), the descriptor will fail to capture the representative variability of images generated by the underlying scene. In this case, MV-HoG reduces to DSP-SIFT [DS15], which differs from SV-SIFT because of domain-size aggregation (averaging over σ). In Fig. 4.4(a) we explore the relation between performance gain and excitation level of the training sequence. As a proxy of the latter, we measure the variance of the intensity relative to the mean using the ℓ_2 distance. The right plot shows that the variance reaches the maximum when most frames are seen. We normalize the variance so that 1 means maximum excitation. The left plot shows accuracy increases with excitation. The fact that accuracy does not saturate is due to the fact that the sufficient excitation is only reachable asymptotically. At test time, all descriptors of n features have the same storage complexity $O(n)$ except that Orb-SIFT stores every instance ($O(kn)$). The search can be done in approximate form using *approximate nearest neighbors* [DBS13]. Fig. 4.4(c) shows the training time using the *fast library for approximate nearest neighbors* (FLANN) vs MV-HoG on a commodity PC with 8GB memory and Xeon E3-1200 processor. MV-HoG scales well and is more memory-efficient while Orb-SIFT requires more training time and occupies more than 60% of the available memory. Another advantage of MV-HoG is that the descriptor can be updated incrementally, and does not require storing processed samples.

4.6 Discussion

By interpreting the SIFT/HOG family as the probability density of sample images conditioned on the underlying scene, with nuisances marginalized, and observing that a single image does not afford proper marginalization, we have been able to extend it using nuisance distributions learned from multiple training samples of the same underlying scene. The result is a multi-view extension of HoG that has the same memory and run-time complexity as its single-view counterpart, but better trades off sensitivity with discriminative power, as shown empirically, even with the classifier trivialized.

Our method has several limitations: It is restricted to static (or slowly-deforming) objects; it requires correspondence in multiple views to be assembled (although it reduces to DSP-SIFT if only one image is available), and is therefore sensitive to the performance of the tracking (MV-HoG) or reconstruction (R-HoG) algorithm. The former also requires sufficient excitation conditions to be satisfied, and the latter requires sufficiently informative data for multi-view stereo to operate, although if this is not the case (for instance in textureless scenes), then by definition the resulting descriptor is insensitive to nuisance factors; it is also, of course, uninformative, as it describes a constant image, and therefore this case is of no interest. It also requires the camera to be calibrated, but for the same reason, this is irrelevant as what matters is not that the reconstruction be correct in the Euclidean sense, but that it yields consistent reprojections.

Our empirical evaluation of R-HoG yields a performance upper bound, as we use a better approximation of the reconstruction (from a structured light sensor or ground truth) rather than multi-view stereo that, while possible, yielded inconsistent results across different samples. As the quality (and speed) of the latter improve, the difference between the two will shrink. We have also neglected the effects of sampling artifacts in the approximation of the ideal descriptor. However, in practice we have found them to be of second-order, compared to the approximation implicit in the spatial independence of the locally-aggregated histograms. Also, we wish to point out that ideal representations, in the sense of sufficient statistics that are (maximally) invariant, are not unique. However, they are equivalent from the informational standpoint [SC14]. Analytical evaluation of our approach is forthcoming [SDK14].

CHAPTER 5

Nuisance Management in Convolutional Architectures

5.1 Introduction

Convolutional neural networks (CNNs) are the de-facto paragon for detecting the presence of objects in a scene, as portrayed by an image. CNNs are described as being “approximately invariant” to nuisance transformations such as planar translation, both by virtue of their architecture (the same operation is repeated at every location akin to a “sliding window” and is followed by local pooling) and by virtue of their approximation properties that, given sufficient parameters and transformed training data, could in principle yield discriminants that are insensitive to nuisance transformations of the data represented in the training set. In addition to planar translation, an object detector must manage variability due to scaling (possibly anisotropic along the coordinate axes, yielding different aspect ratios) and (partial) occlusion. Some nuisances are elements of a transformation group, *e.g.,* the (anisotropic) location-scale group for the case of position, scale and aspect ratio of the object’s support.[1] The fact that convolutional architectures appear effective in classifying images as containing a given object regardless of its position, scale, and aspect ratio [KSH12, SZ15] suggests that the network can effectively manage such nuisance variability.

However, the quest for top performance in benchmark datasets has led researchers away from letting the CNN manage all nuisance variability. Instead, the image is first pre-processed to yield *proposals*, which are subsets of the image domain (bounding boxes) to be tested for the presence of a given class (Regions-with-CNN [GDD14]). Proposal mechanisms aim to remove nuisance variability due to position, scale and aspect ratio, leaving a “Category

[1] The region of the image the objects projects onto, often approximated by a bounding box.

CNN" to classify the resulting bounding box as one of a number of classes it is trained with. Put differently, rather than computing the *posterior* distribution[2] with nuisance transformations automatically marginalized, the CNN is used to compute the *conditional* distribution of classes given the data *and* a sample element that approximates the nuisance transformation, represented by a bounding box. If the goal is the nuisance itself (object support, as in *detection* [DDS09]) it can be found via maximum-likelihood (*max-out*) by selecting the bounding box that yields the highest probability of any class [GDD14, HZR14]. If the goal is the class regardless of the transformation (as in *categorization* [DDS09]), the nuisance can be approximately *marginalized out* by averaging the conditional distributions with respect to an estimation of the nuisance transformations[2].

Now, if a CNN was an effective way of computing the marginals with respect to nuisance variability, there would be no benefit in conditioning and averaging with respect to (inferred) nuisance samples. This is a direct corollary of the Data Processing Inequality (DPI, Theorem 2.8.1 in [CT12]). Proposals are subsets of the whole image, so in theory less informative even after accounting for resolution/sampling artifacts (Fig. 5.1). *A fortiori*, performance should further decrease if the conditioning mechanism is not very representative of the nuisance distribution, as is the case for most proposal schemes that produce bounding boxes based on adaptively downsampling a coarse discretization of the location-scale group [HBD15]. Class posteriors conditioned on such bounding boxes discard the image outside it, further limiting the ability of the network to leverage on side information, or "context". Should the converse be true, *i.e.,* should averaging conditional distributions restricted to proposal regions outperform a CNN operating on the entire image, that would bring into question the ability of a CNN to marginalize nuisances such as translation and scaling or else go against

[2] One can think of the conditional distribution of a class c given an image x, $p(c|x)$, as defined by a CNN, as the class posterior $\int_G p(c|x,g)dP(g|x)$ marginalized with respect to the nuisance group G. If the nuisances are known, one can use the class-conditionals $p(c|x,g_r)$ at each nuisance $g_r \in G$ in order to approximate $p(c|x)$ with a weighted average of conditionals, *i.e.,* $p(c|x) \simeq \sum_r p(c|x,g_r)p(g_r|x)$.

When a CNN is tested on a proposal $r \subseteq x$ determined by a reference frame x_r, it computes $p(c|x_{|_r})$ (x restricted to r), which is an approximation of $p(c|x,g_r)$. Then, explicit marginalization (assuming uniform weights) computes $\frac{1}{|r|}\sum_r p(c|x_{|_r})$ which is different from $\frac{1}{|r|}\sum_r p(c|x,g_r)$ which in turn is different from $\sum_r p(c|x,g_r)p(g_r|x)$. This approach is therefore, on average, a lower bound on proper marginalization, and the fact that it would outperform the direct computation of $p(c|x)$ is worth investigating empirically.

the DPI. In this chapter we test this hypothesis, aiming to answer to the question: *How effective are current CNNs to reduce the effects of nuisance transformations of the input data, such as location and scaling?*

To the best of our knowledge, this has never been done in the literature, despite the keen interest in understanding the properties of CNNs [GLL09, GSS15, NYC15, SVZ14a, SZS14, YCB14, ZF14] following their empirical success. We are cognizant of the dangers of drawing sure conclusions from empirical evaluations, especially when they involve a myriad of parameters and exploit training sets that can exhibit biases. To this end, in Sect. 5.2 we describe a testing protocol that uses recognized existing modules, and keep all factors constant while testing each hypothesis.

5.1.1 Contributions

We first show that a baseline (AlexNet [KSH12]) with single-model top-5 error of 19.96% on ImageNet 2014 Classification slightly *decreases* in performance (to 20.41%) when constrained to the ground-truth bounding boxes (Table 5.1). This may seem surprising at first, as it would appear to violate Theorem 2.6.5 of [CT12] (on average, conditioning on the true value of the nuisance transformation must reduce uncertainty in the classifier). However, note that the restriction to bounding boxes does not just condition on the location-scale group, but also on *visibility*, as the image outside the bounding box is ignored. Thus, the slight decrease in performance measures the loss from discarding context by ignoring the image beyond the bounding box. When we pad the true bounding boxes with a 10-pixel rim, we show that, conditioned on such "ground-truth-with-context" indeed does decrease the error as expected, to 17.65%. In Fig. 5.1 we show the classification performance as a function of the rim size all the way to the whole image for AlexNet and VGG16 [SZ15]. A 25% rim yields the lowest top-5 errors on the ImageNet validation set for both models. This also indicates that the context effectively leveraged by current CNN architectures is limited to a relatively small neighborhood of the object of interest.

The second contribution concerns the *proper sampling* of the nuisance group. If we

Method	AlexNet		VGG16	
Whole image	19.96		13.24	
Ground-Truth Bounding Box (GT)	20.41		12.44	
	Isotropically	Anisotropically	Isotropically	Anisotropically
GT padded with 10 px	17.66	17.65	10.91	10.30
Ave-GT, 4 domain sizes (padded with [0,30] px)	15.96	16.00	9.65	8.90
Ave-GT, 8 domain sizes (padded with [0,70] px)	14.43	14.22	8.66	7.84

Table 5.1: AlexNet's and VGG16's top-5 error on the ImageNet 2014 classification challenge when the ground-truth localization is provided, compared to applying the model on the entire image. We pad the ground truth with various rim sizes both isotropically and anisotropically. Then we show how averaging the class posteriors performs when applying the network on concentric domain sizes around the ground truth.

interpret the CNN restricted to a bounding box as a function that maps samples of the location-scale group to class-conditional distributions, where the proposal mechanism *downsamples* the group, then classical sampling theory [Sha01] teaches that we should retain *not* the value of the function at the samples, but its *local average*, a process known as *anti-aliasing*. Also in Table 5.1, we show that simple uniform averaging of 4 and 8 samples of the isotropic *scale* group (leaving location and aspect ratio constant) reduces the error to 15.96% and 14.43% respectively. This is again unintuitive, as one expects that averaging conditional densities would produce less discriminative classifiers, but in line with recent developments concerning "domain-size pooling" (Chapter 2, [DS15]).

To test the effect of such anti-aliasing on a CNN absent the knowledge of ground truth object location, we follow the methodology and evaluation protocol of [FDB14] to develop a domain-size pooled CNN and test it in their benchmark classification of wide-baseline correspondence of regions selected by a generic low-level detector (MSER [MCU04]). Our third contribution is to show that this procedure improves the baseline CNN by 5–15% mean AP on standard benchmark datasets (Table 5.3 and Fig. 5.5 in Sect. 5.2.3).

Our fourth contribution goes towards answering the question set forth in the preamble: We consider two popular baselines (AlexNet and VGG16) that perform at the state-of-the-art in the ImageNet Classification challenge and introduce novel sampling and pruning methods, as well as an adaptively weighted marginalization based on the inverse Rényi entropy. Now, if *averaging* the conditional class posteriors obtained with various sampling schemes should improve overall performance, that would imply that the *implicit* "marginalization" performed by the CNN is inferior to that obtained by sampling the group, and averaging the resulting class conditionals.[2] This is indeed our observation, *e.g.,* for VGG16, as we achieve an overall performance of 8.02%, compared to 13.24% when using the whole image (Table 5.2). There are, however, caveats to this answer, which we discuss in Sect. 5.3.

Our fifth contribution is to actually provide a method that performs at the state of the art in the ImageNet Classification challenge when using a single model. In Table 5.2 we provide various results and time complexity. We achieve a top-5 classification error of 15.82% and 8.02% for AlexNet and VGG16, compared to 17.55% and 8.85% error when they are tested with 150 regularly sampled crops [SZ15], which corresponds to 9.9% and 9.4% relative error reduction, respectively. Data augmentation techniques such as scale jittering and an ensemble of several models [HZR15, SZ15, SLJ15] could be deployed along with our method.

5.1.2 Related work

The literature on CNNs and their role in Computer Vision is rapidly evolving. Attempts to understand the inner workings of CNNs are being conducted [CSV14, GLL09, GSS15, LXG15, NYC15, SVZ14a, SZS14, YCB14, ZF14], along with theoretical analysis [ARP15, BM13, CW14, SC16] aimed at characterizing their representational properties. Such intense interest was sparked by the surprising performance of CNNs [CSV14, DHG15, GDD14, HZR15, KSH12, RHG15, SEZ14, SZ15, SLJ15] in Computer Vision benchmarks [DDS09, EGW10], where many couple a proposal scheme [ADF12, CS12, CZL14, EST14, HBD15, HLR14, KK14, MGG13, RKB11, USG13, ZD14] with a CNN. As our work relates to a vast body of work, we refer the reader to references in the papers that describe the benchmarks we adopt, namely [CSV14], [KSH12] and [SZ15].

Bilen et. al. [BPT14] also explore the idea of introducing proposals in classification.

However, their approach leverages on a significantly larger number of candidates and focuses on sophisticated classifiers and post-normalization of class posteriors. Our investigation targets selecting a very small subset of the most discriminative candidates among generic object proposals, while building on popular CNN models.

5.2 Experiments

5.2.1 Large-scale Image Classification

What if we trivialize location and scaling? First, we test the hypothesis that eliminating the nuisances of location and scaling by providing a bounding box for the object of interest will improve the classification accuracy. This is not a given, for restricting the network to operate on a bounding box prevents it from leveraging on context outside it. We use the AlexNet and VGG16 pretrained models, which are provided with the MatConvNet open source library [VL15], and test their top-1 and top-5 classification errors on the ImageNet 2014 classification challenge [DDS09]. The validation set consists of $50,000$ images, where at each of them one "salient" class is annotated a priori by a human. However, other ImageNet classes appear in many of the images, which can confound any classifier.

We test the classifier in various settings (Table 5.1); first, by feeding the entire image to it and letting the classifier manage the nuisances. Then we test the ground-truth annotated bounding box and concentric regions that include it. We try both isotropic and anisotropic expansion of the ground-truth region. We observe similar behavior, which is also consistent for both models.

Only for AlexNet at Table 5.1 using the object's ground-truth support performs slightly worse than using the whole image. After we pad the object region with a 10-pixel rim, the top-5 classification error decreases fast. However, there is a trade-off between context and clutter. Providing too much context has diminishing returns. In Fig. 5.1 we show how the errors vary as a function of the rim size around the object of interest. Performance starts dropping down when we add more than 25% rim size. This padding gives 15.08% and 8.37% top-5 error for AlexNet and VGG16, as opposed to 19.96% and 13.24% respectively, when

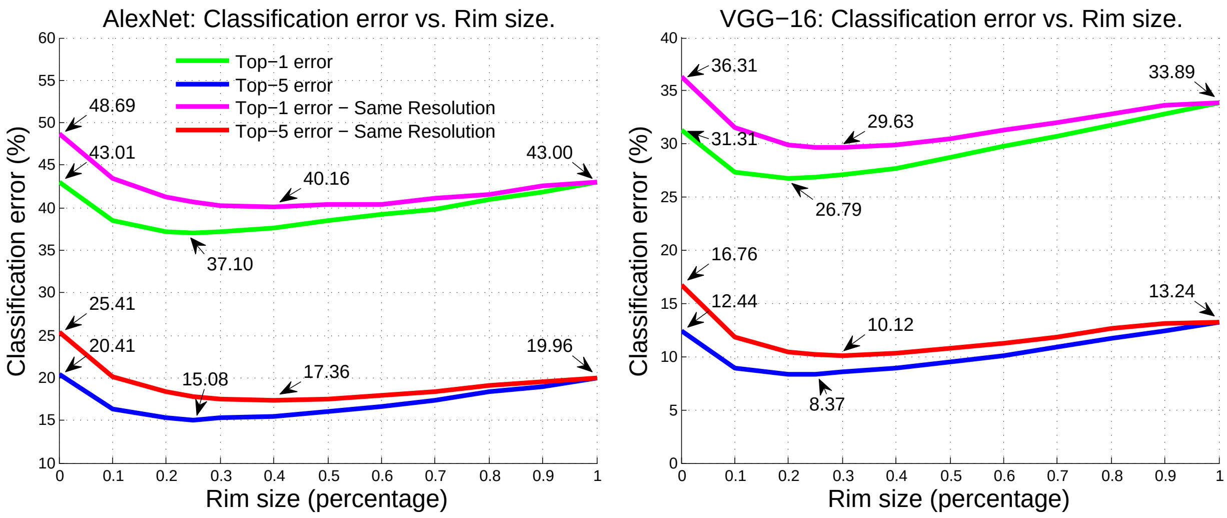


Figure 5.1: The top-1 and top-5 classification errors in ImageNet 2014 as a function of the rim size for AlexNet (above) and VGG16 (below) architecture. A 0 rim size corresponds to the ground-truth bounding box, while 1 refers to the whole image. A relatively small rim around the ground truth provides the best trade-off between informative context and clutter.

classifying the whole image.

To ensure that this improvement is not due to downsampling, we repeat the experiment with fixed resolution for the whole image and every subregion. We achieve this by shrinking each region with the same downsampling factor that we apply to the whole image to pass to the CNN. Finally we rescale the downsampled region to the CNN input. These results appear with the label "same resolution" in Fig. 5.1.

Finally, we apply domain size average pooling on the class posterior (*i.e.*, the network's softmax output layer) with 4 and 8 domain sizes that are concentric with the ground truth. The added rim has the declared size either at both dimensions (for the anisotropic case) or only along the minimum dimension (for the isotropic case), and it is uniformly sampled in the range $[0, 30]$ and $[0, 70]$, respectively. The latter one further reduces the top-5 error to 14.22% for AlexNet, which is lower than any single domain size (*c.f.* Fig. 5.1). This suggests that explicitly marginalizing samples can be beneficial. Next we test whether the improvement stands when using object proposals.

Introducing object proposals. We deploy a proposal algorithm to generate "object" regions within the image. We use Edge Boxes [ZD14], which provide a good trade-off between recall and speed [HBD15].

First, we decide the number of proposals which will provide a satisfactory cover for the majority of objects present in the dataset. In a single image we search for the highest Intersection over Union (IoU) overlap between the ground-truth region and any proposed sample and in turn we evaluate the network's performance on the most overlapping sample. We repeat this process for various number of proposals N in a small subset of validation set and finally choose $N = 80$, which provides a satisfactory trade-off between classification performance and computational cost.

Among the extracted proposals, we choose the most informative subset for our task, based on pruning criteria that we introduce later. Next we discuss what other samples we use, which are also drawn in Fig. 5.2.

Domain-size pooling and regular crops. We investigate the influence of domain-size pooling at test time both as stand-alone technique and as additional proposals for the final method which is described in Algorithm 1. We deploy domain-size aggregation of the network's class posterior over D sizes that are uniformly sampled in the range $[r, 1]$, where 1 is the normalized size of the original image. After parameter search, we choose $D = 5$ and $r = 0.6$. We use both the original and the horizontally flipped area, which gives 10 samples in total.

Finally, we use standard data augmentation techniques from the literature. As customary, the image is isotropically rescaled to a predefined size, and then a predetermined selection of crops is extracted [KSH12, SZ15, SLJ15].

Pruning samples. Continuing to sample patches within the image has diminishing return in terms of discriminability, while including more background patches with noisy class posterior distribution. We adopt an information-theoretic criterion to filter the samples that we use for the subsequent approximate marginalization.

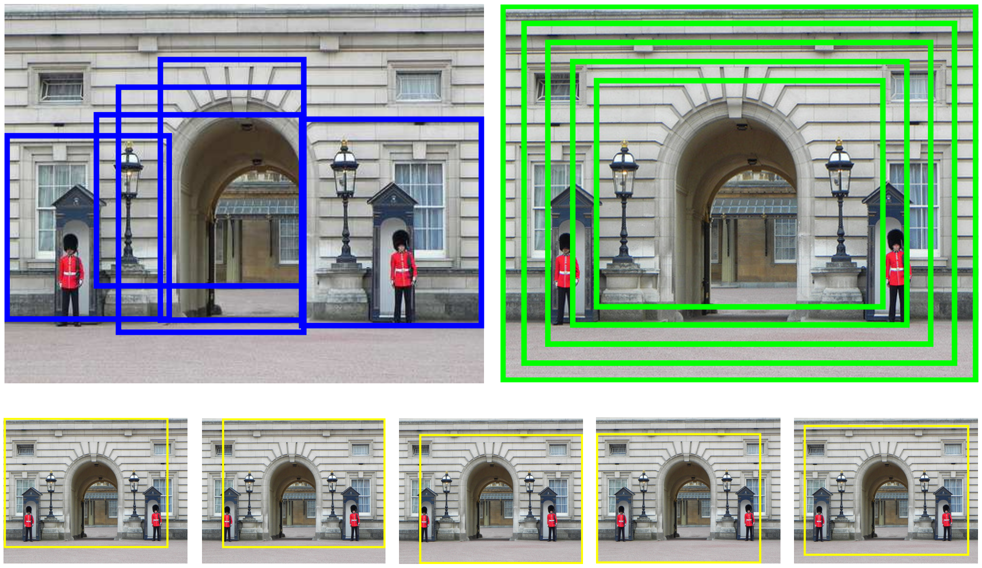


Figure 5.2: Visualizing different sampling strategies. Upper left: Object proposals. Generic proposals using Edge Boxes [ZD14]. Upper right: Concentric domain sizes are centered at the center of the image. Below: Regular crops [KSH12, SZ15, SLJ15].

For each proposal $n \in N$ we evaluate the network and take the normalized softmax output $v^n \in \mathbb{R}^{\mathcal{C}}$, where $v_i^n \in [0,1], i = \{1, \ldots, \mathcal{C}\}$ and $\mathcal{C} = 1,000$ on ILSVRC classification. The output is a set of non-negative numbers which sum up to 1. We can interpret the vector v^n as a probability distribution on the discrete space of classes $\{1, \ldots, \mathcal{C}\}$ and compute the Rényi entropy as $\mathbb{H}_\alpha(v^n) = \frac{1}{1-\alpha} log(\sum_{i=1}^{\mathcal{C}} (v_i^n)^\alpha)$.

Our conjecture is that more discriminative class distributions tend to be more peaky with less ambiguity among the classes, and therefore lower entropy. In Fig. 5.3 we show how selecting a subset of image patches whose class posterior has lower entropy improves classification performance.

We extract N candidate object proposals[3] [ZD14] and evaluate the network for both the original candidates and their horizontal flips. Then we keep a small subset E, whose posterior distribution has the lowest entropy. We use Rényi entropy with relatively small powers

[3] We introduce a prior encouraging the largest proposals among the ones that the standard setting in [ZD14] would give. To this end, instead of directly extracting, for example, $N = 80$ proposals, we generate 200 and keep the 80 largest ones (Algorithm 1).

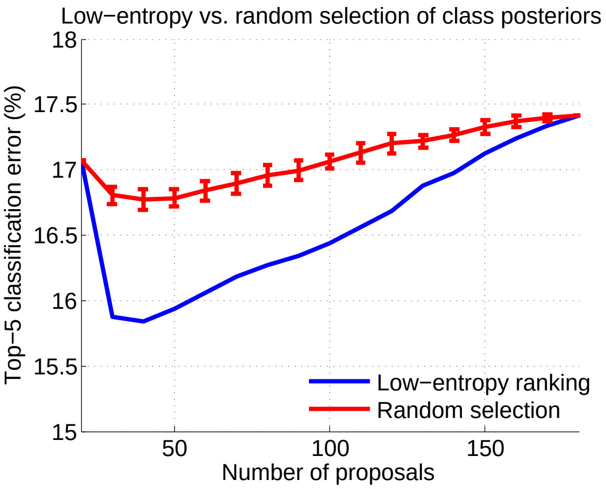


Figure 5.3: We show the top-5 error as a function of the number of proposals we average to produce the final posterior. Samples are generated with Algorithm 1 and classified with AlexNet. The blue curve corresponds to selecting samples with the lowest-entropy posteriors. We compare our method with simple strategies such as random selection, ranking by largest-size or highest confidence of proposals. The random sample selection was run 10 times and we visualize the estimated 99.7% confidence intervals as error-bars. Empirically, the discriminative power of the classifier increases when the samples are selected with the least entropy criterion.

($\alpha = 0.35$), as we found that it encourages selecting regions with more than one highly-confident candidate object. While the parameter α increases, the entropy is increasingly determined by the events of highest probability. Larger α would be more effective for images with a single object, which is not the case in most images in ILSVRC.

Finally we introduce a weighted average of the selected posteriors as $\sum_r p(c|x_{|r})p(x_{|r})$, where $x_{|r}$ is the support of sample r and $p(x_{|r})$ is the weight of its posterior[2]. We try both uniform weights and weights proportional to the inverse entropy of the posterior $p(c|x_{|r})$. The latter is expected to perform better, as it naturally gives higher weight to the most discriminative samples.

Method			AlexNet			VGG16			$\#eval$	$\#ave$
# crops	# sizes	# proposals	top-1	top-5	t (s/im)	top-1	top-5	t (s/im)		
–	$D = 1$	–	43.00	19.96	0.01	33.89	13.24	0.06	1	1
$C = 10$	–	–	41.50	18.69	0.06	27.55	9.29	0.48	10	10
$C = 50$	–	–	41.01	18.05	0.66	27.44	9.12	1.34	50	50
$C = 10 \times 3$	–	–	40.58	17.97	0.16	27.23	8.88	1.26	30	30
$C = 50 \times 3$	–	–	**40.41**	**17.55**	**0.82**	**27.14**	**8.85**	**3.48**	150	150
–	$D = 10$	–	40.00	17.86	0.08	28.16	9.46	0.60	10	10
$C = 10$	$D = 10$	–	39.38	17.08	0.22	26.94	8.83	1.08	20	20
$C = 10 \times 3$	$D = 10$	–	39.36	17.07	0.46	26.76	8.68	1.88	40	40
–	–	$E = 40$	40.18	17.53	1.26	25.60	8.24	3.02	160	40
$C = 10$	–	$E = 20$	38.91	16.63		25.28	7.91		170	30
–	$D = 10$	$E = 12$	38.05	16.19	1.34	25.19	8.11	4.38	170	22
$C = 10$	$D = 10$	$E = 12$	37.69	15.83		25.11	8.01		180	32
$C = 10$	$D = 10$	$E = 12$ (fast)	37.71	15.88	0.94	25.12	8.08	3.70	180	32
$C = 10$	$D = 10$	$E = 12$ (W, fast)	**37.57**	**15.82**	**1.28**	**25.11**	**8.02**	**3.80**	180	32
$C = 10$	$D = 10$	$E = 12$ (*test* set)	37.417	16.018	–	25.117	7.909	–	180	32

Table 5.2: Top-1 and top-5 errors on the ImageNet 2014 classification challenge. The rows 2–5 include the common data augmentation strategies in the literature [KSH12, SZ15, SLJ15] (*i.e.,* regular sampling). The next three rows use concentric domain sizes that are uniformly sampled in the range $[0.6, 1]$ with 1 being the normalized size of the original image (*c.f.* Fig. 5.2). Finally, in the last seven rows, we introduce adaptive sampling, which consists of a data-driven object proposal algorithm [ZD14] and an entropy criterion to select the most discriminative samples on the fly based on the extracted class posterior distribution. The last row shows results on the test set. $\#eval$ stands for the number of samples that are evaluated for each method, while $\#ave$ is the number of samples that are eventually element-wise averaged to produce one single vector with class confidences. The previous top-reported with regular sampling and our results are shown in bold.

Algorithm 1 Regular & adaptive sampling in classification.

- *Object proposals.* We extract several object proposals from the image x (*e.g.,* 200 *Edge Boxes* [ZD14] and keep the N largest ones). Among them we choose E proposals whose class posterior has the lowest *Rényi entropy* with parameter α. After hyper-parameter search, we choose $N = 80$, $E = 12$ and $\alpha = 0.35$.
- D *concentric* domain sizes around the center of x (including their horizontal flip). We use 5 sizes that are uniformly extracted in the normalized range $[0.6, 1]$, where 1 corresponds to the whole image ($D = 10$).
- C *crops.* Regular crops; *e.g.,* $C = 10$ or $C = 50$ in 1 or 3 scales, as in [KSH12, SZ15, SLJ15].
- The class conditionals are approximated as $\sum_r p(c|x_{|r})p(x_{|r})$, where $p(x_{|r})$ is either uniform or equals to the inverse entropy of the posterior $p(c|x_{|r})$.

Comparisons. To compare various sampling and inference strategies, we use the AlexNet and VGG16 models. All classification results in Table 5.2 refer to the validation set of the ILSVRC 2014 [DDS09], except for the last row which demonstrates results on the test set. On the rows 2–5 we show the performance of popular multi-crop methods [KSH12, SZ15, SLJ15]. Then we compare them with strategies that involve concentric domain sizes (rows 6–8) and object proposals (rows 9–14).

Before extracting the crops and in order to preserve the aspect ratio of each single image, we rescale it so that its minimum dimension is 256. The proposals are extracted at the original image resolution and then they are rescaled anisotropically to fit the model's receptive field. Additionally, some multi-crop algorithms resize the image in S different scales and then sample C patches of fixed size 224×224 densely over the image. Szegedy et al. [SLJ15] use $S = 4$ scales and $C = 36$ crops per scale, which yields 144 patches in all. Following the methodology from Simonyan et al. [SZ15], it is comparable to deploy $S = 3$ scales and extract $C = 50$ crops per scale (5×5 regular grid with flips), for a total of 150 crops over 3 scales (row 5 in Table 5.2).

The results, presented in Table 5.2, indicate as expected that scale jittering at test time improves the classification performance for both 10-crop and 50-crop strategies. Additionally, the 50-crop strategy is better than the 10-crop strategy for both models. The results on row 5 in bold are the lowest errors that can be achieved with these specific single models[4] using only regular crops.

Then we present our methods and observe that using the AlexNet network with $D = 10$ concentric domain sizes outperforms most multi-crop algorithms even if it only evaluates and averages 10 patches. Furthermore, combining it with 10 common crops achieves the best results for both networks, even without using 3-scale jittering. One interpretation for these improvements is that the concentric samples serve a natural prior for the majority of ILSVRC images, *i.e.,* the object of interest lies most probably at the center than at the image boundaries. This is a common assumption in the literature that also appears in large-scale video segmentation [KTS14].

Following, we introduce the adaptive sampling mechanism with Algorithm 1 and reduce the top-5 error to 15.83% and 8.01% for AlexNet and VGG16 respectively. To set this in perspective, Krizhevsky et al. [KSH12] report 16.4% top-5 error when they combine 5 models. We improve this performance with one single model. The relative improvement for the deployed instances of AlexNet and VGG16, compared to the data-augmentation methods used in [SZ15, SLJ15], is 9.9% and 9.4%, respectively. Row 14 shows results where the marginalization is weighted based on the entropy (notated as W), while the methods in rows 9–13 use uniform weights (*c.f.* Algorithm 1). At the last row we show results from the ILSVRC test server for our top-performing method (row 13).

Regular and concentric crops assume that objects occupy most of the image or appear near the center. This is a known bias in the ImageNet dataset. To analyze the effect of adaptive sampling, we calculate the intersection over union error between the objects and

[4] Specifically, we use the VGG16 model which is trained without scale jittering at training and appears on the first row of D area in Table 3 in [SZ15]. Pre-trained models for both AlexNet and VGG16 are publicly available with the MatConvNet toolbox [VL15]. Simonyan et al. in their evaluation with 50 crops and 3 scales report 8.6% top-5 error on ImageNet 2014 validation. In contrast our implementation produces 8.85%, which can be attributed to using a different pre-trained model, as the initial weights are sampled from a zero-mean Gaussian distribution with standard deviation 0.01 and there might also be minor differences in the training process.

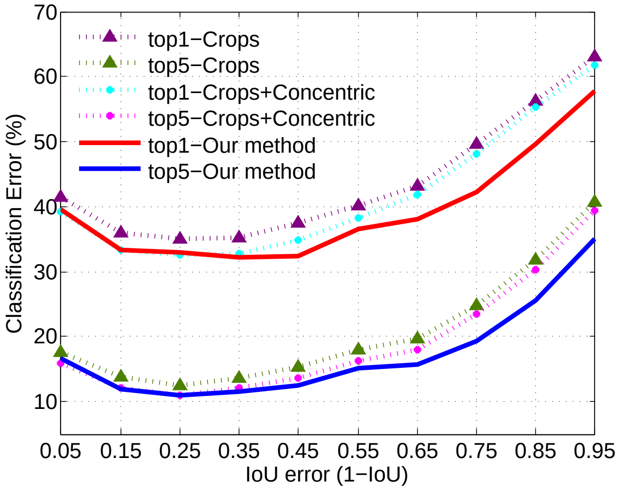


Figure 5.4: Classification error as a function of the IoU error between the objects and the regular and concentric crops.

the regular and concentric crops, and show in Fig. 5.4 the performance of various methods as a function of the IoU error. The improvement of using adaptive sampling (via proposals) over only regular and concentric crops is increased as IoU error grows, indicating that objects occupy less domain or are far away from the center.

5.2.2 Comparison between Marginalization and Max-out.

In the task of classification, nuisance variability of factors such as translation, scale and aspect ratio is explicitly handled by the use of crops, concentric domains and proposals. Each of them represents an element g in the nuisance group G. Conditioned on g, a “Category” convolutional neural network returns a conditional posterior probability of the learned classes, $p(c|x, g)$ where x is the test image. To obtain a prediction independent of the nuisance G, one can either *marginalize*

$$p(c|x) = \int p(c|x, g)dP(g), \tag{5.1}$$

and extract the classes c with maximum posterior or *max-out*

$$\hat{c} = \arg\max_{g,c} p(c|x, g). \tag{5.2}$$

over all possible elements g and classes c. The former has been extensively evaluated in the previous experiments. The latter has an additional benefit that allows to identify the nuisance element g which corresponds to the predicted class c via

$$\hat{g} = \arg\max_g p(c|x, g). \tag{5.3}$$

This helps to "localize" the object(s) of interest up to translation, scale and aspect ratio changes that are modeled by G. In this section, we evaluate the performance of max-out on the ILSVRC benchmark using the same networks (AlexNet and VGG).

As a comparison, we use the same number of crops, concentric domains and proposals as in the penultimate row of Table 5.2 ($C = 10, D = 10$ and $E = 12$). Instead of averaging the conditional posteriors, we find the maxima according to Eq. 5.2. Max-out achieves a top-1 error 40.22% and a top-5 error 17.44%. After inspecting the images where max-out fails, we observe that some of the failure cases are caused by the fact that ILSVRC only allows *one* class annotation while regions of proposals can contain other objects that are not considered the "ground-truth" class.

Time complexity. In Table 5.2 we show the number of evaluated samples (#*eval*) and the subset that is actually averaged (#*ave*) to extract a single class posterior vector. The sequential time needed for each method is linear to the number of evaluated patches #*eval*. We run the experiments with the MatConvNet library and parallelize the load for VGG16 so that the testing is done in batches of $B = 20$ patches. We report the time profile[5] for each method in Table 5.2. A few entries cover two boxes, as their methods are evaluated together. Extracting the proposals is not a major bottleneck if using an efficient algorithm [HBD15], such as Edge Boxes [ZD14]. In rows 13–14 we report results of our faster version, where the Edge Boxes do not leverage edge sharpening and use one decision tree. Overall, compared to the 150-crop strategy, the object proposal scheme introduces marginal computational overhead.

[5] We use a machine equipped with a NVIDIA Tesla K80 GPU, 24 Intel Xeon E5 cores and 64G RAM memory.

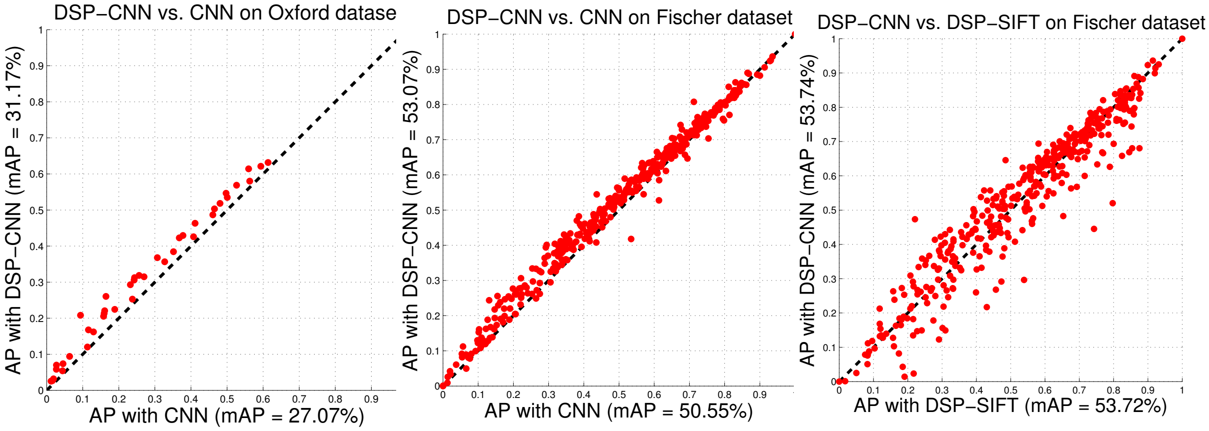


Figure 5.5: Head to head comparison between CNN and DSP-CNN on the Oxford [MTS05] (left) and Fischer's [FDB14] (center) datasets. The layer-4 features of the unsupervised network from [FDB14] are used as descriptors. The DSP-CNN outperforms its CNN counterpart in terms of matching mAP by 15.1% and 5.0%, respectively. Right: DSP-CNN performs comparably to the state-of-the-art DSP-SIFT descriptor [DS15].

5.2.3 Wide-Baseline Correspondence

We test the effect of domain-size pooling in correspondence tasks with a convolutional architecture, as done by [DS15] for SIFT [Low04], using the datasets and protocols of [FDB14]. This is illustrated in Fig. 5.2 (upper right), but here the domain sizes are centered around the detector. We expect that such averaging will increase the discriminability of detected regions and in turn the matching ability, similar to the benefits that we see on the last rows of Table 5.1.

We use maximally-stable extremal regions (MSER) [MCU04] to detect candidate regions, affine-normalize them, align them to the dominant orientation, and re-scale them for head-to-head comparisons. For a detected scale σ at each MSER, the DSP-CNN samples D domain sizes within a neighborhood $[\lambda_1\sigma, \lambda_2\sigma]$ around it, computes the CNN responses on these samples and averages the posteriors. The deployed deep network is the unsupervised convolutional network proposed by [FDB14], which is trained with surrogate labels from an unlabeled dataset (see the methodology in [DSR14]), with the objective of being invariant to several transformations that are commonly observed in images captured from different viewpoints. As opposed to network-classifiers, here the task is correspondence and the network is purely a region descriptor, whose last two layers (3 and 4) are the representations.

Method	Dim	mAP
Raw patch	4,761	34.79
SIFT [Low04]	128	45.32
DSP-SIFT [DS15]	128	**53.72**
CNN-L3 [FDB14]	9,216	48.99
CNN-L4 [FDB14]	8,192	50.55
DSP-CNN-L3	9,216	52.76
DSP-CNN-L4	8,192	53.07
DSP-CNN-L3-L4	17,408	**53.74**
DSP-CNN-L3 (PCA128)	128	51.45
DSP-CNN-L4 (PCA128)	128	52.33
DSP-CNN-L34 (concat. PCA128)	256	**52.69**

Table 5.3: Matching mean average precision for different approaches on Fischer's dataset [FDB14].

In Fig. 5.5 (left) we show the comparison between CNN and DSP-CNN on Oxford dataset [MTS05]. CNN's layer 4 is the representation for each MSER and DSP-CNN simply averages this layer's responses for all D domain sizes. We use $\lambda_1 = 0.7$, $\lambda_2 = 1.5$ and $D = 6$ sizes that are uniformly sampled in this neighborhood. There is a 15.1% improvement based on the matching mean average precision.

Fischer's dataset [FDB14] includes 400 pairs of images, some of them with more extreme transformations than those in the Oxford dataset. The types of transformations include zooming, blurring, lighting change, rotation, perspective and nonlinear transformations. In Fig. 5.5 (center) and Table 5.3 we show comparisons between CNN and DSP-CNN for layer-3 and layer-4 representations and demonstrate 7.7% and 5.0% relative improvement. We use $\lambda_1 = 0.5$, $\lambda_2 = 1.4$ and $D = 10$ domain sizes. These parameters are selected with cross-validation. In Table 5.3 we show comparisons with baselines, such as using the raw data and DSP-SIFT [DS15]. After fine parameter search ($\lambda_1 = 0.5$, $\lambda_2 = 1.24$) and concatenating the layers 3 and 4, we achieve state of the art performance as shown in Fig. 5.5 (right), observing though the high dimensionality of this method compared to local descriptors.

Given the inherent high-dimensionality of CNN layers, we perform dimensionality reduction with principal component analysis to investigate how this affects the matching performance. In Table 5.3 we show the performance for compressed layer-3 and layer-4 representations with PCA to 128 dimensions and their concatenation. There is a modest performance loss, yet the compressed features outperform the single-scale features by a large margin.

5.3 Discussion

Our empirical analysis indicates that CNNs, that are designed to be invariant to nuisance variability due to small planar translations – by virtue of their convolutional architecture and local spatial pooling – and learned to manage global translation, distance (scale) and shape (aspect ratio) variability by means of large annotated datasets, in practice are less effective than a naive and in theory counter-productive practice of sampling and averaging the conditionals based on an ad-hoc choice of bounding boxes and their corresponding planar translation, scale and aspect ratio.

This has to be taken with the due caveats: First, we have shown the statement empirically for *few* choices of network architectures (AlexNet and VGG), trained on *particular* datasets that are unlikely to be representative of the complexity of visual scenes (although they may be representative of the same scenes as portrayed in the test set), and with a specific choice of *parameters* made by their respective authors, both for the classifier and for the evaluation protocol. To test the hypothesis in the fairest possible setting, we have kept all these choices constant while comparing a CNN trained, in theory, to "marginalize" the nuisances thus described, with the same applied to bounding boxes provided by a proposal mechanism. To address the arbitrary choice of proposals, we have employed those used in the current state-of-the-art methods, but we have found the results representative of other choices of proposals.

In addition to answering the question posed in the introduction, along the way we have shown that by framing the marginalization of nuisance variables as the averaging of a *sub-sampling* of marginal distributions we can leverage of concepts from classical sampling theory to *anti-alias* the overall classifier, which leads to a performance improvement both in categorization, as measured in the ImageNet benchmark, and correspondence, as measured in the Oxford and Fischer's matching benchmarks.

Of course, like any universal approximator, a CNN can in principle capture the geometry of the discriminant surface by "learning away" nuisance variability, given sufficient resources in terms of layers, number of filters, and number of training samples. So in the abstract sense a CNN *can* indeed marginalize out nuisance variability. The analysis conducted show that, at the level of complexity imposed by current architectures and training set, it does so less effectively than ad-hoc averaging of proposal distributions.

This leaves researchers the choice of investing more effort in the design of proposal mechanisms [Gir15, RHG15], subtracting duties from the Category CNN downstream, or invest more effort in scaling up the size and efficiency of learning algorithms for general CNNs so as to render the need for a proposal scheme moot.

CHAPTER 6

Visual-Inertial-Semantic Scene Representation for 3D Object Detection

6.1 Introduction

In this chapter, we describe a system to detect objects in three-dimensional space using video and inertial sensors (accelerometer and gyrometer), ubiquitous in modern mobile platforms from phones to drones. Inertials afford the ability to impose class-specific scale priors for objects, and provide a global orientation reference. A minimal sufficient representation, the posterior of semantic (identity) and syntactic (pose) attributes of objects in space, can be decomposed into a geometric term, which can be maintained by a localization-and-mapping filter, and a likelihood function, which can be approximated by a discriminatively-trained convolutional neural network. The resulting system can process the video stream causally in real time, and provides a representation of objects in the scene that is persistent: Confidence in the presence of objects grows with evidence, and objects previously seen are kept in memory even when temporarily occluded, with their return into view automatically predicted to prime re-detection.

We deem an "object detector" to be a system that takes as input *images* and produces as output decisions as to the presence of *objects in the scene.* We design one based on the following premises: (a) Objects exist in the scene, not in the image; (b) they persist, so confidence on their presence should grow as more evidence is accrued from multiple (test) images; (c) once seen, the system should be aware of their presence even when temporarily not visible; (d) such awareness should allow it to predict when they will return into view, based on scene geometry and topology; (e) objects have characteristic shape and *size* in 3D, and vestibular (inertial) sensors provide a global scale and orientation reference that the system should leverage on.

Detecting objects from images is not the same as detecting images of objects (Fig. 6.5). Objects do not flicker in-and-out of existence, and do not disappear when not seen (Fig. 6.6). What we call "object detectors" traditionally refers to algorithms that process a single image and return a decision as to the presence of objects of a certain class in said image, missing several critical elements (a)-(e) above. Nevertheless, such algorithms can be modified to produce *not* decisions, but *evidence* (likelihood) for the presence of objects, which can be processed over time and integrated against the geometric and topological structure of the *scene*, to yield an object detector that has the desired characteristics. The scene context encompasses both the identity and co-occurrence of objects (semantics) but also their spatial arrangement in three-dimensional (3D) space (syntax).

6.1.1 Summary of Contributions and Limitations

To design an object detector based on the premises above, we (a) formalize an explicit model of the posterior probability of object attributes, both semantic (identity) and syntactic (pose), natively in the 3D scene (Sect. 6.3), which (b) maintains and updates such a posterior, processing each image causally over time (Sect. 6.3.2); (c) the posterior distribution is a form of short-term memory (representation), which we use to (d) predict visibility and occlusion relations (Sect. 6.5.3). We exploit the availability of cheap inertial sensors in almost every mobile computing platform to (e) impose class-specific priors on the size of objects (Sect. 6.5.2).

The key insight from the formalization (a) above is that an optimal (minimal sufficient invariant [SC16]) representation for objects in the scene (Eq. 6.1) can be factored into two components: One geometric – which can be computed recursively by a localization (SLAM) system (Eq. 6.3) – and the other a likelihood term, which can be evaluated instantaneously by a discriminatively-trained convolutional neural network (CNN, Eq. 6.4) operating on a single image. Some consequences of this insight are discussed in Sect. 6.6. In practice, this

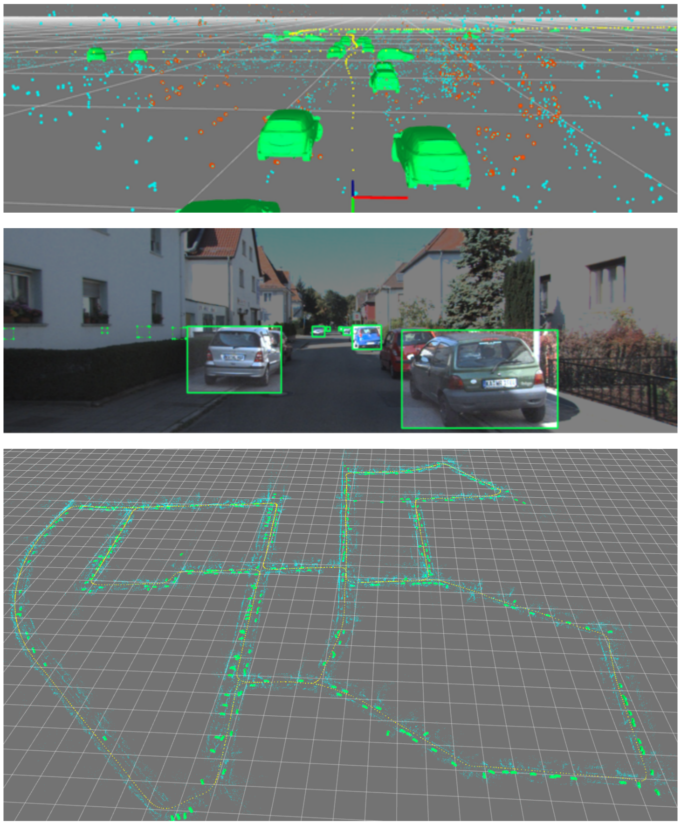


Figure 6.1: Illustration of our system to detect objects-in-scenes. Top: state of the system with reconstructed scene representation (cyan), currently tracked points (red), viewer trajectory from a previous loop (yellow) and current pose (reference frame). All cars detected are shown as point-estimates (the best-aligned generic CAD model) in green, including those previously-seen on side streets (far left). Middle: visualization of the implicit measurement process: Objects in the state are projected onto the current image based on the mean vehicle pose estimate (green boxes) and their likelihood score is computed (visualized as contrast: sharp regions have high likelihood, dim regions low). Cars in different streets, known to not be visible, are visualized as dashed boxes and their score discarded. Bottom: Top view of the state from the entire KITTI-00 sequence (best viewed at 5×).

means that we can implement our system using some off-the-shelf components, fine-tuning a pre-trained CNN, and at least for some rudimentary modeling assumptions, our system operates in real-time, generating object-scene representations at 10-30 frames per second.

In Sect. 6.5 we report the results of a representative sample of qualitative and quantitative tests.

Our system is the first to exploit inertial sensors to provide both scale discrimination and global orientation for visual recognition (Fig. 6.5). Most (image)-object detectors assume images are gravity-aligned, which is a safe bet for photographic images, not so for robots or drones. Our system is also the first to integrate CNN-based detectors in a recursive Bayesian inference scheme, and to implement the overall system to run in real-time [DFK16].

While our formalization of the problem of object detection is general, our real-time implementation has several limitations. First, it only returns a joint geometric and semantic description for *static objects*. Moving objects are detected in the image, but their geometry – shape and pose, estimating which would require sophisticated class-specific deformation priors – is not inferred. Second, it models objects' shape as a parallelepiped, or bounding box in 3D. While this is a step forward from bounding boxes in the image, it is still a rudimentary model of objects, based on which visibility computation is rather crude. We have performed several tests with dense reconstruction [GBS15], as well as with CAD models [ISS16], but matching and visibility computation based on those is not yet at the level of accuracy (dense reconstruction) or efficiency (CAD matching) to enable real-time computation. The third limitation is that a full joint syntactic-semantic prior is not enforced. While ideally we would like to predict not only what objects are likely to become visible based on context, but also *where* they will appear relative to each other, this is still computationally prohibitive at scale.

In Sect. 6.3 we start by defining an object representation as a sufficient invariant for detection, and show that the main factor can be updated recursively as an integral, where the measure represents the syntactic context, and can be computed by a SLAM system, and the other factor can be computed by a CNN. While the update is straightforward and *top-down* (the system state generate predictions for image-projections, whose likelihood is scored by a CNN), initialization requires defining a prior on object identity and pose. For this we use the same CNN in a *bottom-up* mode, where putative detection (high-likelihood regions) are used to initialize object hypotheses (or, rather, regions with no putative detections are assumed free of objects), and several heuristics are put in place for genetic phenomena (birth, death and merging of objects, Sect. 6.4).

6.2 Related Work

This work, by its nature, relates to a vast body of literature on scene understanding in Computer Vision, Robotics [LBR12, PL15] and AI [KAJ11] dating back decades [Wal81]. Most recently, with the advent of cheap consumer range sensors, there has been a wealth of activity in this area [LFU13, TTD12, WLS14, CK13, SK13, DTL15, GAM13, KMF13, SNS13, HFL14, BS15, SGS13, VML15, KMT16, SX16, RS15]. The use of RGB-D cameras unfortunately restricts the domain of applicability mostly indoors and at close range whereas we target mobility applications where the camera, which typically has an inertial sensor strapped on it, but not (yet) a range sensor, can be used both indoor and outdoors. We expect that, on indoor sequences, our method would underperform a structured light or other RGB-D source, but this is subject of future investigation.

There is also work that focuses on scene understanding from visual sensors, specifically video [KLD14, AYB15, LSH16, SHK12, BGC15, YFU12], although none integrates inertial data, despite a resurgent interest in sensor fusion [ZCV15]. Additional related work includes [HZC13, CLC08, BSF08, SHP15].

To the best of our knowledge, no work leverages inertial sensing for object detection. This is critical to provide a scale estimate in a monocular setting, and validate object hypotheses in a Bayesian setting, so that, for instance, a model car in our system is not classified as a car (Fig. 6.5).

Semantic scene understanding *from a single image* is also an area of research ([FHG15] and references therein). We are instead interested in agents embedded in physical space, for which the restriction to a single image is limiting. There is also a vast literature on scene segmentation ([HHX15] and references therein), mostly using range (RGB-D) sensors. One popular pipeline for dense semantic segmentation is adopted by [HFL14, MHD16, VML15, KLD14, ABS16]: Depth maps obtained either from RGB-D or stereo are fused; 2D semantic labeling is transferred to 3D and smoothed with a fully-connected CRF [Kol11]. Also related methods on joint semantic segmentation and reconstruction are [SHL16, UBG16, BVR16].

There is also work on 3D recognition [KKS13, STL14, MCL14], but again with no inertial measurements and no motion. Some focus on real-time operation [CFN14], but most operate off-line [ZSS15, CRU16]. None of the datasets commonly used in these works [COR16, XOT13] provide an inertial reference, except for KITTI. In terms of 3D object detection on KITTI, some authors focus on image-based detection [GDD14, Gir15, RHG15, RDG16, LAE16] and then place objects into the scene [XCL15, XCL17], while others focus on 3D object proposal generation and verification using a network [CKZ16, CKZ15]. [XCL15] trains a 3D Voxel Pattern (3DVP) based detector to infer object attributes and demonstrates the ability to accurately localize cars in 3D on KITTI. Their subsequent work [XCL17] trains a CNN to classify 3DVPs. Different representations of object proposals are also exploited, such as 3D cuboids [FDU12] and deformable 3D wireframes [ZSS15]. Various priors are also considered: [WFU15] exploits geo-tagged images; geometric priors of objects are incorporated into various optimization frameworks to estimate object attributes [ZZD15, CRU16]. While most of these algorithms report very good performance on detection ($\sim 90\%$ mean average precision), none reports scores for the semantic-syntactic state of objects in 3D, except for [XCL15, XCL17] and [CKZ15, CKZ16]. Since the latter are dominated by the former, we take [XCL17] as a paragon for comparison in Sect. 6.5.

The aforementioned 3D object recognition methods are based on 2D detection without temporal consistency. Therefore, the comparison is somewhat unfair as single-image based detectors cannot reliably detect objects in space, which is our main motivation for the proposed approach. For details on comparison methodology, see Sect. 6.5. [CRU16, SC15] use multiple views, but their output is a point-estimate instead of a posterior. Also, the optimization has to be re-run once new datum is available.

Recent work in data association [LZD16] aims to directly infer the association map, which is computationally prohibitive for the scale needed in our real-time system. We therefore resort to heuristics, described in Sect. 6.4. More specifically to our implementation, we leverage existing visual-inertial filters [HKB13, LM14, TCS15] and single image-trained

CNNs [GDD14, RDG16, XCL17].

6.3 Methods

6.3.1 Representations

A scene ξ is populated by a number of objects $z_j \in \{z_1, \dots, z_N\}$, each with geometric (pose, shape)[1] and semantic (label) attributes $z_j = \{s_j, l_j\}$. Measurements (*e.g.,* images) up to the current time t, $y^t \doteq \{y_1, \dots, y_t\}$ are captured from a sensor at pose g_t. A *semantic* representation of the scene is the joint posterior $p(\xi, z^j | y^t)$ for up to the j-th objects seen up to time t, where sensor pose g_t and other nuisances are marginalized. The joint posterior can be decomposed as $p(\xi, z^j|y^t) = p(\xi|z^j)p(z^j|y^t)$ with the first factor ideally updated asynchronously each time a new object z_{j+1} becomes manifest starting from a prior $p(\xi)$ and the second factor updated each time a new measurement y_{t+1} becomes available starting from $t = 0$ and given $p(z)$.

A representation of the scene in support of *(geometric) localization tasks* is the posterior $p(g_t, x|y^t)$ over sensor pose g_t (which, of course, is not a nuisance for this task) and a sparse attributed[2] point cloud $x = [x_1, \dots, x_{N_x}]$, given all measurements (visual I^t and inertial u^t) up to the current time. Conditioning the semantics on the geometry we can write the second factor above as

$$p(z^j|y^t) = \int p(z^j|g_t, x, y^t) dP(g_t, x|y^t) \tag{6.1}$$

where the integrand can be updated as more data y_{t+1} becomes available as $p(z^j|g_{t+1}, x, y^{t+1})$, which is proportional to

$$p(y_{t+1}|z^j, g_{t+1}, x) \int p(g_{t+1}|g_t, u_t) dP(z^j|g_t, x, y^t). \tag{6.2}$$

[1] Object pose is its position and orientation in world frame. With inertials, pose can be reduced to position and rotation around gravity. Sensor pose is full 6 degree-of-freedom position and orientation.

[2] Attributes include sparse geometry (position in the inertial frame) and local photometry (feature descriptor, sufficient for local correspondence).

6.3.2 Approximations

The measure in (6.1) can be approximated in wide-sense using an Extended Kalman Filter (EKF), as customary in simultaneous localization and mapping (SLAM): $p(g_t, x|y^t) \simeq \mathcal{N}(\hat{g}_{t|t}, \hat{x}_{t|t}; P_{t|t})$. (6.1) is a diffusion around the mean/mode $\hat{g}_{t|t}, \hat{x}_{t|t}$; if the covariance $P_{t|t}$ is small, it can be further approximated: Given

$$\hat{g}_{t|t}, \hat{x}_{t|t} = \arg\max_{g_t, x} p_{\text{SLAM}}(g_t, x|y^t), \tag{6.3}$$

$\hat{p}_{g,x}(z^j|y^t) \doteq p(z^j|g_t = \hat{g}_{t|t}, x = \hat{x}_{t|t}, y^t) \simeq p(z^j|y^t)$. Otherwise the marginalization in (6.1) can be performed using samples from the SLAM system. Either way, omitting the subscripts, we have

$$\hat{p}(z|y^{t+1}) \propto \underbrace{p(y_{t+1}|z, \hat{g}_{t|t}u_t, \hat{x}_{t|t})}_{\text{CNN}} \underbrace{\hat{p}(z|y^t)}_{\text{BF}} \tag{6.4}$$

where the likelihood term is approximated by a convolutional neural network (CNN) as shown in Sect. 6.3.3 and the posterior is updated by a Bayesian filter (BF) approximated by a bank of EKFs (Sect. 6.3.4). That only leaves the first factor $p(\xi|z^j)$ in the posterior, which encodes context. While one could approximate it with a recurrent network, that would be beyond our scope here; we even forgo using the co-occurrence prior, which amounts to a matrix multiplication that rebalances the classes following [CLT10], since for the limited number of classes and context priors we experimented with, it makes little difference.

Approximating the likelihood in (6.4) appears daunting because of the purported need to generate future data y_{t+1} (the color of each pixel) from a given object class, shape and pose, and to normalize with respect to all possible images of the object. Fortunately, the latter is not needed since the product on the right-hand side of (6.4) needs to be normalized anyway, which can be done easily in a particle/mixture-based representation of the posterior by dividing by the sum of the weights of the components. Generating actual images is similarly not needed. What is needed is a mechanism that, for a given image y_{t+1}, allows quantifying the likelihood that an object of *any* class with *any* shape being present in *any* portion of the image where it projects to from the vantage point g_t. In Sect. 6.3.3 we will show how a discriminatively-trained CNN can be leveraged to this end.

6.3.3 Measurement Process

At each instant t, an image I_t is processed by "probing functions" ϕ, which can be designed or trained to be invariant to nuisance variability. The SLAM system processes all past image measurements I^t and current inertial measurements u_t, which collectively we refer to as $y_t = \{\phi_\kappa(I_t), u_t\}$, where $\phi_\kappa(I_t)$ is a collection of sparse contrast-invariant feature descriptors computed from the image for N_i visible regions of the scene, and produces a joint posterior distribution of poses g_t and a sparse geometric representation of the scene $x = [x_1, \dots, x_{N_i(t)}]$, assumed uni-modal and approximated by a Gaussian:

$$p_{\text{SLAM}}(g_t, x|y^t) \simeq \mathcal{N}(\hat{g}_{t|t}, \hat{x}_{t|t}; P_{\{g,x\}\,t|t}) \tag{6.5}$$

where $x \in \cup_j s_j$, *i.e.,* the scene is assumed to be composed by the union of objects, including the default class "background" l_0. This localization pipeline is borrowed from [TCS15], and is agnostic of the organization of the scene into objects and their identity. It also restricts x to a subset of the scene that is rigid, co-visible for a sufficiently long interval of time, and located on surfaces that, locally, exhibit Lambertian reflection.

To compute the marginal likelihood for each class $l_k \in \{l_0, \dots, l_K\}$, we leverage on a CNN trained discriminatively to classify a given image region b_j into one of $K+1$ classes, including the background class. The architecture has a soft-max layer preceded by $K+1$ nodes, one per class, and is trained using the cross-entropy loss, providing a normalized score $\phi_{CNN}(l|I_{t_{|b_j}})_{[k]}$ for each class and image bounding box b_j. We discard the soft-max layer, and forgo class-normalization. The activations at the $K+1$ nodes in the penultimate layer of the resulting network provide a mechanism for, given an image I_t, quantifying the likelihood of each object class l_k being present at each bounding box b_j, which we interpret the (marginal) likelihoods for (at least an instance of) each class being present at the given bounding box:

$$\phi_{\text{CNN}}(l|I_{t_{|b_j}})_{[k]} \simeq p(I_t|l_k, b_j). \tag{6.6}$$

This process induces a likelihood on object classes being present in the *visible portion of the scene* regions of s_j and corresponding vantage points g_t, via $b_j = \pi(g_t s_j)$ where π is the projection. Since inertials u_t are directly measured, up to a Gaussian noise, we have:

$$p(y_t|z^j, g_t, x) \simeq \phi_{\text{CNN}}(l|I_{t_{|\pi(g_t s_j)}})_{[k]}\mathcal{N}(\bar{u}; Q) \tag{6.7}$$

where $\bar{u}$ are the inertial biases and Q the noise covariance; here the object attributes z^j are the labels $l_j = l_k$ and geometry s_j. Thus, given an image I_t, for each possible object pose and shape s_j and vantage point g_t, we can test the presence of at least one instance of each class l_k within. Note that the visibility function is implicit in the map π. If an object is not visible, its likelihood given the image I_t is constant/uniform. Note that this depends on the global layout of the scene, since the map π must take into account occlusions, so objects cannot be considered independently.

6.3.4 Dependencies and Co-visibility

Computing the likelihood of an object being present in the scene requires ascertaining whether it is visible in the image, which in turn depends on all other objects, so the scene has to be modeled holistically rather than as an independent collection of objects. In addition, the presence of certain objects, and their configuration, affects the probability that other objects that are not visible be present.[3]

To capture these dependencies, we note that the geometric representation $p(g_t, x|y^t)$ can be used to provide a joint distribution on the position of all objects and cameras $p(g^t, x|y^t)$, which yields *co-visibility* information, specifically the probability of each point in x being visible by any camera in g^t. It is, however, of no use in determining visibility of objects, since it contains no topological information: We do not know if the space between two points is empty, or occupied by an object void of salient photometric features.To enable visibility computation, we can use the point cloud together with the images to compute the *dense shape* of objects in a maximum-likelihood sense: $\hat{s}_j = \arg\max p(s_j|g^t, x, y^t)$ using generic regularizers. This can be done but not at the level of accuracy and efficiency needed for

[3] For instance, seeing a keyboard and a monitor on a desk affects the probability that there is a mouse in the scene, even if we cannot see it at present. Their relative pose also informs the vantage point that would most reduce the uncertainty on the presence of the mouse.

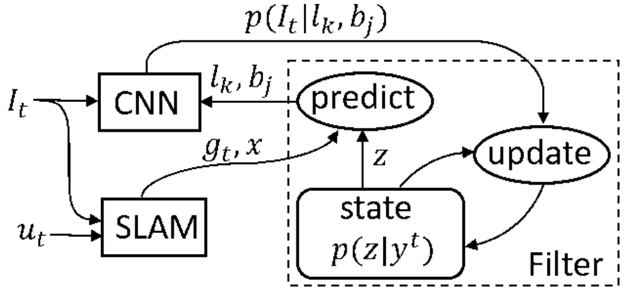


Figure 6.2: System Flow Chart.

live operation. An alternative is to approximate the shape of objects with a parametric family, for instance cuboids or ellipsoids, and compute visibility accordingly, also leveraging the co-visibility graph computed as a corollary from the SLAM system and priors on the size and aspect ratios of objects. To this end, we approximate

$$\hat{p}_{g,x}(z^j|y^t) \doteq p(z^j|y^t, g_t, x) \simeq \prod_j p(z_j|y^t, g_t, x, z^{-j}) \tag{6.8}$$

where z^{-j} indicates all objects but z_j. Each factor $p(s_j, l_j|y^t, g_t, x, z^{-j})$ is then expanded as the product

$$\underbrace{p(s_j|l_j, y^t, g_t, x, s^{-j})}_{\text{EKF}}\underbrace{P(l_j|y^t, g_t, x, l^{-j})}_{\text{PMF}} \tag{6.9}$$

where PMF indicates a probability mass filter; this effectively yields a bank of class-conditional EKFs. These provide samples from $\hat{p}(z|y^t)$ in the right-hand side of (6.4), that are scored with the CNN to update the posterior.

6.4 Implementation Details

We have implemented two renditions of the above program: One operating in real-time and demonstrated live in June 2016 [DFK16]. The other operating off-line and used for the experiments reported in Sect. 6.5. Fig. 6.2 sketches the system flow chart.

In both cases, we have taken some shortcuts to improve the efficiency of the approximation of the likelihood function implemented by a CNN. Also, the semantic filter needs initialization and data association, which requires some heuristics to be computationally viable. We describe such heuristics in order.

Visual Odometry and Baseline 2D CNN We use robust SLAM implemented from [TCS15] to acquire sparse point clouds and camera pose x, g_t at each t. This occurs in $10-20$ms per VGA frame. For the quantitative evaluation on KITTI, we use [MMT15] as the underlying localization pipeline. For our real-time system, we use YOLO [RDG16] as a baseline method to compute object likelihoods in $150-200$ms, whereas in the off-line system we use SubCNN [XCL17]. In either case, the result is, for each given window, a positive score for each class k, read out from the penultimate layer. These are used both to compute the likelihood, and to generate proposals for initialization as discussed later.

Filter Organization Each object is represented by a PMF filter over class labels and K class-conditional EKFs, one for each class (6.9). Thus each object is represented by a mixture of K EKFs, some of which pruned as we describe later. Each maintains a posterior estimate of position, scale and orientation relative to gravity. The state predicts the projection of (each of the K instances of) each object onto the image plane, where the CNN evaluates the likelihood. For some object classes, we use a shape prior, enforced as a pseudo-measurement with uncertainty manually tuned to the expected class-variability. For instance, people are parallelepipeds of $1m^3$ expected volume with an anisotropic covariance along coordinate axes in the range of few decimeters, whereas couches have significantly more uncertainty.

Data Association To avoid running the baseline CNN multiple times on overlapping regions (each object is represented by multiple, often very similar, regions, one per each current class hypothesis), we do not query the CNN sequentially for each prediction. Instead, we run the CNN once, with lax threshold so as to obtain a large number of (low-confidence) regions. While this is efficient, it does create a data association problem, as we must attribute (possibly multiple) image regions to each (of multiple) object hypotheses, each of which has multiple possible class labels [AZD14]. We avoid explicit data association by opting simple heuristics instead: first we generate predictions from the filter; then occluded objects are excluded from likelihood evaluation. For all others, we generate four-tuple coordinates of the bounding box, as a 4-dimensional Gaussian given the projection of the current state. This is a sloppy prediction, for the image of a parallelepiped is in general not an axis-aligned rectangle on the image. Nevertheless, we use this for scoring the use of the likelihood produced by the CNN for each predicted class. A (class-dependent) threshold is used to decide if the bounding box should be used to update the object. Bounding boxes with lower likelihood are given small weights in the filter update. This requires accurate initialization, which we will describe below. The silver lining is that inter-frame motion is usually small, so data association proceeds smoothly, unless multiple instances of the same object class are present nearby and partially occlude each other.

Initialization Putative 2D CNN detections not associated to any object are used as (bottom-up) proposals for initialization. The new object is positioned at the weighted centroid of the sparse points whose projections lie within the detection region. The weight at center is the largest and decreases exponentially outwards. Orientation is initialized as the "azimuth" from SubCNN, rotated according to camera pose and gravity. Given the position and orientation, scale is optimized by minimizing the reprojection error.

Merge Objects are assumed to be simply-connected and compact, so two objects cannot occupy the same space. Yet, their projected bounding boxes can overlap. If multiple instances from the same object are detected, initialized and propagated, they will eventually merge when their overlap in space is sufficiently large. Only objects from the same class are allowed to merge as different classes may appear co-located and intersecting in their sloppy parallelepipedal shape model, *e.g.,* a chair under a table.

Termination Each object maintains a probability over K classes, each associated with a class-conditional filter. If one of the classes becomes dominant (maximum probability above a threshold), all other filters will be eliminated to save computational cost. Most objects converge to one or two classes (*e.g.,* chair, couch) within few iterations. Objects that disappear from view are retained in the state (short-term memory), and if not seen for a sufficiently long time, they are stored in long-term memory ("semantic map") for when they will be seen again.

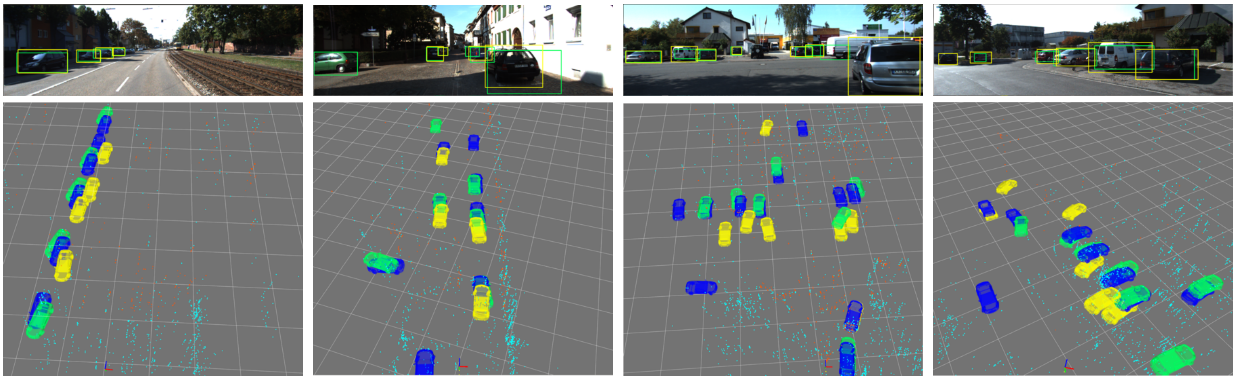


Figure 6.3: Qualitative comparison with SubCNN. Top: Images with back-projected objects from our method (Green), the same with SubCNN (Yellow). Bottom: top-view of the corresponding portion of the scene. Ground truth is shown in Blue.

	Position error	< 0.5 m			< 1 m			< 1.5 m		
Orientation error	method	#TP	Precision	Recall	#TP	Precision	Recall	#TP	Precision	Recall
$< 30^\circ$	Ours-FNL	**150**	**0.14**	**0.10**	**355**	**0.34**	**0.24**	**513**	**0.49**	**0.35**
	Ours-INST	135	0.13	0.09	270	0.26	0.18	368	0.35	0.25
	SubCNN	99	0.10	0.07	254	0.26	0.17	376	0.38	0.26
$< 45^\circ$	Ours-FNL	**157**	**0.15**	**0.11**	**367**	**0.35**	**0.25**	**533**	**0.50**	**0.36**
	Ours-INST	141	0.13	0.10	283	0.27	0.19	388	0.37	0.26
	SubCNN	99	0.10	0.07	257	0.26	0.17	383	0.38	0.26
–	Ours-FNL	**169**	**0.16**	**0.11**	**425**	**0.40**	**0.29**	**618**	**0.58**	**0.42**
	Ours-INST	149	0.14	0.10	320	0.30	0.22	450	0.43	0.31
	SubCNN	104	0.10	0.07	272	0.27	0.18	409	0.41	0.28

Table 6.1: Quantitative evaluation on KITTI and comparison with SubCNN [XCL17]. The number of true positives having positional error (row), and angular error (column) less than a threshold is shown, along with Precision and Recall. Scores are aggregated across all 3501 ground-truth labeled frames in the dataset, with 498 annotated objects. The last 3 rows discard orientation error.

6.5 Experiments

6.5.1 Quantitative Results

As explained in Sec. 6.2, we choose SubCNN [XCL17] as the paragon, even though it is based on a single image, because it is the top performer for 3D recognition in KITTI among non-anonymous and reproducible ones, in particular it dominates [CKZ16]. Being single-image based, SubCNN returns different results in each frame, therefore naturally at a disadvan-

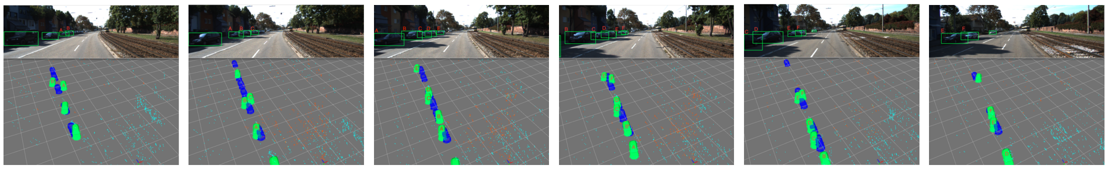


Figure 6.4: Evolution of the state (Green) against ground-truth annotation (Blue) (best viewed at 5×, images shown at the top for ease of reference). When first seen (Leftmost) cars 'A' and 'B' are estimated to be side-by-side; after a few frames, however, 'A' and 'B' fall into place, but a new car 'C' appears to flank 'B'. As time goes by, 'C' too falls into place, as new cars appear, 'D', 'E', 'F.' The error in pose (position and orientation) relative to ground truth can be appreciated qualitatively. Quantitative results are shown in Table 6.1.

tage. To make the comparison fair, one would have to average or integrate detections for each object across all frames when it is visible. However, SubCNN does not provide data association, making direct comparison challenging. To make comparison as fair as possible, without developing an alternate aggregation method for SubCNN, we compare it to our algorithm on a frame-by-frame basis. Specifically, for each frame, we transfer the ground truth to the camera frame, and remove occluded objects. Then we can compare detections from SubCNN to our point estimate (conditional mean) computed causally by the filter at the current time. We call this method Ours-INST. On the other hand, we can benefit from aggregating temporal information for as long as possible, so we also report results based on the point-estimate of the filter state at the last time instant when each object is seen. The estimate is then mapped back to the current frame, which we call Ours-FNL. To the best of our knowledge, there are no known methods for 3D recognition that causally update posterior estimates of object identity/presence and geometric attributes, and even naive temporal averaging of a method like [XCL17] is not straightforward because of the absence of data association across different frames. This is precisely what motivates us.

6.5.1.1 Dataset

There are many datasets for image-based object detection [EGW10, RDS15] which provide 2D ground truth. There are also 3D object detection datasets [XOT13], most using extra sensor data, *e.g.,* depth from a structured-light sensor. None provide inertial measurements, except KITTI [GLS13], whose object detection benchmark contains 7181 images, from which we exclude 3682 frames used for SubCNN training [XCL17], leaving us a validation set of 3799 frames. We then find 10 videos which cover most of the validation set. After removing moving objects, 498 objects are observed 18468 times at 3501 instants, which is the same order of magnitude of the 2D validation set.

6.5.1.2 Evaluation Metrics

KITTI provides ground-truth object *tracklets* we use to define true positives, miss detections and false alarms. A *true positive* is the nearest detection of a ground truth object within a specified error threshold in both position and orientation (Table 6.1). A *miss* occurs if there is no detection within the threshold. A *false alarm* occurs when an object is detected despite no true object being within the threshold in distance and orientation. *Precision* is the fraction of true positives over all detections, and *Recall* is the percentage of detected instances among all true objects.

6.5.1.3 Benchmark Comparison

Table 6.1 shows result on the KITTI dataset, averaged over all sequences. On average, Ours-INST already outperforms SubCNN even if our initialization can be rather inaccurate. Note that our method requires evidence to be accumulated over time before claiming the existence of an object in the scene, so Ours-INST is penalized heavily in the first few frames when a new object is spotted. Ours-FNL further improves the results by a large margin. Fig. 6.4 shows how our method refines the state over time. Visual comparison is shown in Fig. 6.3 for ground truth (Blue), Ours-FNL (Green) and SubCNN (Yellow).

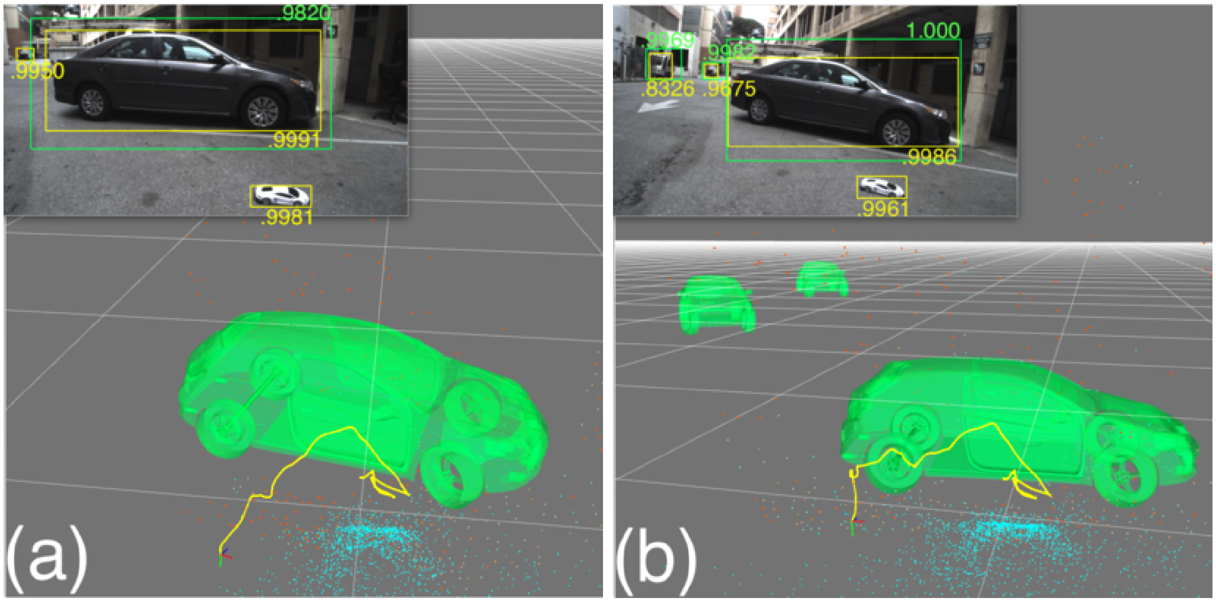


Figure 6.5: Class-specific scale prior. (a): A real car is detected by our system, unlike the toy car, despite both scoring high likelihood and therefore being detected by an image-based system (Yellow). As time goes by, the confidence on the real car increases (best viewed at 5×) (b). See online video at [DFS17].

6.5.2 Class-specific Priors

Objects have characteristic scales, which are lost in perspective projection but inferable with an inertial sensor. We impose a class-dependent prior on size and shape (*e.g.,* volume, aspect ratios). In Fig. 6.5, a toy car is detected as a car by an image-based detector (Yellow), but rejected by our system as inconsistent with the scale prior (Green). Fig. 6.5(b) shows two background cars in the far field, whose images are smaller than the toy car, yet they are detected correctly, whereas the toy car is rejected[4].

6.5.3 Occlusion and Memory

Our system represents objects in the state even while they are not visible, or detected by an image-based detector. This allows predicting the re-appearance of objects in future frames,

[4] All supplementary videos are available online at http://vision.cs.ucla.edu/vis.html.

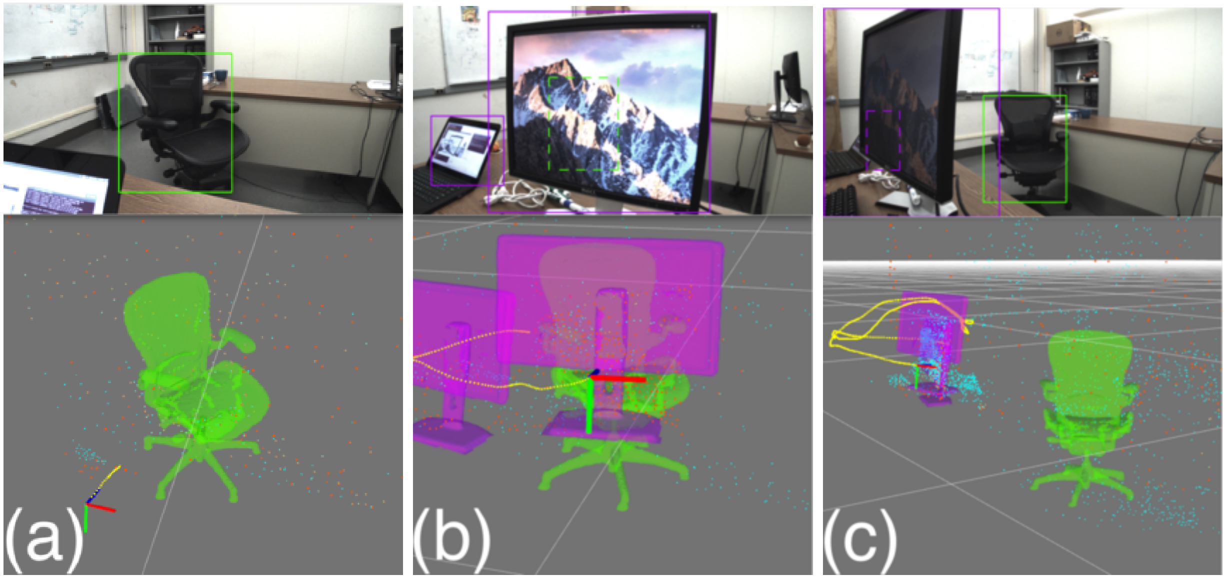


Figure 6.6: Occlusion management and short-term memory. (a): A chair is detected and later becomes occluded by the monitor (b). Its projection onto the image is shown in dashed lines, indicating occlusion. The model allows prediction of dis-occlusion (c) which allows resuming update when the chair comes back into view. See online video at [DFS17].

and to resume update if new evidences appear. Fig. 6.6 shows a chair first detected and then occluded by a monitor, later reappearing. The system predicts the chair to be completely occluded, and therefore does not use the image to update the chair, but resumes doing so when it reappears, by which time it is known to be the *same* chair that was previously seen (re-detection). In Sect. 6.5.4, we show the same phenomenon in a large-scale driving sequence.

6.5.4 Large-scale Driving Sequences

Fig. 6.1 and online video at [DFS17] show our results on a 3.7km-long sequence from KITTI. It contains hundreds of cars along the route. Once recognized as a car, we replace the bounding box with a CAD model of similar car, aligned with the pose estimate from the filter, in a manner similar to [SNS13], that however uses RGB-D data. In this sequence, we can also see cars on different streets "through walls" if they have been previously detected, which can help navigation.

6.5.5 Indoor Sequences

We have tested our system live in a public demo [DFK16], operating in real time in cluttered environments with people, chairs, tables, monitors and the like. Representative examples are shown for simpler scenes, for illustrative purposes, in Fig. 6.7, where again CAD models of objects are rendered once detected, a' la [SNS13]. Our system does not produce exact orientation estimates, as seen in Fig. 6.7, so there is plenty of room for improvement.

6.6 Discussion

Inertial sensors are in every modern phone, tablet, car, even many toys, all devices embedded in physical space and occasionally in need to interact with it. It makes sense to exploit inertials, along with visual sensors, to help detecting objects that exist in 3D physical space, and have characteristic shape and size, in addition to appearance. We have recorded tremendous progress in object detection in recent years, if by object one means a group of pixels in an image. Here we leverage such progress to design a detector that follows the prescriptions (a)-(e) indicated in the introduction.

We start by defining a representation as a minimal sufficient invariant statistic of object attributes, in line with [SC16]. We then marginalize on camera Euclidean pose – which allows us to enforce priors on the class-specific scale of objects – and update the measure by a Bayesian filter, where a CNN is in charge of computing the likelihood function.

We note that a minimal sufficient invariant for localization is an attributed point cloud, and therefore there is no need to deploy the machineries of Deep Learning to determine camera pose (Deep Learning could still be used to infer the attributes at points, which are used for correspondence). Instead, we use an Extended Kalman Filter, conditioned on which the update for object attributes can be performed by a Mixture-of-Kalman filter.

The result is a system whereby objects do not flicker in-and-out of existence, our confidence in their presence grows with accrued evidence, we know of their presence even if temporarily occluded, we can predict when they will be seen, and we can enforce known

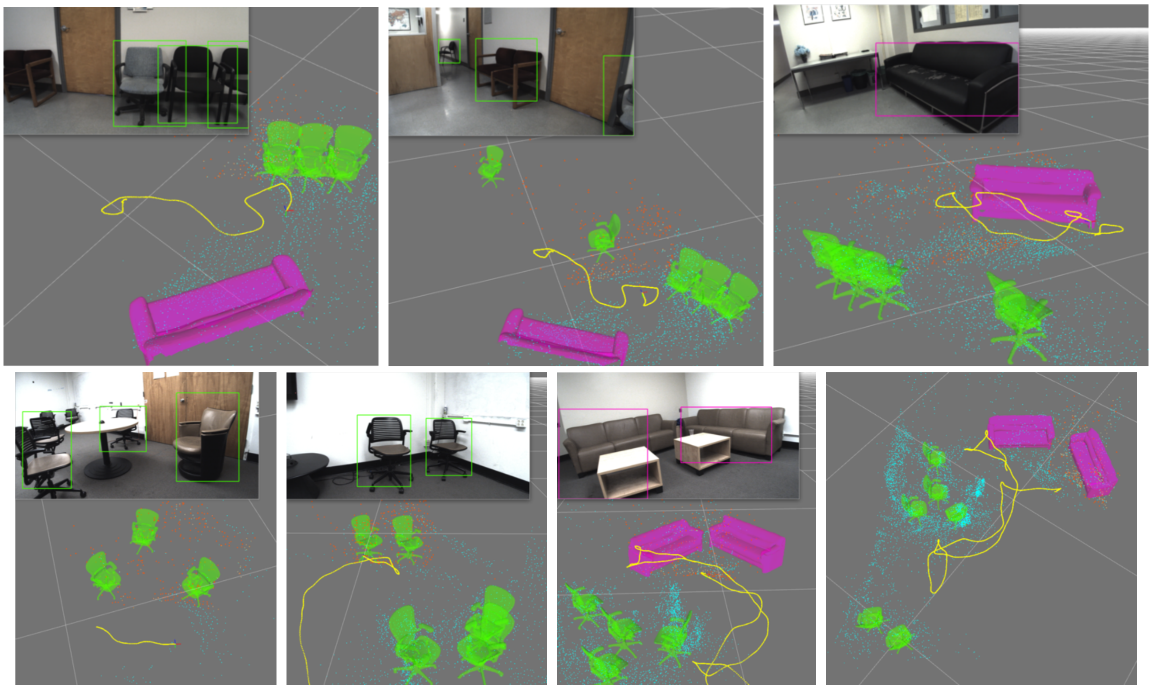


Figure 6.7: Indoor sequences. Top: An office area. Bottom: A Lounge area. Both videos are available at [DFS17].

scale priors to reject spurious hypotheses from the bottom-up proposal mechanism.

We have made stringent and admittedly restrictive assumptions in order to keep our model viable for real-time inference. One could certainly relax some of these assumptions and obtain more general models, but forgo the ability to operate in real time.

The main limitation of our system is its restriction to static objects. While in theory the framework is general, the geometry of moving and deforming objects is not represented, and therefore their attributes remain limited to what can be inferred in the image. Also, our representation of objects' shape is rather rudimentary, and as a result visibility computation rather fragile. These are all areas prime for further future development.

CHAPTER 7

Discussion

Building a meaningful representation is central to many computer vision, robotics and artificial intelligence tasks. It is inferred from the sensing data (*e.g.,* visual and possibly inertial measurements), and is used to answer queries related to a certain task of interest. What "meaningful" or "useful" means depends on the task. So does the definition of "nuisance" factors. In this dissertation, several optimal visual representations are constructed for various purposes.

In the local image descriptor design, we have constructed a descriptor that is a minimal sufficient statistic of the scene that is also maximally invariant to nuisance group actions on the data. Minimal sufficiency maintains all the information we would like to retain given the task of interest. Invariance to nuisances (such as translation, rotation) is the key to many widely-adopted local descriptors. We have established a theoretical link between the sampling theory and the design considerations adopted by these descriptors, and extended them to handle planar similarity given one single image is provided, and more general diffeomorphism when multiple views of the same scene are provided.

The same idea has been extended to image-level classifier to handle nuisance variables irrelevant to the identity of the objects depicted in each image. By coupling a conventional deep neural network architecture with explicit nuisance marginalization, we are able to achieve substantial improvements over the baseline network which is trained end-to-end to learn away nuisance variability. The difference between image classification and object detection is discussed. The discussion justified the use of marginalization and max-out operation in each task due to the fact that the same quantity (*e.g.,* translation, rotation, scale, aspect ratio, *etc.*) changes its role between nuisance and information in different tasks.

Finally, a real-time system to detect objects in three-dimensional space using video and inertial sensors are presented. A minimal sufficient representation, the posterior of semantic (identity) and syntactic (pose) attributes of objects in space is decomposed into a geometric term and a likelihood function. The former is maintained by a localization-and-mapping filter where the poses of the sensing platform over time are estimated and integrated out to yield an estimate of the object's geometry that is invariant to the view point. The likelihood function is approximated by a discriminatively-trained convolutional neural network, and is used to update the belief of the object semantics in space. The resulting system processes the video stream causally in real time, and provides a representation of objects in the scene that is persistent.

While we have focused on three different levels of granularity and thus different target applications where an optimal visual representation is estimated for each, the principles outlined in the dissertation are applicable to many other problems and tasks in the broad fields of computer vision, machine perception and artificial intelligence.

APPENDIX A

Appendix: Derivation of DSP-SIFT

A.1 Relation to Sampling Theory

This first section summarizes the background needed for the derivation, reported in the next section.

A.1.1 Sampling and aliasing

In this section we refer to a general scalar signal $f : \mathbb{R} \to \mathbb{R}; x \mapsto f(x)$, for instance the projection of the albedo of the scene onto a scanline. We define a *detector* to be a mechanism to select samples x_i, and a *descriptor* ϕ_i to be a statistic computed from the signal of interest and associated with the sample i. In the simplest case, x is regularly sampled, so the detector does not depend on the signal, and the descriptor is simply the value of the function at the sample $\phi_i = f(x_i)$. Other examples include:

A.1.1.1 Regular sampling (Shannon '49)

The detector is trivial: $\{x_i\} = \Lambda$ is a lattice, independent of f. The descriptor is a weighted average of f in a neighborhood of fixed size σ (possibly unbounded) around x_i: $\phi_i = \phi(\{f(x),\ x \in \mathcal{B}_\sigma(x_i)\})$. Neither the detector nor the descriptor function ϕ depend on f (although the *value* of the latter, of course, does).

If the signal was band-limited, Shannon's sampling theory would offer guarantees on the exact reconstruction $\hat{f}$ of $f(x), x \in \mathbb{R}$ from its sampled representation $\{x_i, \phi_i\}$. Unfortunately, the signals of interest are not band-limited (images are discontinuous), and therefore the reconstruction $\hat{f}$ can only approximate f. Typically, the approximation include "alien structures," *i.e.,* spurious extrema and discontinuities in $\hat{f}$ that do not exist in f. This phenomenon is known as *aliasing.* To reduce its effects, one can *replace* the original data f with another $\tilde{f}$ that is (closer to) band-limited and yet close to f, so that the samples can encode $\hat{f} = \tilde{f}$ free of aliasing artifacts. The conflicting requirements of faithful approximation of f and restriction on bandwidth trade off discriminative power (reconstruction error) with complexity, which is one of the goals of communications engineering. This tradeoff can be optimized by choice of *anti-aliasing operator*, that is the function that produces $\tilde{f}$ from f, usually via convolution with a low-pass filter. In our context, we seek for a tradeoff between discriminative power and *sensitivity to nuisance factors.* This will come naturally when anti-aliasing is performed with respect to the action of nuisance transformations.

A.1.1.2 Adaptive sampling (Landau '67)

The detector could be "adapted" to f by designing a functional ψ that selects samples $\{x_i\} = \psi(f)$. Typically, spatial frequencies of f modulate the length of the interval $\delta x_i \doteq x_{i+1} - x_i$. A special case of adaptive sampling that does not requires stationarity assumptions is described next. The descriptor may also depend on ψ, *e.g.,* by making the statistic depend on a neighborhood of variable size σ_i: $\phi_i = \phi(\{f(x),\ x \in \mathcal{B}_{\sigma_i}(x_i)\})$.

A.1.1.3 Tailored sampling (Logan '77)

For signals that are neither stationary nor band-limited, we can leverage on the violations of these assumptions to design a detector. For instance, if f contains discontinuities, the detector can place samples at discontinuous locations ("corners"). For band-limited signals, the detector can place samples at critical points (maxima, or "blobs", minima, saddles). A (location-scale) *co-variant detector* is a functional ψ whose zero-level sets

$$\psi(f; s, t) = 0 \tag{A.1}$$

define isolated (but typically multiple) samples of scales $s_i > 0$ and locations $t_i \in \mathbb{R}$ locally as a function of f via the implicit function theorem [GP74], in such a way that if f is trans-

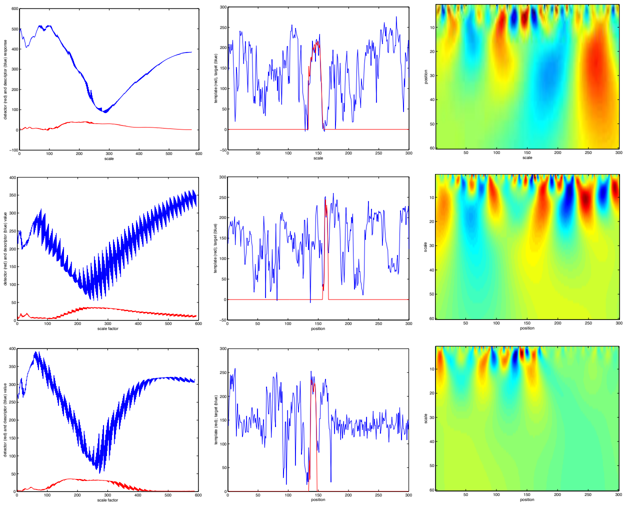


Figure A.1: Detector specificity vs. descriptor sensitivity. (Left) Change of detector response (red) as a function of scale, computed around the optimal location and scale (here corresponding to a value of 245), and corresponding change of descriptor value (blue). An ideal detector would have high specificity (sharp maximum around the true scale) and an ideal descriptor would have low sensitivity (broad minimum around the same). The opposite is true. This means that it is difficult to precisely select scale, and selection error results in large changes in the descriptor. Experiments are for the DoG detector and identity descriptor. Referring to the notation in Appendix (see details therein), (middle) template ρ (red) and target f (blue). (Right) corresponding scale-space $[f]$. Note that the maximum detector response may even not correspond to the true location. The jaggedness of the response is an aliasing artifact.

formed, for instance via a linear operator depending on location τ and scale σ parameters, $W(\sigma,\tau)f$, then so are the samples: $\psi(W(\sigma,\tau)f; s+\sigma, t+\tau) = 0$.

The associated descriptor can then be any function of the image in the reference frame defined by the samples t_i, s_i, the most trivial being the restriction of the original function f to the neighborhood $\mathcal{B}_{s_i}(t_i)$. This, however, does not reduce the dimensionality of the representation. Other descriptors can compute statistics of the signal in the neighborhood, or on the entire line. Note that descriptors ϕ_i could have different dimensions for each i.

A.1.1.4 Anti-aliasing and “pooling”

In classical sampling theory, anti-aliasing refers to low-pass filtering or smoothing that *typically*[1] does not cause *genetic phenomena* (spurious extrema, or *aliases*, appearing in the reconstruction of the smoothed signal.) Of course, anti-aliasing typically has *destructive* effects, in the sense of eliminating extrema that are instead present in the original signal.

A side-effect of anti-aliasing, which has implications when the goal is *not* to reconstruct, but to detect or localize a signal, is to reduce the sensitivity of the relevant variable (descriptor) to variations of the samples (detector). If we sample translations, $x_i = x + t_i$, and just store $f_i = f(x_i)$, an arbitrarily small translation of the sample dx can cause an arbitrarily large variation in the representation $\delta f(x_i) = f(x_i + dx) - f_i$, when x_i is a discontinuity. So, the sensitivity $S(f) = \frac{\delta f}{dx} = \infty$. An anti-aliasing operator $\phi(f)$ should reduce sensitivity to translation: $\frac{\delta \phi(f)}{dx} \ll \frac{\delta f}{dx}$. Of course, this could be trivially achieved by choosing $\phi(f) = 0$ for any f. The goal is to trade off sensitivity with *discriminative power.* For the case of translation, this tradeoff has been described in [BM11]. However, similar considerations holds for scale and domain-size sampling.

A.2 Derivation

The derivation of DSP-SIFT and its extensions follows a series of steps summarized as follows:

- We start from the correspondence, or matching, task: Classify a given datum f (test image, or target) as coming from one of M model classes, each represented by an image ρ_j (training images, or templates), with $j = 1, \dots, M$.
- Both training and testing data are affected by nuisance variability due to changes of (i) illumination (ii) vantage point and (iii) partial occlusion. The former is approximated by local contrast transformations (monotonic continuous changes of intensity values), a maximal invariant to which is the gradient orientation. Vantage point changes are

[1] This central tenet of scale-space theory only holds for scalar signals. Nevertheless, genetic effects have been shown to be *rare* in two-dimensional Gaussian scale-space [CE11].

decomposed as a translation parallel to the image plane, approximated by a planar translation of the image, and a translation orthogonal to it, approximated by a scaling of the image. Partial occlusions determine the shape of corresponding regions in training and test images, which are approximated by a given shape (say a circle, or square) of unknown size (scale). These are very crude approximations but nevertheless implicit to most local descriptors. In particular, camera rotations are not addressed in this work.

- Solving the (local) correspondence problem amounts to an $M+1$-hypothesis testing problem, including the background class. Nuisance (i) is eliminated at the outset by considering gradient orientation instead of image intensity. Dealing with nuisances (ii)–(iii) requires searching across all (continuous) translations, scales, and domain sizes.
- The resulting matching function must be discretized for implementation purposes. Since the matching cost is quadratic in the number of samples, sampling should be reduced to a minimum, which in general introduces artifacts ("aliasing").
- Anti-aliasing operators can be used to reduce the effects of aliasing artifacts. For the case of (approximations of) the likelihood function, such as SIFT, anti-aliasing corresponds to marginalizing residual nuisance transformations, which in turn corresponds to pooling gradient orientations across different locations, scales and domain sizes.
- The samples can be thought of as a special case of "deformation hypercolumns" [Soa10] (samples with respect to the orientation group) with the addition of the size-space semi-group (Fig. 2.9). Most importantly, the samples along the group are anti-aliased, to reduce the effects of structural perturbations.

A.3 Formalization

For simplicity, we formalize the matching problem for a scalar image (a scanline), and neglect contrast changes for now, focusing on the location-scale group and domain size instead.

Let $\rho_j : \mathbb{R} \to \mathbb{R}$, with $j = 1, \dots, M$ possible models (templates, or ideal training images). The data (test image) is $f : [0, \dots, N] \to \mathbb{R}$ with each sample $f(x_i)$ obtained from one of the ρ_j via translation by $\tau \in \mathbb{R}$, scaling by $\sigma > 0$, and sampling with interval ϵ, *if* x_i *is in the visible domain* $[a,\ b]$. Otherwise, the scene ρ_j is occluded and $f(x_i)$ has nothing to do with it.

The forward model that, given ρ and all nuisance factors σ, τ, a, b, generates the data, is indicated as follows: If $x_i \in [a,\ b]$, then

$$f(x_i) = W_\epsilon(x_i; \sigma, \tau)\rho_j + n_{ij} \tag{A.2}$$

where n_i is a sample of a white, zero-mean Gaussian random variable with variance κ. Otherwise, $x_i \notin [a,\ b]$, and $f(x_i) = \beta(x_i)$ is a realization of a process independent of ρ_j (the "background"). The operator W_ϵ is linear[2] and given by

$$W_\epsilon(x_i; \sigma, \tau)\rho \doteq \int_{\mathcal{B}_\epsilon(x_i)} \rho\left(\frac{x-\tau}{\sigma}\right) dx \tag{A.6}$$

where $\mathcal{B}_\epsilon(x_i)$ is a region corresponding to a pixel centered at x_i. Matching then amount to a hypothesis testing problem on whether a given measured $f = \{f(x_i)\}_{i=1}^N$ is generated by any of the ρ_j – under suitable choice of nuisance parameters – or otherwise is just labeled as background:

$$\begin{aligned} H_0 :\ \exists\ j, a, b, \sigma, \tau \mid p(f(x_i)|\rho_j, a, b, \sigma, \tau) = & \\ p_\beta(\{f(x_k),\ x_k \notin [a,\ b]\}) \prod_{x_i \in [a,\ b]} \mathcal{N}(f(x_i) - W_\epsilon(x_i; \sigma, \tau)\rho_j), \kappa) & \quad \text{(A.7)} \end{aligned}$$

[2] $W : \mathbb{L}^2(\mathbb{R}) \ \to \mathbb{R}^N$ can be written as an integral on the real line using the characteristic function $\chi_{\mathcal{B}_\epsilon}(x - x_i)$ or a more general sampling kernel $k_\epsilon(x - x_i)$, for instance a Gaussian with zero-mean and standard deviation ϵ. Then we have

$$\int_{\mathcal{B}_\epsilon(x_i)} \rho\left(\frac{x-\tau}{\sigma}\right) dx = \int k_\epsilon(x - x_i)\rho\left(\frac{x-\tau}{\sigma}\right) dx = \iint \delta\left(y - \frac{x-\tau}{\sigma}\right) k_\epsilon(x - x_i)\rho\,(y)\,dxdy \tag{A.3}$$

$$= \iint \delta\left(y + \frac{\tau}{\sigma} - \frac{x}{\sigma}\right) k_\epsilon(x - x_i)dx\rho\,(y)\,dy = \iint \delta\left(y + \frac{\tau}{\sigma} - \bar{x}\right) k_\epsilon(\sigma\bar{x} - x_i)\sigma d\bar{x}\rho\,(y)\,dy \tag{A.4}$$

$$= \sigma \int k_\epsilon(\sigma y + \tau - x_i)\rho\,(y)\,dy \tag{A.5}$$

and the alternate hypothesis is simply $p_\beta(\{f(x_i)\}_{i=1}^N)$. If the background density p_β is unknown, the likelihood ratio test reduces to the comparison of the product on the right-hand side to a threshold, typically tuned to the ratio with the second-best match (although some recent work using extreme-value theory improves this [FSR13]). In any case, the log-likelihood for points in the interval $x_i \in [a,\ b]$ can be written as

$$r_{ij}(a, b, \sigma, \tau) = \frac{1}{|b-a|} \sum_{x_i \in [a,b]} |f(x_i) - W_\epsilon(x_i; \sigma, \tau)\rho_j| \tag{A.8}$$

which will have to be minimized for all pixels $i = 1, \dots, N$ and templates $j = 1, \dots, M$, of which there is a finite number. However, it also has to be minimized over the continuous variables a, b, σ, τ. Since r is in general neither convex nor smooth as a function of these parameters, analytical solutions are not possible. Discretizing these variables is necessary,[3] and since the minimization amounts to a search in $2+4$ dimensions, we seek for methods to *reduce the number of samples with respect to the arguments* a, b, σ, τ as much as possible.

There are many ways to sample, some described in Sect. A.1.1, so several questions are in order: (a) How should each variable be sampled? Regularly or adaptively? (b) If sampled regularly, when do aliasing phenomena occur? Can anti-aliasing be performed to reduce their effects? (c) The search is jointly over a, b and σ, τ, and given one pair, it is easy to optimize over the other. Can these two be "separated"? (d) Is it possible to quantify and optimize the tradeoff between the number of samples and classification performance? Or for a given number of samples develop the "best" anti-aliasing ("descriptor")? (e) For a histogram descriptor, how is "anti-aliasing" accomplished?

A.4 Common approaches and their rationale

Concerning question (a) above, most approaches in the literature perform *tailored sampling* (Sect. A.1.1.3) of both τ and σ, by deploying a location-scale covariant detector [Low04]. When time is not a factor, it is common to forgo the detector and compute descriptors

[3] Coarse-to-fine, homotopy-based methods or jump-diffusion processes can alleviate, but not remove, this burden.

"densely" (a misnomer) by regularly subsampling the image lattice, or possibly undersampling by a fixed "stride." Sometimes, scale is also regularly sampled, typically at far coarser granularity than the scale-space used for scale selection, for obvious computational reasons. In general, regular sampling requires assumptions on band limits. The function $W\rho$ is *not* band-limited as a function of τ. Therefore, *tailored sampling* (detector/descriptor) is best suited for the translation group.[4] We will therefore assume that τ has been tailor-sampled (detected, or canonized), but only up to a localization error. Without loss of generality we assume the sample is centered at zero, and the residual translation τ is in the neighborhood of the origin. In Fig. A.1 we show that the sensitivity to scale of a common detector (DoG), which should be high, and is instead lower than the sensitivity of the resulting descriptor, which should be low. Therefore, small changes in scale cause large changes in scale sample localization, which in turn cause large changes in the value of the descriptor. Therefore, we forgo scale selection, and instead finely sample scale. This causes complexity issues, which prompt the need to sub-sample, and correspondingly to anti-alias or aggregate across scale samples. Alternatively, as done in Sect. 5.2, we can have a coarse adaptive or tailored sampling of scales, and then perform fine-scale sampling and anti-aliasing around the (multiple) selected scales.

Concerning (b), anti-aliasing phenomena appear as soon as Nyquist's conditions are violated, which is almost always the case for scale and domain-size (Fig. A.2). While most practitioners are reluctant to down-sample spatially, leaving millions of locations to test, it is rare for anyone to employ more than a few tens of scales, corresponding to a wild down-sampling of scale-space. This is true *a fortiori* for domain-size, where the domain size is often fixed, say to 69×69 or 91×91 locations [FDB14]. And yet, spatial anti-aliasing is routinely performed in most descriptors, whereas none – to the best of our knowledge – perform scale or domain-size anti-aliasing. Anti-aliasing should ideally decrease the sensitivity of the descriptor, without excessive loss of discriminative power. This is illustrated in Fig. A.2.

[4] Purported superiority of "dense SIFT" (regularly sampled at thousands of location) compared to ordinary SIFT (at tens or hundreds of detected location), as reported in few empirical studies, is misleading as comparison has to be performed for a comparable number of samples.

For (c), we make the choice of fixing the domain size in the target (test) image, and regularly sampling scale and domain-size, re-mapping each to the domain size of the target (Fig. 2.1). For comparison with [FDB14], we choose this to be 69 × 69. While the choice of fixing one of the two domains entails a loss, it can be justified as follows: Clearly, the hypothesis cannot be tested independently on each datum $f(x_i)$. However, testing on any *subset* of the "true inlier set" $[a,\ b]$ reduces the *power*, but not the validity, of the test. Vice-versa, using a "superset" that includes outliers invalidates the test. However, a small percentage of outliers can be managed by considering a robust (Huber) norm $\|f - W\rho\|_{\mathcal{H}}$ instead of the $\mathbb{L}^2$ norm. Therefore, one could consider the sequential hypothesis testing problem, starting from each $x_i \in [a = b]$ as an hypothesis, then "growing" the region by one sample, and repeating the test. Note that the optimization has to be solved at each step.[5] As a first-order approximation, one can *fix* the interval $[a,\ b]$ and accept a less powerful test (if that is a subset of the actual domain) or a test corrupted by outliers (if it is a superset). This is, in fact, done in most local feature-based registration or correspondence methods, and even in region-based segmentation of textures, where statistics must be pooled in a region.

While (d) is largely an open question, (e) follows directly from classical sampling considerations, as described in Sect. A.1.1.

A.5 Anti-aliasing descriptors

In the case of matching images under nuisance variability, it has been shown [DKD15] that the *ideal descriptor* computed at a location x_i is not a vector, but a function that approximates the likelihood, where the nuisances are marginalized. In practice the descriptor is approximated with a regularized histogram, similar to SIFT (2.1). In this case, anti-aliasing corresponds to a weighted average across different locations, scales and *domain sizes*. But

[5] In this interpretation, the test can be thought of as a setpoint change detection problem. Another interpretation is that of (binary) region-based segmentation, where one wishes to classify the *range* of a function $f - W\rho$ into two classes, with values coming from either ρ or the background, but the thresholds is placed on the *domain* of the function $[a,\ b]$. Of course, the statistics used for the classification depend on a, b so this has to be solved as an alternating minimization, but it is a convex one [CE05].

the averaging in this case is simply accomplished by pooling the histogram across different locations *and* domain-sizes, as in (2.2). The weight function can be design to optimize the tradeoff between sensitivity and discrimination, although in Sect. 5.2 we use a simple uniform weight.

To see how pooling can be interpreted as a form of generalized anti-aliasing, consider the function f sampled on a discretized domain $f(x_i)$ and a neighborhood $\mathcal{B}_\sigma(x_i)$ (for instance the sampling interval). The pooled histogram is

$$p_{x_i}(y) = \frac{1}{\sigma} \sum_{x_j \in \mathcal{B}_\sigma(x_i)} \delta(y - f(x_j)) \tag{A.9}$$

whereas the anti-aliased signal (for instance with respect to the pillbox kernel) is

$$\phi(x_i) = \frac{1}{\sigma} \sum_{x_j \in \mathcal{B}_\sigma(x_i)} f(x_j) \tag{A.10}$$

The latter can be obtained as the mean of the former

$$\phi(x_i) = \sum_y y p_{x_i}(y) \tag{A.11}$$

although former can be used for purposes other than computing the mean (which is the best estimate under Gaussian (ℓ^2) uncertainty), for instance to compute the median (corresponding to the best estimate under uncertainty measured by the ℓ^1 norm), or the mode:

$$\hat{f}(x_i) = \arg\max_y p_{x_i}(y). \tag{A.12}$$

The approximation is accurate only to the extent in which the underlying distribution $p_x(y) = p(f(x) = y)$ is stationary and ergodic (so the spatially pooled histogram approaches the density), but otherwise it is still a generalization of the weighted average or mean.

This derivation also points the way to how a descriptor can be used to synthesize images: Simply by sampling the descriptor, thought of as a density for a given class [DKD15, VKM13]. It also suggests how descriptors can be compared: Rather than computing descriptors in both training and test images, a test datum can just be fed to the descriptor, to yield the likelihood of a given model class [FMR08], without computing the descriptor in the test image.

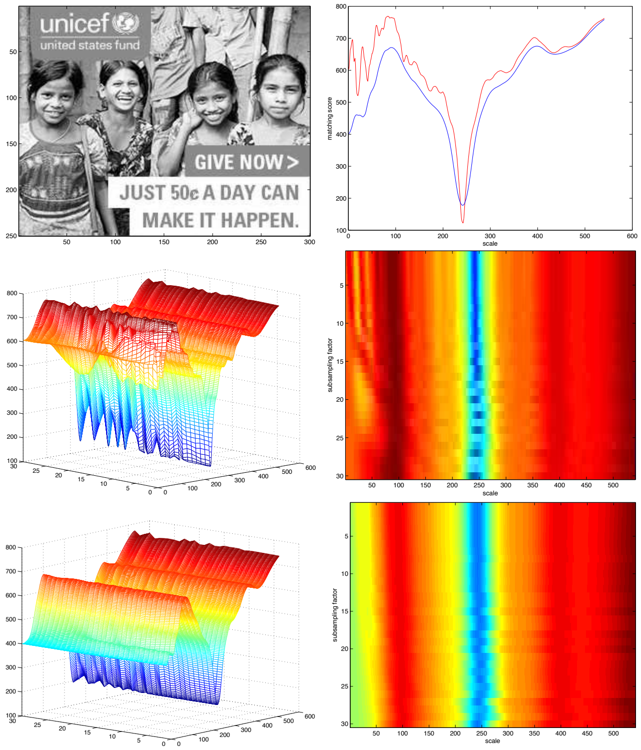


Figure A.2: Aliasing: (Top left) A random row is selected as the target f and re-scaled to yield the orbit $[f]$; a subset of f, cropped, re-scaled, and perturbed with noise, is chosen as the template ρ. The distance E between ρ and $[f]$ is shown in red (right) as a function of scale. The same exercise is repeated for different sub-sampling of $[f]$, and rescaled for display either as a mesh (middle left) or heat map (right) that clearly show aliasing artifacts along the optimal ridge. Anti-aliasing scale (bottom) produces a cleaner ridge (left, right). The net effect of anti-aliasing has been to smooth the matching score E (top-right, in blue) but without computing it on a fine grid. Note that the valley of the minimum is broader, denoting decreased sensitivity to scale, and the value is somewhat higher, denoting a decreased discriminative power and risk of aliasing if the value raises above that of other local minima.

REFERENCES

[ABS16] U. Asif, M. Bennamoun, and F. Sohel. “Simultaneous dense scene reconstruction and object labeling.” In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[ADF12] B. Alexe, T. Deselaers, and V. Ferrari. “Measuring the objectness of image windows.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.

[AGL93] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel. “Axioms and fundamental equations of image processing.” *Archive for Rational Mechanics and Analysis*, **123**(3):199–257, 1993.

[AOV17] A. Alahi, R. Ortiz, and P. Vandergheynst. “Freak: Fast retina keypoint.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[ARP15] F. Anselmi, L. Rosasco, and T. Poggio. “On Invariance and Selectivity in Representation Learning.” *arXiv preprint arXiv:1503.05938*, 2015.

[AYB15] S. Aditya, Y. Yang, C. Baral, C. Fermuller, and Y. Aloimonos. “Visual commonsense for scene understanding using perception, semantic parsing and reasoning.” *AAAI Spring Symposium Series*, 2015.

[AZD14] N. Atanasov, M. Zhu, K. Daniilidis, and G. Pappas. “Semantic localization via the matrix permanent.” *Robotics: Science and Systems*, 2014.

[Bah54] R. R. Bahadur. “Sufficiency and statistical decision functions.” *Annals of Mathematical Statistics*, **25**(3):423–462, 1954.

[BGC15] L. Baraldi, C. Grana, and R. Cucchiara. “Scene segmentation using temporal clustering for accessing and re-using broadcast video.” In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2015.

[BM01] A. Berg and J. Malik. “Geometric blur for template matching.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[BM11] J. Bruna and S. Mallat. “Classification with scattering operators.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[BM13] J. Bruna and S. Mallat. “Invariant scattering convolution networks.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013.

[BPL10] Y. L. Boureau, J. Ponce, and Y. LeCun. “A theoretical analysis of feature pooling in visual recognition.” In *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

[BPS14] J. Balzer, M. Peters, and S. Soatto. “Volumetric reconstruction applied to perceptual studies of size and weight.” In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2014.

[BPT14] H. Bilen, M. Pedersoli, and T. Tuytelaars. “Weakly supervised object detection with posterior regularization.” In *Proceedings of British Machine Vision Conference*, 2014.

[BRP09] J. V. Bouvrie, L. Rosasco, and T. Poggio. “On invariance in hierarchical models.” In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[BS13] J. Balzer and S. Soatto. “CLAM: Coupled localization and mapping with efficient outlier handling.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[BS15] D. Banica and C. Sminchisescu. “Second-order constrained parametric proposals and sequential search-based structured prediction for semantic segmentation in rgb-d images.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[BSF08] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. “Segmentation and recognition using structure from motion point clouds.” In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[BTG06] H. Bay, T. Tuytelaars, and L. V. Gool. “Surf: Speeded up robust features.” In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2006.

[BVR16] M. Bláha, C. Vogel, A. Richard, J. D. Wegner, T. Pock, and K. Schindler. “Large-scale semantic 3d reconstruction: an adaptive multi-resolution model for multi-class volumetric labeling.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[CE05] T. Chan and S. Esedoglu. “Aspects of Total Variation Regularized L1 Function Approximation.” *SIAM Journal on Applied Mathematics*, **65**(5):1817–1837, 2005.

[CE11] C. Chen and H. Edelsbrunner. “Diffusion runs low on persistence fast.” In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[CFN14] C. Couprie, C. Farabet, L. Najman, and Y. Lecun. “Convolutional nets and watershed cuts for real-time semantic labeling of rgbd videos.” *The Journal of Machine Learning Research*, 2014.

[CK13] C. Cadena and J. Košecka. “Semantic parsing for priming object detection in rgb-d scenes.” *Workshop on Semantic Perception, Mapping and Exploration*, 2013.

[CKZ15] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. “3d object proposals for accurate object class detection.” In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[CKZ16] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. "Monocular 3d object detection for autonomous driving." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[CLC08] N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. "3d urban scene modeling integrating recognition and reconstruction." *International Journal of Computer Vision*, 2008.

[CLS10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. "Brief: Binary robust independent elementary features." In *Proceedings of European Conference on Computer Vision*. Springer, 2010.

[CLT10] M. Choi, J. Lim, A. Torralba, and A. Willsky. "Exploiting hierarchical context on a large database of object categories." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, 2010.

[CLV11] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. "The devil is in the details: an evaluation of recent feature encoding methods." In *Proceedings of British Machine Vision Conference*, 2011.

[COR16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. "The cityscapes dataset for semantic urban scene understanding." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, 2016.

[CRU16] F. Chhaya, D. Reddy, S. Upadhyay, V. Chari, M. Z. Zia, and K. M. Krishna. "Monocular reconstruction of vehicles: Combining slam with shape priors." In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2016.

[CS12] J. Carreira and C. Sminchisescu. "CPMC: Automatic object segmentation using constrained parametric min-cuts." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2012.

[CSV14] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. "Return of the devil in the details: Delving deep into convolutional nets." In *Proceedings of the British Machine Vision Conference*, 2014.

[CT12] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 2012.

[CTC09] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod. "Chog: Compressed histogram of gradients a low bit-rate feature descriptor." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[CW14] T. Cohen and M. Welling. "Learning the irreducible representations of commutative lie groups." In *Proceedings of the International Conference on Machine Learning*, 2014.

[CZL14] M. Cheng, Z. Zhang, W. Lin, and P. Torr. "BING: Binarized normed gradients for objectness estimation at 300fps." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[DBS13] D. Davis, J. Balzer, and S. Soatto. "Asymmetric sparse kernel approximations for nearest neighbor search." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[DDS09] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F.-F. Li. "ImageNet: A large-scale hierarchical image database." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[DFK16] J. Dong, X. Fei, N. Karianakis, K. Tsotsos, and S. Soatto. "VL-SLAM: Real-Time Visual-Inertial Navigation and Semantic Mapping." *The IEEE Conference on Computer Vision and Pattern Recognition, Live Demo*, 2016.

[DFS17] J. Dong, X. Fei, and S. Soatto. "Visual-Inertial-Semantic Scene Representation for 3D Object Detection." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Supplementary Videos at http://vision.cs.ucla.edu/vis.html*, 2017.

[DHG15] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. "Long-term recurrent convolutional networks for visual recognition and description." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[DKD15] J. Dong, N. Karianakis, D. Davis, J. Hernandez, J. Balzer, and S. Soatto. "Multi-view feature engineering and learning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[DNO07] E. Delponte, N. Noceti, F. Odone, and A. Verri. "The importance of continuous views for real-time 3d object recognition." *ICCV Workshop on 3D Representation for Recognition*, 2007.

[DS15] J. Dong and S. Soatto. "Domain-size pooling in local descriptors: DSP-SIFT." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[DSR14] A. Dosovitskiy, J. Springenberg, M. Riedmiller, and T. Brox. "Unsupervised feature learning by augmenting single images." In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[DT05] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[DTL15] Z. Deng, S. Todorovic, and L. Latecki. "Semantic segmentation of rgbd images with mutex constraints." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[EGW10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. "The pascal visual object classes (voc) challenge." *International Journal of Computer Vision*, 2010.

[EST14] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. "Scalable object detection using deep neural networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[FCN12] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. "Scene parsing with multiscale feature learning, purity trees, and optimal covers." *ArXiv preprint:1202.2160*, 2012.

[FDB14] P. Fischer, A. Dosovitskiy, and T. Brox. "Descriptor matching with convolutional neural networks: a comparison to sift." *ArXiv preprint:1405.5769*, 2014.

[FDU12] S. Fidler, S. Dickinson, and R. Urtasun. "3d object detection and viewpoint estimation with a deformable 3d cuboid model." In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[FFP04] L. Fei-Fei, R. Fergus, and P. Perona. "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories." *CVPR Workshop on Generative-Model Based Vision*, 2004.

[FHG15] D. F. Fouhey, W. Hussain, A. Gupta, and M. Hebert. "Single image 3d without a single 3d image." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[FK96] O. D. Faugeras and R. Keriven. "Variational principles, surface evolution, PDE's, level set methods and the stereo problem." *INRIA TR*, 1996.

[FMR08] P. Felzenszwalb, D. McAllester, and D. Ramanan. "A discriminatively trained, multiscale, deformable part model." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[FSR13] V. Fragoso, P. Sen, S. Rodriguez, and M. Turk. "Evsac: Accelerating hypotheses generation by modeling matching scores with extreme value theory." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[GAM13] S. Gupta, P. Arbelaez, and J. Malik. "Perceptual organization and recognition of indoor scenes from rgb-d images." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[GB05] M. Grabner and H. Bischof. "Object recognition based on local feature trajectories." *I cognitive vision works*, 2005.

[GBS15] G. Graber, J. Balzer, S. Soatto, and T. Pock. "Efficient minimal surface regularization of perspective depth maps in variational stereo." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[GDD14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[Gir15] R. Girshick. “Fast R-CNN.” In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[GLL09] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng. “Measuring invariances in deep networks.” In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[GLS13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. “Vision meets robotics: The kitti dataset.” *International Journal of Robotics Research*, 2013.

[GP74] V. Guillemin and A. Pollack. *Differential Topology.* Prentice-Hall, 1974.

[GSS15] I. Goodfellow, J. Shlens, and C. Szegedy. “Explaining and Harnessing Adversarial Examples.” In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[GWG14] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. “Multi-scale orderless pooling of deep convolutional activation features.” *ArXiv preprint:1403.1840*, 2014.

[HBD15] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. “What makes for effective detection proposals?” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2015.

[HFL14] A. Hermans, G. Floros, and B. Leibe. “Dense 3d semantic mapping of indoor scenes from rgb-d images.” In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[HHX15] D. Hoiem, J. Hays, J. Xiao, and A. Khosla. “Guest editorial: Scene understanding.” *International Journal of Computer Vision*, 2015.

[HKB13] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis. “Towards consistent vision-aided inertial navigation.” *Algorithmic Foundations of Robotics X*, 2013.

[HLB11] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck. “Temporal pooling and multiscale learning for automatic annotation and ranking of music audio.” In *Proceedings of the International Society of Music Information Retrieval*, 2011.

[HLR14] A. Humayun, F. Li, and J. M. Rehg. “RIGOR: Reusing inference in graph cuts for generating object regions.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[HMZ12] V. Hassne, T.and Mayzels, and L. Zelnik-Manor. “On sifts and their scales.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[HZC13] C. Hane, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. "Joint 3D scene reconstruction and class segmentation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[HZR14] K. He, X. Zhang, S. Ren, and J. Sun. "Spatial pyramid pooling in deep convolutional networks for visual recognition." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[HZR15] K. He, X. Zhang, S. Ren, and J. Sun. "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[ISS16] H. Izadinia, Q. Shan, and S. M. Seitz. "IM2CAD." *arXiv preprint arXiv:1608.05137*, 2016.

[JHD12] Y. Jia, C. Huang, and T. Darrell. "Beyond spatial pyramids: Receptive field learning for pooled image features." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[JS11] E. Jones and S. Soatto. "Visual-inertial navigation, localization and mapping: a scalable real-time large-scale approach." *International Journal of Robotics Research*, 2011.

[KAJ11] H. Koppula, A. Anand, T. Joachims, and A. Saxena. "Semantic labeling of 3d point clouds for indoor scenes." In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[KD84] J. J. Koenderink and A. J. van Doorn. "The structure of images." *Biological Cybernetics*, **50**(5):363–370, 1984.

[KK14] P. Krähenbühl and V. Koltun. "Geodesic object proposals." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[KKS13] B. Kim, P. Kohli, and S. Savarese. "3d scene understanding by voxel-crf." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[KLD14] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. "Joint semantic segmentation and 3d reconstruction from monocular video." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[KMF13] A. Karpathy, S. Miller, and L. Fei-Fei. "Object discovery in 3d scenes via shape analysis." In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[KMT16] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab. "Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[Kol11] V. Koltun. “Efficient inference in fully connected crfs with gaussian edge potentials.” In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “Imagenet classification with deep convolutional neural networks.” In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[KSJ14] A. Kanazawa, A. Sharma, and D. Jacobs. “Locally scale-invariant convolutional neural networks.” *arXiv preprint arXiv:1412.5104*, 2014.

[KTS14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. “Large-scale video classification with convolutional neural networks.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[KWB14] F. S. Khan, J. van de Weijer, A. D. Bagdanov, and M. Felsberg. “Scale coding bag-of-words for action recognition.” In *Proceedings of the IEEE International Conference on Pattern Recognition*, 2014.

[LAE16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. “Ssd: Single shot multibox detector.” In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[LBR12] K. Lai, L. Bo, X. Ren, and D. Fox. “Detection-based object labeling in 3d scenes.” In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[LCS11] S. Leutenegger, M. Chli, and R. Y. Siegwart. “BRISK: Binary robust invariant scalable keypoints.” In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[LeC12] Y. LeCun. “Learning invariant feature hierarchies.” In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2012.

[LFU13] D. Lin, S. Fidler, and R. Urtasun. “Holistic scene understanding for 3d object detection with rgbd cameras.” In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[LHB04] Y. LeCun, F. J. Huang, and L. Bottou. “Learning methods for generic object recognition with invariance to pose and lighting.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[Lin98] T. Lindeberg. “Principles for automatic scale selection.” *Technical Report, KTH, Stockholm, CVAP*, 1998.

[LK81] B. D. Lucas and T. Kanade. “An iterative image registration technique with an application to stereo vision.” In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 2, pp. 674–679, 1981.

[LLF05] V. Lepetit, P. Lagger, and P. Fua. "Randomized trees for real-time keypoint recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[LM14] M. Li and A. Mourikis. "Online temporal calibration for camera–imu systems: Theory and algorithms." *International Journal of Robotics Research*, 2014.

[Low04] D. G. Lowe. "Distinctive image features from scale-invariant keypoints." *International Journal of Computer Vision*, **2**(60):91–110, 2004.

[LS11a] T. Lee and S. Soatto. "Learning and matching multiscale template descriptors for real-time detection, localization and tracking." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[LS11b] T. Lee and S. Soatto. "Video-based descriptors for object recognition." *Image and Vision Computing*, **29**(10):639–652, 2011.

[LSH16] G. Lin, C. Shen, A. Hengel, and I. Reid. "Exploring context with deep structured models for semantic segmentation." *arXiv preprint arXiv:1603.03183*, 2016.

[LXG15] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. "Deeply-supervised nets." In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2015.

[LZD16] S. Leonardos, X. Zhou, and K. Daniilidis. "Distributed consistent data association." *arXiv preprint arXiv:1609.07015*, 2016.

[MCL14] X. Mottaghi, R.and Chen, X. Liu, N. Cho, S. Lee, S. Fidler, R. Urtasun, and A. Yuille. "The role of context for object detection and semantic segmentation in the wild." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[MCM02] J. Matas, O. Chum, M.Urban, and T. Pajdla. "Robust wide baseline stereo from maximally stable extremal regions." In *Proceedings of the British Machine Vision Conference (BMVC)*, 2002.

[MCU04] J. Matas, O. Chum, M. Urban, and T. Pajdla. "Robust wide-baseline stereo from maximally stable extremal regions." *Image and Vision Computing*, 2004.

[Mem13] R. Memisevic. "Learning to relate images." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **35**(8):1829–1846, 2013.

[MGG13] S. Manen, M. Guillaumin, and L. V. Gool. "Prime object proposals with randomized Prim's algorithm." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[MHD16] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. "Semantic fusion: Dense 3d semantic mapping with convolutional neural networks." *arXiv preprint arXiv:1609.05130*, 2016.

[MMT15] R. Mur-Artal, J. Montiel, and J. D. Tardós. "Orb-slam: a versatile and accurate monocular slam system." In *Proceedings of the IEEE Transactions on Robotics*, 2015.

[MP07] P. Moreels and P. Perona. "Evaluation of features detectors and descriptors based on 3d objects." *International Journal of Computer Vision*, **73**(3):263–284, 2007.

[MS05] K. Mikolajczyk and C. Schmid. "A performance evaluation of local descriptors." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 1615–1630, 2005.

[MSK03] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An invitation to 3-D vision, from images to geometric models*. Springer Verlag, 2003.

[MTS04] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. "A comparison of affine region detectors." *International Journal of Computer Vision*, **1**(60):63–86, 2004.

[MTS05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. "A comparison of affine region detectors." *International Journal of Computer Vision*, 2005.

[MV15] Aravindh Mahendran and Andrea Vedaldi. "Understanding deep image representations by inverting them." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[NYC15] A. Nguyen, J. Yosinski, and J. Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[PL15] S. Pillai and J. Leonard. "Monocular slam supported object recognition." *Robotics: Science and Systems*, 2015.

[PML11] T. Poggio, J. Mutch, J. Leibo, L. Rosasco, and A. Tacchetti. "The computational magic of the ventral stream: sketch of a theory (and why some deep architectures work).", 2011.

[QT09] A. Quattoni and A. Torralba. "Recognizing indoor scenes." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[RD06] E. Rosten and T. Drummond. "Machine learning for high-speed corner detection." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[RDG16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[RDS15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge." *International Journal of Computer Vision*, 2015.

[RHB07] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. "Unsupervised learning of invariant feature hierarchies with applications to object recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[RHG15] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks." In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[RKB11] E. Rahtu, J. Kannala, and M. Blaschko. "Learning a category independent object detection cascade." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[RRK11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: an efficient alternative to SIFT or SURF." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[RS15] Z. Ren and E. B. Sudderth. "Three-dimensional object detection and layout prediction using clouds of oriented gradients." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[SC14] S. Soatto and A. Chiuso. "Visual scene representations: Sufficiency, minimality, invariance and deep approximation." *ArXiv preprint: 1411.7676*, 2014.

[SC15] S. Song and M. Chandraker. "Joint SFM and detection cues for monocular 3D localization in road scenes." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[SC16] S. Soatto and A. Chiuso. "Visual Representations: Defining properties and deep approximation." In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[SDK14] S. Soatto, J. Dong, and N. Karianakis. "Visual scene representations: Contrast, scaling and occlusion." *ArXiv preprint: 1412.6607*, 2014.

[SEZ14] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. "Overfeat: Integrated recognition, localization and detection using convolutional networks." In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[SGS13] S. Sengupta, E. Greveson, A. Shahrokni, and P. Torr. "Semantic modelling of urban scenes." In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[Sha98] J. Shao. *Mathematical Statistics.* Springer Verlag, 1998.

[Sha01] C. E. Shannon. "A mathematical theory of communication." In *Proceedings of the ACM SIGMOBILE Mobile Computing and Communications Review*, 2001.

[SHK12] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. "Indoor segmentation and support inference from rgbd images." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[SHL16] N. Savinov, C. Haene, L. Ladicky, and M. Pollefeys. "Semantic 3d reconstruction with continuous regularization and ray potentials using a visibility consistency constraint." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[SHP15] N. Savinov, C. Hane, M. Pollefeys, and et al. "Discrete optimization of ray potentials for semantic 3d reconstruction." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[SK13] G. Singh and J. Kosecka. "Nonparametric scene parsing with adaptive feature relevance and semantic context." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[SLJ15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[SMH11] J. Susskind, R. Memisevic, G. E. Hinton, and M. Pollefeys. "Modeling the joint density of two images under a variety of transformations." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[SNS13] R. Salas-Moreno, R. Newcombe, H. Strasdat, P. Kelly, and A. Davison. "Slam++: Simultaneous localisation and mapping at the level of objects." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[Soa10] S. Soatto. "Steps towards a theory of visual information: Active perception, signal-to-symbol conversion and the interplay between sensing and control." *ArXiv preprint: 1110.2053*, 2010.

[SOP07] T. Serre, A. Oliva, and T. Poggio. "A feedforward architecture accounts for rapid categorization." *Proceedings of the National Academy of Sciences*, **104**(15):6424–6429, 2007.

[SPV09] G. Sundaramoorthi, P. Petersen, V. S. Varadarajan, and S. Soatto. "On the set of images modulo viewpoint and contrast changes." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[SRA09] L. Sharan, Ruth. Rosenholtz, and Edward. Adelson. “Material perception: What can you see in a brief glance?” *Journal of Vision*, **9**(8), 2009.

[STL14] A. Sharma, O. Tuzel, and M. Liu. “Recursive context propagation network for semantic scene labeling.” In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[SVZ13] K. Simonyan, A. Vedaldi, and A. Zisserman. “Deep fisher networks for large-scale image classification.” In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[SVZ14a] K. Simonyan, A. Vedaldi, and A. Zisserman. “Deep inside convolutional networks: Visualising image classification models and saliency maps.” In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[SVZ14b] K. Simonyan, A. Vedaldi, and A. Zisserman. “Learning local feature descriptors using convex optimisation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **36**(8):1573 – 1585, 2014.

[SX16] S. Song and J. Xiao. “Deep sliding shapes for amodal 3d object detection in rgb-d images.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[SZ03] J. Sivic and A. Zisserman. “Video google: A text retrieval approach to object matching in videos.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[SZ05] S. Smale and D. X. Zhou. “Shannon Sampling: Connections to Learning Theory.” *Appl. and Comp. Harm. An.*, **19**(3), 2005.

[SZ15] K. Simonyan and A. Zisserman. “Very deep convolutional networks for large-scale image recognition.” In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[SZS14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. “Intriguing properties of neural networks.” In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[TCS15] K. Tsotsos, A. Chiuso, and S. Soatto. “Robust filtering for visual inertial sensor fusion.” In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[TFL10] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. “Convolutional learning of spatio-temporal features.” In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2010.

[TH14] M. Tau and T. Hassner. “Dense correspondences across scenes and scales.” *ArXiv preprint:1406.6323*, 2014.

[TLF10] E. Tola, V. Lepetit, and P. Fua. "Daisy: An efficient dense descriptor applied to wide-baseline stereo." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(5), 2010.

[TTD12] A. Toshev, B. Taskar, and K. Daniilidis. "Shape-based object detection via boundary structure segmentation." *International Journal of Computer Vision*, 2012.

[UBG16] A. O. Ulusoy, M. J. Black, and A. Geiger. "Patches, planes and probabilities: A non-local prior for volumetric 3d reconstruction." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[USG13] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. "Selective search for object recognition." *International Journal of Computer Vision*, 2013.

[VF10] A. Vedaldi and B. Fulkerson. "Vlfeat: An open and portable library of computer vision algorithms." In *Proceedings of the ACM International Conference on Multimedia*, 2010.

[VKM13] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. "Hog-gles: Visualizing object detection features." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[VL15] A. Vedaldi and K. Lenc. "MatConvNet: Convolutional neural networks for MATLAB." In *Proceedings of the ACM Conference on Multimedia Conference*, 2015.

[VML15] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Pérez, and et al. "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction." In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[Wal81] D. L. Waltz. "Understanding and Generating Scene Descriptions." In *Elements of Discourse Understanding*, pp. 266–281. Cambridge University Press, 1981.

[Wat83] G. S. Watson. *Statistics on spheres.* Wiley, 1983.

[WB07] S. Winder and M. Brown. "Learning local image descriptors." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[WFU15] S. Wang, S. Fidler, and R. Urtasun. "Holistic 3d scene understanding from a single geo-tagged image." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[WLS14] C. Wu, I. Lenz, and A. Saxena. "Hierarchical semantic labeling for task-relevant rgb-d perception." *Robotics: Science and systems*, 2014.

[XCL15] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. "Data-driven 3d voxel patterns for object category recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[XCL17] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. "Subcategory-aware convolutional neural networks for object proposals and detection." In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2017.

[XOT13] J. Xiao, A. Owens, and A. Torralba. "Sun3d: A database of big spaces reconstructed using sfm and object labels." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013.

[YCB14] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. "How transferable are features in deep neural networks?" In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[YFU12] J. Yao, S. Fidler, and R. Urtasun. "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[YLS15] Y. Yang, Z. Lu, and G. Sundaramoorthi. "Coarse-to-fine region selection and matching." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[ZCV15] R. Zhang, S. Candra, K. Vetter, and A. Zakhor. "Sensor fusion for semantic segmentation of urban scenes." In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[ZD14] C. L. Zitnick and P. Dollár. "Edge boxes: Locating object proposals from edges." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[ZF14] M. D. Zeiler and R. Fergus. "Visualizing and understanding convolutional networks." In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[ZSS15] M. Z. Zia, M. Stark, and K. Schindler. "Towards scene understanding with detailed 3d object representations." *International Journal of Computer Vision*, 2015.

[ZZD15] M. Zhu, X. Zhou, and K. Daniilidis. "Single image pop-up from discriminatively learned parts." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.