Registration and Representation in Computer Vision

by

Hilton Bristow
B.E. Mechatronics (Hons I)

Science and Engineering Faculty
Queensland University of Technology

A dissertation submitted in fulfilment of the
requirements for the degree of
Doctor of Philosophy
2016

Keywords: capacity, locality, stationarity, sparsity, semantic correspondence, SVM, LDA, HOG, SIFT, ADMM.

In accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Science and Engineering, I present the following thesis entitled,

Registration and Representation in Computer Vision

This work was performed under the supervision of Professor Simon Lucey and Professor Sridha Sridharan. I declare that the work submitted in this thesis is my own, except as acknowledged in the text and footnotes, and has not been previously submitted for a degree at The Queensland University of Technology or any other institution.

Hilton Bristow

To those who seek knowledge for the satisfaction of the mind,
and understanding of the world

Publications

Hilton Bristow, Jack Valmadre and Simon Lucey. Dense Semantic Correspondence where Every Pixel is a Classifier. *International Conference on Computer Vision (ICCV)*, 2015 [Under Review].

Hilton Bristow and Simon Lucey. Regression-Based Image Alignment for General Object Categories. Chapter in *Dense Correspondences in Computer Vision, Springer SBM*, 2015.

Hilton Bristow and Simon Lucey. Why do Linear SVMs Trained on HOG Features Perform so Well?, *arXiv* preprint, 2014.

Hilton Bristow and Simon Lucey. Optimization Methods for Convolutional Sparse Coding. *arXiv* preprint, 2014.

Hilton Bristow, Anders Eriksson and Simon Lucey. Fast Convolutional Sparse Coding. *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Hilton Bristow and Simon Lucey. V1-Inspired Features Induce a Weighted Margin in SVMs. *European Conference on Computer Vision (ECCV)*, 2012.

Acknowledgements

For me, embarking upon a PhD would become as much an excercise in discovering the generosity of others as it would be in learning about my own joys, talents and deficiencies. There have been inumerable people who have made my time fun and truly fulfilling.

First and foremost, I would like to acknowledge my principal supervisor Professor Simon Lucey. He was endlessly patient with me, and quickly learned where I needed support and where I could really thrive. He gave me opportunities I never thought possible, and immense freedom to learn and make my own mistakes. He put his trust in me, and often believed in my ideas more than I did.

I would also like to thank my secondary supervisor Professor Sridha Sridharan, with whom I had little contact but for whom I will always have great respect. He sheltered me from the realities of university politics so I could focus on my research, and kept me on the straight and narrow.

To my esteemed colleagues and friends Mark Cox, Jason Saragih, Jesus Nuevo, Kit Ham and Ashton Fagg, with whom I spent countless hours scrawling on whitebaords or discussing problems over pints. Many interesting ideas came from these exchanges, and more than a few grew to become the basis of this thesis. I would like to single out Mark in particular, who became a mentor to me and challenged many of my thought processes. He was a patient but persistent teacher, and all credit goes to him for convincing me to learn Common Lisp. I'm a more talented and condescending programmer for it.

To my dearest friends Kieran Wynn and Jack Valmadre who have been there for me all the way. They each provided support and comfort in their own special way, and we had a lot of fun between the hard work. I'm sure there are still many good times ahead, and no matter what continent, my home will always be open to you.

It would be remiss of me not to mention that one special teacher in my formative years who helped to shape my life ambitions. Brad Hampson was my 5th grade teacher, and kindled my interest in science and reason. The products of a teacher's labours are so often subsumed and temporally delayed, and I hope I can one day find a way to manifest this appreciation.

To my family, my pillars and foundations, who have supported me at each stage of life, and stood by every decision. Mum and dad introduced me to the beauty of logic and discovery from an early age, and always challenged me to question my understanding of how things work. We've had countless conversations stretching into the early hours of the morning discussing life, the universe and everything. Dad helped me realize the beauty of the natural world, its plight, and how to tread lightly. Mum taught me how to take care of my mind, and introduced me to the joy of yoga. My sister, Jules, is the glue of our family. She has been my cheerleader, always encouraging and excited to hear what I'm doing.

Finally, to Ana, who decided to put up with me on her own volition. Her smile is the first thing I see every morning, and her hand is always open, ready to take me on the next adventure.

Abstract

VISUAL OBJECT PERCEPTION is central to computer vision research. It is a particularly compelling computational problem because human performance at the task is exceptional, and could thus yield valuable insights into how the brain functions. Advances in visual perception over the last few years can largely be attributed to a more sophisticated understanding of which prior assumptions to encode in image representations. Even in fully learned architectures such as convolutional networks, prior plays an important role in steering models to a good solution given the relative scarcity of labelled training data.

In fact, until recently, image representations had relied solely upon prior to perform well: they had neither conception of the objective being solved, nor the training data available. It is remarkable, therefore, how well they historically performed. Understanding the underlying principles of their success is central not only to improving the theory of visual perception and its mechanics, but also to improving priors for supervised and unsupervised applications alike.

This dissertation investigates a number of priors that have proven invaluable to visual perception, and how they can be effectively and efficiently leveraged across a broad range of learning contexts. We focus on the problem of managing uncertainty in geometric alignment between images. That is, when describing the semantic similarity between two misaligned images, how should that misalignment be treated?

We consider two common but opposing strategies for dealing with geometric misalignment, first by marginalizing over the uncertainty, and then by solving for it. In both situations, we demonstrate how *capacity*, *locality*, *stationarity* and *sparsity* are effective mechanisms for dealing with uncertainty. We close by presenting new methods for aligning novel images based on their semantic content, by efficiently leveraging these prior assumptions.

Contents

Introduction

It is well recognized that for visual recognition tasks, a nearest-neighbour classifier is optimal given infinite training data. Classification reduces to a simple lookup operation that indexes into the perfect world knowledge. This idealistic situation is far from attainable, however. A realistic goal of visual recognition is therefore to recover semantic properties of images, and learn models that draw upon these properties to generalize to unseen images and scenarios. Central to this definition of visual recognition is the concept of *prior* – the knowledge, structure and statistical assumptions to encode in the absence of data.

Training data is a precious resource in object detection. There is a constant conflict between developing more sophisticated models, and collecting enough training data to satisfy these models. As a result, understanding which prior assumptions models should encode is integral to their continued improvement. Even in fully learned architectures such as convolutional networks, prior plays an important role in steering models to a good solution given the relative scarcity of labelled training data.

Labelling training data is an expensive and time-consuming process, and while work on crowd engineering [76, 91, 92] has reduced the human interaction required to generate labelled data, the real costs involved are still significant. As a result, learning from vast quantities of *unlabelled* data with minimal feedback is a practical long-term objective for computer vision. This dissertation investigates a number of approaches to leveraging statistical structure in an unsupervised manner, to assist in solving both supervised and unsupervised problems.

A fundamental question in computer vision and neuroscience alike asks *how does the mammalian visual system attain a high degreee of invariance, whilst*

Figure 1: A pair of images stemming from the same visual class. Although the images have photometric, geometric and stylistic differences, at a macro level they are both clearly of lions. At a micro level, human annotators can precisely localize numerous features across the two lions. The third panel illustrates the agreement between subjects on one such task.

maintaining selectivity? Under 2D image projections, geometry and appearance are intrinsically related. The apparent shape of an object is defined by its projected appearance, and its projected appearance is influenced by light interacting with the true $3D$ shape. The correlation of multiple sources of variation contributes significantly to the challenge of visual recognition.

Consider the objects that appear in Figure 1. While they evidently represent "lions", there is little in the way of raw pixel information that suggests the images are related. The pixel values at fixed locations within the images are significantly different, as are differences between pairs of pixels at a fixed displacement, indicating variation in lighting and geometry across the two images. Despite this, humans are able to reflexively recognize the objects as lions, and precisely localize similar features between them (*e.g.* nose, ears, eyes).

In order to reliably compare the content of two images, we require (i) a method of representing the semantic content in the images, and (ii) a metric for computing the semantic similarity in this representational space. Naively, one could represent each image as a vector of pixels $\mathbf{x}_A$ and $\mathbf{x}_B$, and estimate their similarity by computing the Euclidean distance,

$$y = ||\mathbf{x}_A - \mathbf{x}_B||_2^2 \,. \tag{1}$$

Whilst y is a good measure of pixel similarity, it is a poor measure of *perceptual* similarity. Indeed, it is well established that pixels are a poor representation for comparing aspects of images on a semantic level, since they conflate many sources of variation.

To understand this intuitively, we reproduce a thought experiment by Kingdom, Field and Olmos [28] in Figure 2. The top panel shows a reference image, and the bottom panel shows pertubations of that image under different types of

Figure 2: Perturbations of an image with different sources of deformation: (left) with white noise, and (right) horizontal stretching. Although the left image appears more perceptually corrupt than the right image, both have identical Euclidean distance to the reference image.

deformation. The left image is perturbed by adding random white noise, while the right image is perturbed by an affine transform, approximating a small change in viewpoint of the scene. The magnitude of the perturbations applied are chosen such that the Euclidean distance between the reference and the two perturbed images is the same. To observers, however, the left image is more perceptually corrupted than the right image. Indeed, the right image is visually indistinguishable from the reference image.

This illustrates two important points: (i) Euclidean distance on pixel intensities is a poor measure of perceptual similarity, and (ii) the human visual system is least sensitive to the transformations most commonly experienced in the natural world.

As a result, significant object detection literature has focused on how to

represent and match images, such that,

$$y = \mathcal{K}(\Phi(\mathbf{x}_A), \Phi(\mathbf{x}_B)) \tag{2}$$

is a good estimate of perceptual similarity, where Φ is a feature transform and $\mathcal{K}$ is the matching function.

Historically, object detection tasks have relied on a fixed feature space in which an "optimal" matching function is learned. Given a set of labelled training data, the matching function can be learned (in a possibly convex, possibly optimal manner) to maximize (minimize) its output for similar images, and minimize (maximize) its output for different images.

The representation and matching function are intrinsically related, since complexity in one can be traded for simplicity in the other. A sufficiently complex representation can entertain simple linear or nearest-neighbour classifiers. Conversely, a simple representation requires a high capacity matching function.

At both extremes, however, the algorithm must distil properties useful for recognition, either structurally or through the training regime, in order to avoid overfitting or infinite complexity in time and space.

Fundamentals of Visual Recognition

The mammalian visual cortex has evolved over millions of years to efficiently cope with observations of the natural environment. Resource scarcity is a very real constraint for evolutionary processes, and metabolic efficiency is driven by the relative scarcity of energy. Given these constraints, and the remarkable proficiency with which animals observe their environment, it is believed that the visual cortex has discovered a particularly efficient way to represent the visual world.

The notion of efficiency in image representation is founded largely upon Barlow's principle of redundancy reduction [6]. In contrast with Shannon, who was primarily interested in redundancy from a transmission perspective, he described redundancy reduction in sensory coding as decorrelating statistical dependencies between inputs, or at least making them explicit.

In fact, in order to accurately estimate the probability of a particular real-world event given a retinal stimulus, the working representation should be as sparse as possible, which necessarily implies an *increase* in redundancy (in an information sense).

This can be understood by considering the process of image formation. Light falling on the retina is encoded by photoreceptors which, via the optic nerve, act as an input to the visual cortex. The activities of the photoreceptors represent the complex statistical dependencies arising from the interaction of geometry and lighting being projected onto the retina. As a consequence, visually observing the real world is inherently ambiguous. In the context of biology, the significance of the inverse problem is clear: if the information on the retina precludes direct knowledge of the real world, how is it that we can interact so effortlessly with our environment?

One answer is that the hierarchical visual system progressively decorrelates more complex sources of variation, whilst quantifying uncertainty over each source, and making predictions based on information gain, utility gain, or risk minimization.

A significant problem that has confounded neuroscientists and computer vision researchers alike, is what primitive computational operations are required to decorrelate visual signals.

Primary Visual Cortex

The primary visual cortex (V1) produces the initial neural representation of the imaged world, and forms the foundation for all higher-level visual cognition. V1 is a particularly compelling starting point to study visual computation since it is retinotopic – adjacent regions within the visual field correspond directly to adjacent regions within V1.

The canonical V1 model was first proposed by Hubel and Wiesel in their seminal 1962 work on the cat's visual cortex [39]. By probing cortical neurons with electrodes, the cells fired action potentials only when a bar of light was in a certain part of the visual field, and at a certain orientation. By probing a neighbourhood of neurons, they were able to generate to topographic map of responses. They showed that complex V1 cells exhibited selectivity to specific types of stimuli, particularly oriented edges at different frequencies.

Coding

While Hubel and Wiesel's work effectively demonstrated *what* V1 was doing, it failed to describe *how* it worked, both in operation and its computational objective. Marr and Poggio were particularly instrumental in pursuing a different

approach to vision, which enquired directly about the information processing problems inherent in the task of vision [61, 72]. Central to this approach was discovering properties of the visual world that constrained the computational problem to make it well-defined and solvable.

The appeal of such an approach is that it leads to a science solidly based on the physics of the real world and on the basic laws of image formation. We now understand that forming a computational theory of vision is more complex than ever anticipated, however there have been a number of pivotal findings which have led to a better understanding of the computational challenge inherent in vision.

Olshausen and Field showed in their seminal 1996 Nature article that the V1 model can be posed as a sparse coding problem [67]. For the first time, the empirical observations of V1 were reflected by a well-defined computational goal. That goal was to minimize the reconstruction error of a set of natural image patches from a learned basis and coefficients, with the coefficients subjected to sparsity constraints. With careful preprocessing of the image patches, the learned basis elements approximately represent Gabor filters – frequency and orientation selective edge filters.

When the dictionary is overcomplete (the number of basis elements exceeds the signal dimensionality), accurate reconstructions can be achieved with few active coefficients. In this case the basis learned is not orthogonal, since the number of bases required to span the full rank of the space is exceeded, and so signal redundancy increases. Per Barlow's conjecture, the reconstruction coefficients more uniquely and compactly describe the *structure* of the underlying signal than the original pixel (retinal) representation. It would appear that a similar coding process forms the first of the visual cortex's hierarchy of decorrelation.

The principal component that enables sparse coding is the statistical regularity of the natural world. There are two important aspects to this statement. First, the statistics of natural images are *stationary*. That is, a translated natural image still forms a valid natural image, and has the same statistical structure. Second, that statistical structure is non-random. It is well established that natural images obey a $\frac{1}{F}$ frequency power spectrum [29]. This means that most of the variance in natural images is low-frequency, with very little information content in the higher frequencies. However, that variance in natural images is structured. While they follow the same power spectrum as pink noise, they have significantly more regularity, owing to the regularity of geometric objects in the world. These regularities have/ given rise to a tech-

nique called *compressive sensing*, which breaks the conventional wisdom about the sampling limit required to reconstruct an image from an incomplete number of observations [16].

Whilst sparse coding was a major breakthrough for the theory of visual cognition, it only went so far to describing the algorithmic behaviour of the visual cortex. Sparsity clearly plays an important functional role in coding, however the computational procedure of sparse coding as we currently formulate it is far too computationally demanding for the cortex to actually perform.

Stationarity

Stationarity of natural image statistics plays an important role in recognition, and especially learning. As discussed earlier, the primary visual cortex leverages stationarity to learn compact representations of the visual world, and assist in the process of decorrelating sources of variation.

More recently, stationarity has been leveraged as an effective mechanism for describing the distribution of natural images in a particular feature space. Such a distribution forms the basis of a "negative class," since in many detection problems any window within any image that does not contain the object of interest is considered a negative example.

Image Representations for Recognition

Visual representations in computer vision have historically been motivated by a combination of understandings of the visual cortex, computational constraints, and clever engineering. A majority of these employ a similar strategy:

- convolution with a bank of (possibly orientation selective) filters
- a rectifying (*i.e.* non-negative) non-linear transform
- pooling to achieve tolerance to local geometric variation
- cross-channel normalization to achieve tolerance to local contrast variation

This general strategy is pervasive amongst HOG [22], SIFT [56], SURF [7], LBP [66], BRIEF [15], BRISK [51] and FREAK [1]. Convolutional networks typically compose many layers all derived from this strategy.

The remarkable aspect of these handcrafted representations is that they depend *solely* on prior belief about the types of pixel interactions important to

visual recognition. That is, the representation can be described independently of data or a learning procedure. In fact, V1 inspired features paired with a linear classifier still form the basis of many visual recognition tasks including pedestrian detection, facial landmark localization and object tracking. In other words, a feature representation with *no knowledge of image content* paired with a *linear decision boundary* is still one of the most capable methods for visual recognition.

Matching Functions

Historically, object detection tasks have relied on a fixed image representation on which an "optimal" matching function is learned. That representation typically encodes the types of tolerance to geometric and photometric transformations described above. Given a set of labelled training data $\mathbf{x}$, and target labels y, the matching function can be learned (in a possibly convex, possibly optimal manner) such that,

$$\mathcal{K}(\mathbf{x}_A, \mathbf{x}_B) = \begin{cases} 1 & y_A = y_B \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Another approach to visual recognition is to solve an optimization problem within the matching function to account for geometric deformations rather than gain invariance to them. This has the additional benefit of modelling structured information about the prediction (warp parameters, location of parts, *etc.*) but is typically more computationally demanding. In such a setting, the matching might resemble,

$$y = \mathcal{K}(\Phi(\mathbf{x}_A), \Phi(\mathbf{x}_B)) = \min_{\mathbf{p} \in \mathcal{P}} k(\Phi(\mathbf{x}_A), \Phi(\mathbf{x}_B(\mathbf{p}))) \tag{4}$$

where k is a measure of alignment quality, $\mathbf{p}$ is a parametrization of the image that allows it to deform, and $\mathcal{P}$ is the set of allowable warps. Structured output detection methods such as deformable face fitting, parts modelling, optical flow and semantic segmentation fall into this category.

The representation and matching function are intrinsically related, since complexity in one can be traded for simplicity in the other. A sufficiently complex representation can entertain simple linear or nearest-neighbour classifiers. Conversely, a simple representation requires a high-capacity matching function. More recently, people have started optimizing both the representation

and matching function, in effect trading the global optimality of the matching function for joint local optimality of the representation and matching function.

Convolutional networks, in particular, blur the distinction between representation and matching, invariance and alignment. They pair high capacity with substantial training data to treat visual recognition as a learnable function from pixels to class labels. However, there exists a gap between what is theoretically representable by the network, and what can actually be learned with the given data and optimization procedure. Improvements in our understanding of visual mechanics and physics - expressed through prior - will in turn provide better constraints on the behaviours that high-capacity networks should learn.

Challenges

One of the fundamental questions in computer vision regards how to best represent object appearance in the face of geometric and photometric distortions. On *how* to address this question, the computer vision community is divided. There are broadly two schools of enquiry: (i) data driven, and (ii) model driven recognition. The former seeks to reduce assumptions and prior and let the structure of the problem fall from the data alone. This very general approach is powerful, and performs well given sufficient data. However, since the behaviour of the system is largely emergent, it can be difficult to interpret the error modes and learned structure. At the other end of the spectrum, model driven recognition attempts to explicitly describe the structure of the problem. This is useful in instances when data is limited or the domain is constrained, but requires significant domain knowledge to build the system.

Much of this thesis focuses on a middle ground, building a theory, understanding and application of the mechanics of vision. We approach the problem of visual recognition with a number of "axiomatic principles." We leverage natural image statistics as a basis for forming image representations and matching functions, and convolution as a natural way of embedding invariance to geometry. We tackle unsupervised learning problems, or problems where labelled training data is scarce.

Contributions

This thesis makes three key contributions to the visual perception literature. First, we present an empirically fast approach to the convolutional sparse coding problem. Stationarity of natural images can be efficiently exploited to perform sparse coding, and unlike traditional patch-based sparse coding, explicitly modeling stationarity with the convolution operator leads to more expressive bases. This comes with an extra computational burden, however, limiting its appeal. Our fast method makes it a useful building block for many higher-levels tasks.

Second, we show how the translation-invariant properties of many V1-inspired representations arise from a strategy for preserving local pairwise interactions of pixels. We directly relate the contribution such features make towards the capacity of an associated linear support vector machine used for classification. We show a direct link between the amount of training data required and the degree of geometric uncertainty for a task.

Finally, we present two methods for recovering the geometric uncertainty between misaligned images. We consider the problem of aligning general object classes, which is inherently difficult due to the lack of precisely annotated training data. We show that natural imagery can be effectively used as a source of pre-training data to assist in representing semantic correspondences between pixels. We also observe that V1-inspired representations, which are usually used for gaining invariance to geometry, are also often necessary as a basis for solving for geometric misalignment.

Outline

This thesis dedicates a chapter to discussing each of three concepts for efficiently utilizing prior information in visual recognition. We begin Chapter 1 by discussing *coding* and *sparsity*, the most fundamental component of visual processing and the earliest component performed in the mammalian visual system. We show how the stationarity of natural image statistics is integral to coding, and present an efficient method for leveraging stationarity to perform sparse coding. We further show how the resulting codes can form the basis of translation invariance.

In Chapter 2, we consider *locality* and *capacity* through a canonical V1-inspired representation, and show how its translation invariant properties can arise from a strategy for sampling geometric perturbations of an underlying

training set whilst preserving only local pairwise interactions between pixels. From this we observe that such a strategy induces added capacity in the classifier applied, and this capacity is critical to good classification, even under controlled geometric alignment of the dataset.

Finally, in Chapter 3, we look at *stationarity* and *correspondence* within the context of unsupervised alignment of general object classes. We show two intriguing properties of V1-inspired representations under such conditions. First, traditional gradient-based alignment methods can be applied - even though the features themselves are non-differentiable - by estimating descent directions via efficient regression. Second, the stationarity of natural images can once again be leveraged to compactly describe the infinite negative class that forms when performing pixel-wise alignment.

We conclude the thesis with a brief summary of the chapters, and some closing remarks regarding the use of prior in computer vision. Since each chapter considers a different application of the underlying principles of prior, the relevant works are presented within the chapter. As such, each chapter aims to be independently readable without necessitating context from other chapters.

Notation

Before we begin, a brief aside to discuss notation throughout this piece. Regular face symbols (*i.e.* n, N) indicate scalars, with lowercase variants reserved for indexing and uppercase for ranges and dimensions. Boldface symbols (*i.e.* $\mathbf{x}, \mathbf{X}$) represent vectors and matrices. Caligraphic symbols (*e.g.* $\mathcal{W}$) represent functions. We sometimes refer to images as functions rather than vectors or matrices to indicate that non-integer pixels can be addressed (by sub-pixel interpolation). To keep notation terse, we often vectorize expressions. Therefore, in many instances, functions have vector-valued outputs, though we endeavour to be explicit or obvious when this happens.

Chapter 1

Coding and Sparsity

Olshausen and Field showed in their seminal 1996 article in Nature that orientation and frequency selectivity in V1 arise naturally as a solution to a visual sparse coding problem [67]. This was the first time that the empirical effectiveness of V1 was met with an understanding of the computational problem it was solving. The role of sparsity was motivated by the metabolic constraints of neuron firing, which is both energy intensive and has an associated refractory period. A sparse representation reflects the sparse neural activation observed in response to natural stimuli.

Inspired by this result and more recent works on compressive sensing, sparse constraints now form an integral part of many representational learning objectives. However, many such approaches operate on "patches" of the input images. Independent sampling of patches from the input, however, ignores any structure in the original signal.

Consider the signal in Figure 1.1 reproduced from [52]. The signal is composed of two distinct modes, appearing at multiple intervals (you could consider the modes to be expressions that are repeated over the course of a conversation, features that appear multiple times within an image, or particular motifs in a musical passage or speech).

If the signal is segmented into blocks, the latent structure becomes obfuscated, and the basis learned must have the capacity to reconstruct each block in isolation. In effect, we learn a basis that is higher rank than the true basis due to the artificial constraints placed on the temporal alignment of the bases.

Convolutional sparse coding makes no such assumptions, allowing shiftable basis functions to discover a lower rank structure, and we make the case that it should therefore be preferred in situations where the basis alignment is not known *a priori*.

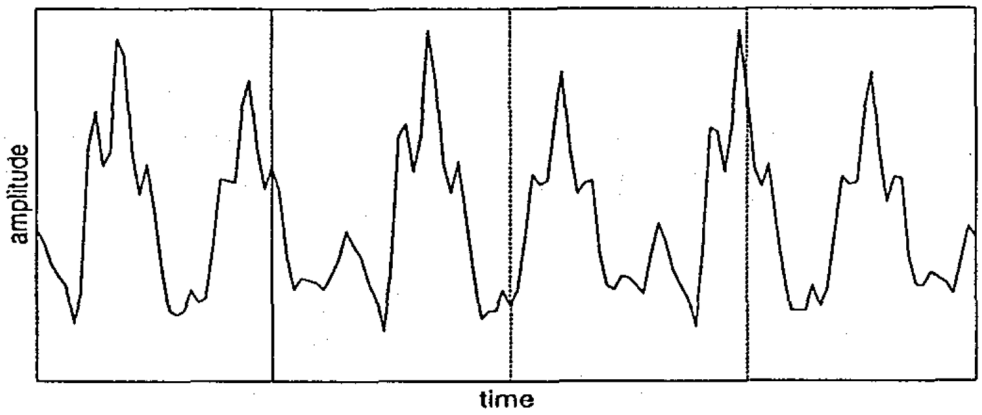


Figure 1.1: Blocking results in arbitrary alignment of the underlying signal structure, artificially inflating the rank of the basis required to reconstruct the signal. Convolutional sparse coding relaxes this constraint, allowing shiftable basis functions to discover a lower rank structure.

Sparse and convolutional constraints form a natural prior for many optimization problems that arise from physical processes. Detecting motifs in speech and musical passages, super-resolving images, compressing videos, and reconstructing harmonic motions can all leverage redundancies introduced by convolution. Solving problems involving sparse and convolutional constraints remains a difficult computational problem, however.

In this chapter, we present an overview of convolutional sparse coding, and introduce an algorithm for performing the optimization in an efficient manner. We present a broad suite of examples covering different signal and application domains to illustrate the general applicability of convolutional sparse coding, and the efficacy of the available optimization methods.

1.1 Related Work

The notion of translation invariant optimization stems from the thought experiments of Simoncelli *et al.* [80]. His motivation for considering translation invariance came from the context of wavelet transforms for signal processing. Amongst others, he had observed that block-based wavelet algorithms were sensitive to translation and scaling of the input signal.

As an example, he chose an input signal to be one of the wavelet basis functions (yielding a single reconstruction coefficient), then perturbed that signal slightly to produce a completely dense set of coefficients. The abrupt change in

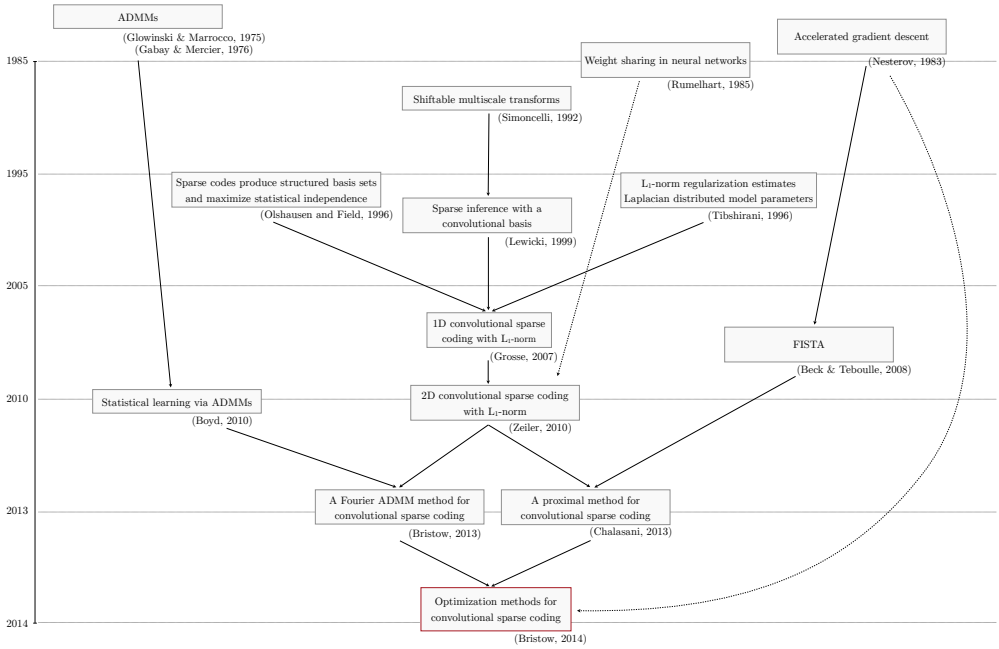


Figure 1.2: A brief history of the works that influenced the direction of the convolutional sparse coding problem, and how it is optimized. The text within each box indicates the theme or idea that the paper introduced.

representation in the wavelet domain due to a small change in the input illustrated the wavelet transform's unsuitability for higher-level summarization.

Olshausen and Field showed that sparsity alone is a sufficient driver for learning structured overcomplete representations from signals, and used it to learn a basis for natural image patches [67]. The resulting basis, featuring edges at different scales and orientations, was similar to the receptive fields observed in the primary visual cortex.

The strategy of sampling patches from natural images has come under fire however, since many of the learned basis elements are simple translations of each other - an artefact of having to reconstruct individual patches, rather than entire image scenes [44]. Removing the artificial assumption that image patches are independent - by modelling interactions in a convolutional objective - results in more expressive basis elements that better explain the underlying mechanics of the signal.

Lewicki and Sejnowski [52] made the first steps towards this realization, by finding a set of sparse coefficients (value and temporal position) that reconstructed the signal with a fixed basis. They remarked at the spike-like responses observed, and the small number of coefficients needed to achieve satisfactory

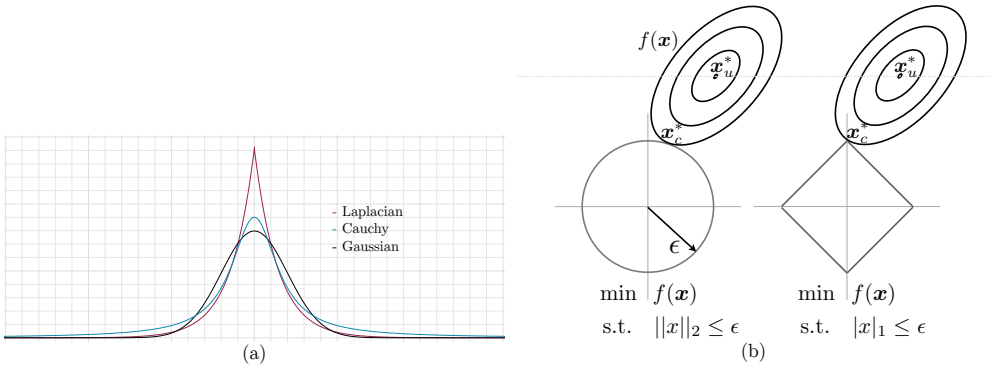


Figure 1.3: Sparsifying distributions and their effect as regularizers. (a) For a given variance, the Laplacian and Cauchy distributions have more probability mass centered around zero. (b) As a result, $L1$ constraints favour axis-aligned solutions (only the second dimension is non-zero).

reconstructions.

Introducing sparsity brought with it a set of computational challenges that make the resulting objectives difficult to optimize. [67] explored coefficients drawn from a Cauchy distribution (a smooth heavy-tailed distribution) and a Laplacian distribution (a non-smooth heavy-tailed distribution), citing that in both cases *they favour among activity states with equal variance, those with fewest non-zero coefficients.* [67] inferred the coefficients as the equilibrium solution to a differential equation. [52] assumed Laplacian distributed coefficients, and noted that due to the high sparsity of the desired response, it would be sufficient to replace exact inference with a procedure for guessing the values and temporal locations of the non-zero coefficients, then refining the results through a modified conjugate gradient local search.

Tibshirani [85] introduced a convex form of the sparse inference problem - estimating Laplacian distributed coefficients which minimize a least-squares reconstruction error - using L_1-norm regularization and presented a method for solving it with existing quadratic programming techniques.

The full convolutional sparse coding algorithm culminated in the work of Grosse *et al.* [33]. Fundamentally, Grosse extended Olshausen and Field's sparse coding algorithm to include convolutional constraints and generalized Lewicki and Sejnowski's convolutional sparse inference to 2D. Algorithmically, Grosse drew on the work of Tibshirani [85] in expressing Laplacian distributed coefficients as L_1-norm regularization, and used the feature sign search minimization algorithm proposed by his colleague in the same year [49] to solve it efficiently. The form he introduced is the now canonical bilinear convolutional

sparse coding algorithm.

Convolutional sparse coding has found application in learning Gabor-like bases that reflect the receptive fields of the primary visual cortex [68], elemental motifs of visual [101], speech [33, 52] and musical [33, 64] perception, a basis for human motion and articulation [106], mid-level discriminative patches [82] and unsupervised learning of hierarchical generative models [50, 101].

The large-scale nature of the latter applications have placed great demands on the computational efficiency of the underlying algorithms. Coupled with the steady advances in machine learning and computing, this has given rise to a range of optimization approaches for convolutional sparse coding. Chalasani *et al.* [17] introduced a convolutional extension to the FISTA algorithm for sparse inference [8]. We introduce a Fourier method based on the closely-related ADMM [12].

This chapter addresses the following concepts:

1. We argue that *convolutional* sparse coding makes a set of assumptions that are more appropriate in many tasks where block or sampled sparse coding is currently used.

2. We discuss practical optimization of convolutional sparse coding, including speed, memory usage and assumptions on boundary conditions.

3. We show through a number of examples the applicability of convolutional sparse coding to a wide range of problems that arise in computer vision, and when particular problems benefit from different optimization methods.

1.2 Convolutional Sparse Coding

The convolutional sparse coding problem consists of minimizing a convolutional model-fitting term f and a sparse regularizer g,

$$\arg\min_{\mathbf{d},\mathbf{z}} \quad f(\mathbf{d},\mathbf{z}) + \beta g(\mathbf{z}) \tag{1.1}$$

where $\mathbf{d}$ is the convolutional kernel, $\mathbf{z}$ are the set of sparse coefficients, and β controls the tradeoff between reconstruction error and sparsity of representation. The input can be reconstructed via the convolution,

$$\mathbf{x} = \mathbf{d} * \mathbf{z} \; . \tag{1.2}$$

Assuming Gaussian distributed noise and Laplacian distributed coefficients, Equation 1.1 can be written more formally as,

$$\begin{aligned} \arg\min_{\mathbf{d},\mathbf{z}} \quad & \frac{1}{2}||\mathbf{x} - \mathbf{d} * \mathbf{z}||_2^2 + \beta||\mathbf{z}||_1 \\ \text{subject to} \quad & ||\mathbf{d}||_2^2 \leq 1 \end{aligned} \tag{1.3}$$

The remainder of our analysis is based around efficient methods of optimizing this objective. The objective naturally extends to multiple images and filters,

$$\begin{aligned} \arg\min_{\mathbf{d},\mathbf{z}} \quad & \frac{1}{2}\sum_{i=1}^{M}||\mathbf{x}_i - \sum_{j=1}^{N}(\mathbf{d}_j * \mathbf{z}_{i,j})||_2^2 + \beta\sum_{i=1}^{M}\sum_{j=1}^{N}||\mathbf{z}_{i,j}||_1 \\ \text{subject to} \quad & ||\mathbf{d}_j||_2^2 \leq 1 \;\; \forall\, j \in 1 \ldots N \end{aligned} \tag{1.4}$$

however this form quickly becomes unwieldy, so we only use it when the number of filters or images needs to be emphasized.

Contrast the objective of Equation 1.3 with that of conventional sparse coding,

$$\begin{aligned} \arg\min_{\mathbf{B},\mathbf{Z}} \quad & ||\mathbf{X} - \mathbf{B}\mathbf{Z}||_2^2 + \beta||\mathbf{Z}||_1 \\ \text{subject to} \quad & ||\mathbf{B}_i||_2^2 \leq 1 \;\; \forall\, i \end{aligned} \tag{1.5}$$

Here, we are solving for a set of basis vectors $\mathbf{B}$ and sparse coefficients $\mathbf{Z}$ in alternation that reconstruct patches or samples of the signal $\mathbf{X}$ *in isolation*. This is an important distinction to make, and forms the fundamental difference between convolutional and patch-based sparse coding.

In the limit when $\mathbf{X}$ contains *every patch* from the full image, the two sparse coding algorithms behave equivalently, however the patch-based algorithm must store a redundant amount of data, and cannot take advantage of fast methods for evaluating the inner product of the basis with each patch (*i.e.* convolution).

The objective of Equation 1.3 is bilinear – solving for each variable whilst holding the other fixed yields a convex subproblem, however the objective is not jointly convex in both. We optimize the objective in alternation, iterating until convergence. There are no guarantees that the final minima reached is the global minima, however in practice multiple trials reach minima of comparable quality, even if the exact bases and distribution of coefficients learned are slightly different.

In the sections that follow, we introduce a range of algorithms that can

solve for the filters and coefficients. Since the alternation strategy treats each independently, we largely consider the algorithms in isolation, however some care must be taken in matching appropriate boundary condition assumptions.

1.2.1 Other Formulations

It should be noted that the algorithm of Equation 1.3 is not the only conceivable formulation of convolutional sparse coding. In particular, there has been growing interest in non-convex sparse coding with hyper-Laplacian, L_{0+} and other exotic priors [102, 96]. Whilst these methods have not yet been extended to *convolutional* sparse coding, there is no fundamental barrier to doing so.

1.3 Solving for Coefficients

It is natural to begin with methods for convolutional sparse inference, since this is where most of the research effort has been focussed. Solving for the coefficients $\mathbf{z}$ involves optimizing the unconstrained objective,

$$\arg\min_{\mathbf{z}} ||\mathbf{x} - \mathbf{d} * \mathbf{z}||_2^2 + \beta||\mathbf{z}||_1 \tag{1.6}$$

where β controls the tradeoff between sparsity and reconstruction error. This objective is difficult to solve because (i) it involves a non-smooth regularizer, (ii) the least-squares system cannot be solved directly (forming explicit convolutional matrices for large inputs is infeasible), and (iii) the system involves a large number of variables, especially if working with megapixel imagery, *etc.*

In the case of multiple images and filters,

$$\begin{aligned}\arg\min_{\mathbf{z}} \; & \frac{1}{2}\sum_{i=1}^{M} ||\big(\mathbf{x}_i - \sum_{j=1}^{N}(\mathbf{d}_j * \mathbf{z}_{i,j})\big)||_2^2 \\ & + \beta \sum_{i=1}^{M}\sum_{j=1}^{N} ||\mathbf{z}_{i,j}||_1\end{aligned} \tag{1.7}$$

and given that the minima of the sum of convex functions is the sum of their minima,

$$\min_{\mathbf{z}} \sum_{i=1}^{M} f_i(\mathbf{z}) \;=\; \sum_{i=1}^{M} \min_{\mathbf{z}} f_i(\mathbf{z}) \tag{1.8}$$

the coefficients for each image can be inferred separately,

$$\mathbf{z}_i^\star = \arg\min_{\mathbf{z}} \ \frac{1}{2}||\mathbf{x}_i - \sum_{j=1}^{N}(\mathbf{d}_j * \mathbf{z}_{i,j})||_2^2 + \beta \sum_{j=1}^{N} ||\mathbf{z}_{i,j}||_1 \tag{1.9}$$

The two methods that we introduce are both based around partitioning the objective into the smooth model-fitting term and non-smooth regularizer that can then be handled separately.

1.3.1 ADMM Partitioning

The alternating direction method of minimizers (ADMM), was proposed jointly by [32, 31], though the idea can be traced back as early as Douglas-Rachford splitting in the mid-1950s. [12] presents a thorough overview of ADMMs and their properties. ADMMs solve problems of the form,

$$\begin{aligned} \arg\min_{\mathbf{z},\mathbf{t}} \quad & g(\mathbf{z}) + h(\mathbf{t}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{t} = \mathbf{c} \end{aligned} \tag{1.10}$$

To express convolutional sparse inference in this form, we perform the (somewhat unintuitive) substitution,

$$g(\mathbf{z}) = \frac{1}{2}||\mathbf{x} - \mathbf{d} * \mathbf{z}||_2^2 \qquad h(\mathbf{z}) = \beta||\mathbf{z}||_1 \tag{1.11}$$

$$\mathbf{A} = \mathbf{I} \qquad \mathbf{B} = -\mathbf{I} \qquad \mathbf{c} = 0 \tag{1.12}$$

to obtain,

$$\begin{aligned} \arg\min_{\mathbf{z},\mathbf{t}} \quad & \frac{1}{2}||\mathbf{x} - \mathbf{d} * \mathbf{z}||_2^2 + \beta||\mathbf{t}||_1 \\ \text{subject to} \quad & \mathbf{z} = \mathbf{t} \end{aligned} \tag{1.13}$$

By introducing a proxy $\mathbf{t}$, the loss function can be treated as a sum of functions of two independent variables, and with the addition of equality constraints, the minima of the new constrained objective is the same as the original unconstrained objective.

Whilst the individual functions may be easier to optimize, there is added complexity in enforcing the equality constraints. Taking the Lagrangian of the

augmented objective,

$$\mathbf{L}(\mathbf{z}, \mathbf{t}, \mathbf{u}) = g(\mathbf{z}) + h(\mathbf{t}) + \frac{\rho}{2}||\mathbf{z} - \mathbf{t} + \mathbf{u}||_2^2 \tag{1.14}$$

the solution is to minimize the primal variables, and maximize the dual variables,

$$\mathbf{z}^\star, \mathbf{t}^\star, \mathbf{u}^\star \;=\; \arg\max_{\mathbf{u}} \left(\arg\min_{\mathbf{z},\mathbf{t}} \mathbf{L}(\mathbf{z}, \mathbf{t}, \mathbf{u}) \right) \tag{1.15}$$

Optimizing in alternation (which accounts for the term *alternating direction*) yields a strategy for updating the variables involving a function of a single variable plus a proximal term binding all variables,

$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \;\; g(\mathbf{z}^k) + \frac{\rho}{2}||\mathbf{z}^k - (\mathbf{t} - \mathbf{u})||_2^2 \tag{1.16}$$

$$\mathbf{t}^{k+1} = \arg\min_{\mathbf{t}} \;\; h(\mathbf{t}^k) + \frac{\rho}{2}||\mathbf{t}^k - (\mathbf{z} + \mathbf{u})||_2^2 \tag{1.17}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \rho(\mathbf{z} - \mathbf{t}) \tag{1.18}$$

The intuition behind this strategy is to find a set of model parameters which are close to the regularization parameters, then visa versa, to find a set of regularization parameters which are close to the model parameters. The Lagrange variables impose a linear descent direction that force the two primal variables to equality over time.

For this strategy to be effective, the sum of the function g or h and an isotropic least-squares must be easy to solve.

Substituting the convolutional sparse inference objective into Equation 1.16,

$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \frac{1}{2}||\mathbf{x} - \mathbf{d} * \mathbf{z}^k||_2^2 + \frac{\rho}{2}||\mathbf{z}^k - (\mathbf{t} - \mathbf{u})||_2^2 \tag{1.19}$$

$$\mathbf{t}^{k+1} = \arg\min_{\mathbf{t}} \beta||\mathbf{t}^k||_1 + \frac{\rho}{2}||\mathbf{t}^k - (\mathbf{z} + \mathbf{u})||_2^2 \tag{1.20}$$

The $\mathbf{z}$ update takes the form of generalized Tikhonov regularization. In the $\mathbf{t}$ update, the L_1-regularizer can now be treated independently of the model-fitting term, and importantly the added proximal term is an isotropic Gaussian, so each pixel of $\mathbf{t}$ can be updated independently,

$$t^{k+1} = \arg\min_{t} \;\; \beta|t^k| + \frac{\rho}{2}(t^k - z - u)^2 \tag{1.21}$$

the solution to which is the soft thresholding operator,

$$t^\star = \mathbf{S}(z+u) = \text{sgn}\,(z+u) \cdot \max\left\{|z+u| - \frac{\beta}{\rho}, 0\right\} \tag{1.22}$$

In the ADMM, the minimizer $\mathbf{t}^\star$ is exactly sparse, whilst the minimizer $\mathbf{z}^\star$ is only close to sparse. If exact sparsity is a concern in the problem domain, $\mathbf{t}^\star$ should be retained at the point of convergence.

1.3.2 FISTA/Proximal Gradient

Proximal gradient methods are a close parallel to ADMMs. They generalize the problem of projecting a point onto a convex set, and often admit closed-form solutions. [70] present a thorough review of proximal algorithms and their relation to ADMMs. The convolutional Lasso problem introduces the splitting,

$$g(\mathbf{z}) = \frac{1}{2}||\mathbf{x} - \mathbf{d} * \mathbf{z}||_2^2 \qquad h(\mathbf{z}) = \beta||\mathbf{z}||_1 \tag{1.23}$$

with gradient and proximal operator,

$$\nabla g(\mathbf{z}) = \mathbf{D}^T(\mathbf{D}\mathbf{z} - \mathbf{x}) \qquad \text{prox}_h(\mathbf{z}) = \mathbf{S}(\mathbf{z}) \tag{1.24}$$

where $\mathbf{D}$ is an explicit convolutional matrix representation of $\mathbf{d}$, and $\mathbf{S}$ is the soft thresholding operator of Equation 1.22. In this case, the proximal algorithm is finding the closest minimizer to the convolutional least squares model fitting term that projects onto the L_1-ball (with radius proportional to β).

The FISTA algorithm of [8] presents an efficient update strategy for solving this problem by incorporating an optimal first-order method in the gradient computation (discussed in Section 1.4.2).

FISTA and ADMMs both have similar computational complexity, each requiring updates to a least-squares problem and evaluation of a soft thresholding operator. One disadvantage of FISTA is the requirement of gradient-based updates to the functional term. ADMMs, on the other hand, make a more general set of assumptions, requiring only that the objective value in each iteration is reduced. In cases where solving the objective is similar in complexity to evaluating the gradient, ADMMs may converge faster.

1.3.3 Iterative Optimization

So far we have neglected to show how convolution in Equation 1.11 and Equation 1.23 is actually performed.

The classical approach to convolution is to assume Dirichlet boundary conditions, *i.e.* that values outside the domain of consideration are zero. In such a case $\mathbf{x} \in \mathcal{R}^{P \times Q}$, $\mathbf{d} \in \mathcal{R}^{R \times S}$ and $\mathbf{z} \in \mathcal{R}^{P \times Q}$. Another approach is to take only the convolution between the fully overlapping portions of the signals – 'valid' convolution – which results in $\mathbf{z} \in \mathcal{R}^{(P+R-1)\times(Q+S-1)}$.

In both approaches, the signals *must* be convolved in the spatial domain, which is an $(PQRS)$ operation.

One way to alleviate the computational cost is to assume periodic extension of the signal, where convolution is then diagonalized by the Fourier transform,

$$\mathbf{d} * \mathbf{z} = \mathcal{F}^{-1} \left\{ \mathcal{F}(\mathbf{d}) \cdot \mathcal{F}(\mathbf{z}) \right\} \tag{1.25}$$

where $\mathbf{d} \in \mathcal{R}^{P \times Q}$ and $\mathbf{z} \in \mathcal{R}^{P \times Q}$ (see Section 1.4.4 for the correct method of padding $\mathbf{d}$ to size). This method of convolution has cost $(PQ \log(PQ))$.

Both ADMMs and FISTA can take advantage of iterative methods, by taking accelerated (proximal) gradient steps. In the next section we show how solving the entire system of equations in the Fourier domain is only slightly more complex than performing a single gradient step, and can lead to faster overall convergence of the ADMM. An important distinction between FISTA and ADMMs is that FISTA is constrained to use gradient updates – it cannot take advantage of direct solvers.

1.3.4 Direct Optimization

Given Dirichlet boundary assumptions, a filter kernel $\mathbf{d}$ can be represented (in 2D) as a block-Toeplitz-Toeplitz matrix such that $g(\mathbf{z})$ of Equation 1.23 becomes,

$$g(\mathbf{z}) = \frac{1}{2} ||\mathbf{x} - \mathbf{D}\mathbf{z}||_2^2 \tag{1.26}$$

where,

$$\mathbf{D} = [\mathbf{D}_1 \dots \mathbf{D}_N] \qquad \mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_N \end{bmatrix} \tag{1.27}$$

which is the canonical least-squares problem. There are a plethora of direct solvers for this problem, however unlike Toeplitz matrices, there are no known fast ($< (n^2)$) methods for inverting block-Toeplitz-Toeplitz matrices, so general purpose solvers must be used. This requires constructing the full $\mathbf{D}$. For a $P \times Q$ input, and N filters each of support $R \times S$, the matrix $\mathbf{D}$ will have $(NPQRS)$ non-zero values.

One way to alleviate the computational cost is to assume periodic extension of the signal, where convolution is then diagonalized by the Fourier transform.

This provides an efficient strategy for inverting the system,

$$g(\hat{\mathbf{z}}) = \frac{1}{2}||\hat{\mathbf{x}} - \hat{\mathbf{D}}\hat{\mathbf{z}}||_2^2 \tag{1.28}$$

where,

$$\hat{\mathbf{D}} = \left[\text{diag}(\hat{\mathbf{d}}_1) \dots \text{diag}(\hat{\mathbf{d}}_N)\right] \qquad \hat{\mathbf{z}} = \begin{bmatrix} \hat{\mathbf{z}}_1 \\ \vdots \\ \hat{\mathbf{z}}_N \end{bmatrix} \tag{1.29}$$

The equivalence between Equation 1.23 and Equation 1.28 relies upon Parseval's theorem,

$$\mathbf{x}^T\mathbf{x} = K\hat{\mathbf{x}}^T\hat{\mathbf{x}} \tag{1.30}$$

where K is a constant scaling factor between the domains. Since the L_2 norm is rotation invariant, the minimizer of Equation 1.28 is the Fourier transform of the minimizer of Equation 1.23.

The system of Equation 1.19 can be solved directly in an efficient manner in the Fourier domain by observing that each frequency band in a single image $\mathbf{x}$ is related only by the same frequency band in each of the filters and coefficients, *e.g.*,

$$\hat{\mathbf{x}}_{1,1} = \hat{\mathbf{d}}_{1,n}^T\hat{\mathbf{z}}_{n,1} \quad \forall\, n \in 1 \dots N \tag{1.31}$$

Note that although $\hat{\mathbf{d}}$ is a column vector, it is derived by taking a single frequency component across all filters. Finding the optima of $\hat{\mathbf{z}}$ across all n channels for frequency band 1 now involves

$$\hat{\mathbf{z}}_{n,1} = \left(\hat{\mathbf{d}}_{1,n}\hat{\mathbf{d}}_{1,n}^T + \rho\mathbf{I}_{n,1}\right)^{-1}\left(\hat{\mathbf{x}}_{1,1}\hat{\mathbf{d}}_{1,n} + \rho(\hat{\mathbf{t}}_{n,1} - \hat{\mathbf{u}}_{n,1})\right) \tag{1.32}$$

Note that $\hat{\mathbf{d}}_{1,n}\hat{\mathbf{d}}_{1,n}^T$ is a rank-1 matrix. Thus from the matrix inversion lemma,

$$\begin{aligned}\left(\hat{\mathbf{d}}_{1,n}\hat{\mathbf{d}}_{1,n}^T + \rho\mathbf{I}_{n,1}\right) &= \frac{1}{\rho}\left(\mathbf{I}_{n,1} - \frac{1}{\rho + \hat{\mathbf{d}}_{1,n}^T\hat{\mathbf{d}}_{1,n}}\right)\hat{\mathbf{d}}_{1,n}\hat{\mathbf{d}}_{1,n}^T \\ &= \frac{1}{\rho}(\mathbf{I}_{n,1} - K)\hat{\mathbf{d}}_{1,n}\hat{\mathbf{d}}_{1,n}^T \end{aligned} \tag{1.33}$$

where K is a scalar, since $\mathbf{d}_{1,n}^T\mathbf{d}_{1,n}$ is a scalar. The block of $\mathbf{z}_{n,1}$ can thus be solved by,

$$\hat{\mathbf{z}}_{n,1} = \frac{1}{\rho}(\mathbf{I}_{n,1} - K)\hat{\mathbf{d}}_{1,n}\hat{\mathbf{d}}_{1,n}^T\left(\hat{\mathbf{x}}_{1,1}\hat{\mathbf{d}}_{1,n} + \rho(\hat{\mathbf{t}}_{n,1} - \hat{\mathbf{u}}_{n,1})\right) \tag{1.34}$$

which is just a series of multiplications, so a solution can be found in (n^2) time – no inversion is required.

The assumption of periodic extension is sometimes not indicative of the structure observed in the signal of interest, and can cause artefacts along the boundaries. In such a case, a more sensible assumption is to assume Neumann boundary conditions, or *symmetric* reflection across the boundary. This assumption tends to minimize boundary distortion for small displacements. The resulting matrix can be diagonalised by the discrete cosine transform (DCT). There exists a generalization of Parseval's theorem that extends to the DCT, however some care must be taken with understanding the nuances between the 16 types of DCT. [62] provides a comprehensive exposé on the topic.

1.4 Solving for Filters

Solving for the filters $\mathbf{d}$ involves optimizing,

$$\begin{aligned}\arg\min_{\mathbf{d}} \quad & \frac{1}{2}||\mathbf{x} - \mathbf{d} * \mathbf{z}||_2^2 \\ \text{subject to} \quad & ||\mathbf{d}||_2^2 \leq 1\end{aligned} \tag{1.35}$$

This is a classical least squares objective with norm inequality constraints. Norm constraints on the bases are necessary since there always exists a linear transformation of $\mathbf{d}$ and $\mathbf{z}$ which keeps $(\mathbf{d} * \mathbf{z})$ unchanged whilst making $\mathbf{z}$ approach zero. Inequality constraints are sufficient since deflation of $\mathbf{z}$ will always cause $\mathbf{d}$ to lie on the constraint boundary $||\mathbf{d}||_2^2 = 1$ (whilst forming a convex set).

Equation 1.35 is a quadratically constrained quadratic program (QCQP), which is difficult to optimize in general. Each of the following methods relax this form in one way or another to make it more tractable to solve.

The convolutional form of the least squares term also poses some challenges for optimization, since (i) the filter has smaller support than the image it is being convolved with, and (ii) forming an explicit multiplication between the filters and a convolutional matrix form of the images is prohibitively expensive (as per Section 1.3.4).

In the case of multiple images and filters,

$$\begin{aligned} \arg\min_{\mathbf{d}} \quad & \frac{1}{2}||\sum_{i=1}^{M}\left(\mathbf{x}_i - \sum_{j=1}^{N}(\mathbf{d}_j * \mathbf{z}_{i,j})\right)||_2^2 \\ \text{subject to} \quad & ||\mathbf{d}_j||_2^2 \leq 1 \;\; \forall\, j \in 1 \ldots N \end{aligned} \tag{1.36}$$

one can see that all filters are jointly involved in reconstructing the inputs, and so must be updated jointly.

1.4.1 Gradient Descent

The gradient of the objective is given by,

$$\nabla f_i = \left(\sum_{j=1}^{N} \mathbf{d}_j * \sum_{k=1}^{M}(\mathbf{z}_{i,j} \star \mathbf{z}_{i,k})\right) - \sum_{k=1}^{M}(\mathbf{x}_i \star \mathbf{z}_{i,j}) \tag{1.37}$$

This involves collecting the correlation statistics across the coefficient maps and images. Since the filters are of smaller support than the coefficient maps and images, we collect only "valid" statistics, or regions that don't incur boundary effects. In the autocorrelation of $\mathbf{z}$, one of the arguments must be zero padded to the appropriate size.

Given the gradient direction, updating the the filters involves,

$$\mathbf{d}^{k+1} \;=\; \mathbf{d}^k - t\nabla f \tag{1.38}$$

Computing the step size, t, can be done either via line search or by solving a 1D optimization problem which minimizes the reconstruction error in the gradient direction,

$$\arg\min_{t} \; ||\mathbf{x} - \underbrace{\left(\mathbf{d}^k - t\nabla f\right)}_{\mathbf{d}^{k+1}} * \mathbf{z}||_2^2 \tag{1.39}$$

the closed-form solution to which is,

$$t \;=\; \frac{(\mathbf{x} - \mathbf{d} * \mathbf{z})^T(\nabla f * \mathbf{z})}{(\nabla f * \mathbf{z})^T(\nabla f * \mathbf{z})} \tag{1.40}$$

and involves evaluating only $2N$ convolutions (N if the $\mathbf{x} - \mathbf{d} * \mathbf{z}$ term has been previously computed as part of a stopping criteria, *etc.*). In the case of a large number of inputs M, stochastic gradient descent is typically used [59].

After each gradient step, if the new iterate exists outside the L_2 unit ball, the result is projected back onto the ball. This is not strictly the correct way to enforce the norm-constraints, however it tends to work without side-effects in practice. The ADMM method we present (Section 1.4.3) enjoys the property that this projection solves for the norm constraints exactly.

1.4.2 Nesterov's Accelerated Gradient Descent

Prolific mathematician Yurii Nesterov introduced an optimal[1] first-order method for solving smooth convex functions [65]. Convolutional least squares objectives require a straightforward application of accelerated proximal gradient (APG).

Without laboring on the details introduced by Nesterov, the method involves iteratively placing an isotropic quadratic tangent to the current gradient direction, then shifting the iterate to the minima of the quadratic. The curvature of the quadratic is computed by estimating the Lipschitz smoothness of the objective.

Checking for Lipschitz smoothness feasibility using backtracking requires two projections per iteration. This can be costly since each projection involves multiple convolutions. In practice we find this method no faster than regular gradient descent with optimal step-size calculation.

[1] Optimal in the sense that it has a worst-case convergence rate that cannot be improved whilst remaining first-order.

1.4.3 ADMM Partitioning

As per Section 1.3.1, treating convolution in the Fourier domain can lead to efficient direct optimization. Unlike solving for the coefficients, however, the learned filters must be constrained to be small support, and there is no way to do this explicitly via Fourier convolution.

An approach to handling this is via ADMMs again,

$$\begin{aligned} \arg\min_{\mathbf{d},\mathbf{s}} \quad & \frac{1}{2}||\hat{\mathbf{x}} - \hat{\mathbf{Z}}\hat{\mathbf{s}}||_2^2 \\ \text{subject to} \quad & \mathbf{d}_j = \mathbf{\Phi}^T \hat{\mathbf{s}}_j \quad \forall\, j \in 1 \dots N \\ & ||\mathbf{d}_j||_2^2 \leq 1 \quad \forall\, j \in 1 \dots N \end{aligned} \tag{1.41}$$

where,

$$\begin{aligned} \hat{\mathbf{x}} &= \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \vdots \\ \hat{\mathbf{x}}_M \end{bmatrix} \qquad \hat{\mathbf{d}} = \begin{bmatrix} \hat{\mathbf{d}}_1 \\ \vdots \\ \hat{\mathbf{d}}_N \end{bmatrix} \\ \hat{\mathbf{Z}} &= \begin{bmatrix} \text{diag}(\hat{\mathbf{z}}_{1,1}) & \dots & \text{diag}(\hat{\mathbf{z}}_{1,N}) \\ \vdots & \ddots & \\ \text{diag}(\hat{\mathbf{z}}_{M,1}) & & \text{diag}(\hat{\mathbf{z}}_{M,N}) \end{bmatrix} \end{aligned} \tag{1.42}$$

and $\mathbf{\Phi}$ is a submatrix of the Fourier matrix that corresponds to a small spatial support transform.

Intuitively, we are trying to learn a set of filters $\hat{\mathbf{s}}$ that minimize reconstruction error in the Fourier domain and are small support in the spatial domain.

Taking the augmented Lagrangian of the objective and optimizing over $\mathbf{d}$ and $\mathbf{s}$ in alternation yields the update strategy,

$$\hat{\mathbf{s}}^{k+1} = \arg\min_{\hat{\mathbf{s}}} \quad ||\hat{\mathbf{x}} - \hat{\mathbf{Z}}\hat{\mathbf{s}}^k||_2^2 + \frac{\rho}{2}||\hat{\mathbf{s}}^k - (\mathbf{\Phi}\mathbf{d} - \hat{\mathbf{u}})||_2^2 \tag{1.43}$$

$$\begin{aligned} \mathbf{d}^{k+1} = \arg\min_{\mathbf{d}} \quad & ||\mathbf{\Phi}\mathbf{d}^k - (\hat{\mathbf{s}} + \hat{\mathbf{u}})||_2^2 \\ \text{subject to} \quad & ||\mathbf{d}_j||_2^2 \leq 1 \quad \forall\, j \in 1 \dots N \end{aligned} \tag{1.44}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \rho(\mathbf{\Phi}\mathbf{d} - \hat{\mathbf{s}}) \tag{1.45}$$

In a similar fashion to Equation 1.32, each frequency component in $\hat{\mathbf{s}}$ jointly across all filters can be solved for independently via a variable reordering to produce PQ dense systems of equations.

Solving for $\mathbf{d}$ appears more involved, however the unconstrained loss func-

tion can be minimized in closed-form,

$$\begin{aligned}\mathbf{d}^\star &= \mathbf{d}^T\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{d} - 2\mathbf{d}^T\mathbf{\Phi}^T(\hat{\mathbf{s}} - \hat{\mathbf{u}}) + c \\ &= \mathbf{\Phi}^T(\hat{\mathbf{s}} - \hat{\mathbf{u}})\end{aligned} \tag{1.46}$$

since $\mathbf{\Phi}$ is an orthonormal matrix, and thus $\mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{I}$. Further, the matrix multiplication can instead be replaced by the inverse Fourier transform, followed by a selection operator $\mathcal{M}$ which keeps only the small support region,

$$\mathbf{d}^\star = \mathcal{M}\left(\mathcal{F}^{-1}\{\hat{\mathbf{s}} - \hat{\mathbf{u}}\}\right) \tag{1.47}$$

Handling the inequality constraints is now trivial. Since $\mathbf{\Phi}$ is orthonormal (implying an isotropic regression problem), projecting the optimal solution to the unconstrained problem onto the L_2 ball,

$$\mathbf{d}^\star = \begin{cases} ||\mathbf{d}_k^\star||_2^{-2}\mathbf{d}_k^\star, & \text{if } ||\mathbf{d}_k^\star||_2^2 \geq 1 \\ \mathbf{d}_k^\star, & \text{otherwise} \end{cases} \tag{1.48}$$

is *equivalent* to solving the constrained problem.

1.4.4 Small Support Convolution in the Fourier Domain

In order to convolve two signals in the Fourier domain, their lengths must commute. This involves padding the shorter signal to the length of the longer. Some care must be taken to avoid introducing phase shifts into the response, however. Given a 2D filter $\mathbf{z} \in \mathcal{R}^{P,Q}$, we can partition it into 4 blocks,

$$\mathbf{z} = \begin{vmatrix} \mathbf{z}_{1,1} & \mathbf{z}_{1,2} \\ \mathbf{z}_{2,1} & \mathbf{z}_{2,2} \end{vmatrix} \tag{1.49}$$

where, in the case of odd-sized filters, the blocks are partitioned *above and to the left* of the central point. Given a 2D image $\mathbf{x} \in \mathcal{R}^{M,N}$, the padded representation $\mathbf{z}^\star \in \mathcal{R}^{M,N}$ can thus be formed as,

$$\mathbf{z} = \begin{vmatrix} \mathbf{z}_{2,2} & & & & \mathbf{z}_{2,1} \\ & \ddots & & & \\ & \dots & \mathbf{0} & \dots & \\ & & & \ddots & \\ \mathbf{z}_{1,2} & & & & \mathbf{z}_{1,1} \end{vmatrix} \tag{1.50}$$

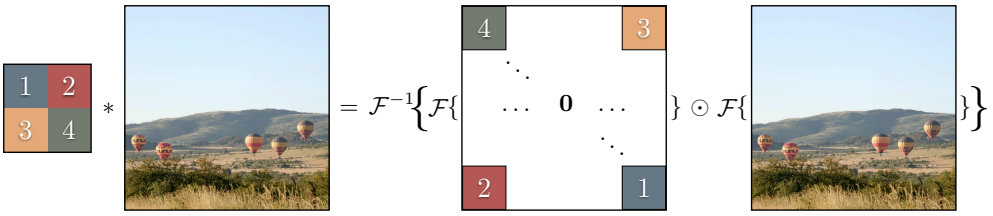


Figure 1.4: Swapping quadrants and padding the filter to the size of the image permits convolution in the Fourier Domain

This transform is illustrated in Figure 1.4. For a comprehensive guide to Fourier domain transforms and identities, including appropriate handling of boundary effects and padding, see [69].

1.5 Stopping Criteria

For the gradient-based algorithms – gradient descent, APG and FISTA – a sufficient stopping criteria is to threshold the residual between two iterates,

$$||\mathbf{x}^{k+1} - \mathbf{x}^{k}||_2^2 \leq \epsilon \tag{1.51}$$

where $\mathbf{x}$ is the variable being minimized.

Estimating convergence of the ADMM-based methods is more involved, including deviation from primal feasibility,

$$||\mathbf{d} - \mathbf{s}||_2^2 \leq \epsilon \,, \qquad ||\mathbf{z} - \mathbf{t}||_2^2 \leq \epsilon \tag{1.52}$$

and dual feasibility,

$$||\mathbf{u}^{k+1} - \mathbf{u}^{k}||_2^2 \leq \epsilon \tag{1.53}$$

However, in practice it is usually sufficient to measure only primal feasibility, since if the iterates have reached primal feasibility, dual feasibility is unlikely to improve (this is doubly true when using a strategy for increasing ρ).

1.6 Applications

1.6.1 Example 1 - Image and Video Compression

Many image and video coding algorithms such as JPEG [93] and H.264 [83] discretize each image into blocks which are transformed, quantized and coded independently.

Using convolutional sparse coding, an entire image $\mathbf{x}$ can be coded onto a basis $\mathbf{d}$ with sparse reconstruction coefficients $\mathbf{z}$. Quality and size can be controlled via the β parameter. The basis can either be specific to the image, or a generic basis (such as Gabor) which is part of the coding spec and need not be transferred with the image data.

Since the coefficients of $\mathbf{z}$ are exactly sparse by virtue of the soft-thresholding operator, the representation can make effective use of run-length and Huffman entropy coding techniques. [2]

To reconstruct the image $\mathbf{x}_r$, the decoder simply convolves the bases with the transmitted coefficients,

$$\mathbf{x}_r = \sum_{j=1}^{N} \mathbf{d}_j * \mathbf{z}_j \tag{1.54}$$

This matches the media model well: media is consumed more frequently than it is created, so encoding can be costly (in this case an inverse inference problem) but decoding should be fast – convolutional primitives are hardware-accelerated on almost all modern chipsets.

1.6.2 Example 2 - A basis for Natural Images

The receptive fields of complex cells in the mammalian primary visual cortex can be characterized as being spatially localized, oriented and bandpass. [67] hypothesized that such fields could arise spontaneously in an unsupervised strategy for maximizing the sparsity of the representation. Sparsity can be interpreted biologically as a metabolic constraint - firing only a few neurons in response to a stimulus is clearly more energy efficient than firing a large number.

[2] JPEG also uses run-length coding but its efficiency is a function of the quantization artefacts.

[67] use traditional patch-based sparse coding to solve for a set of basis functions. The famous result is that the learned basis functions resemble Gabor filters. However, a large number of the bases learned are translations of others – an artefact of sampling and treating each image patch independently.

It is well-understood that the statistics of natural images are translation invariant [40], *i.e.* the covariance of natural images depends only on the distance,

$$\Sigma\left(\mathbf{I}(x,y),\ \mathbf{I}(x',y')\right) = f\left(\mathbf{I}(x-x',y-y')\right) \tag{1.55}$$

Thus it is sufficient to code natural images in a manner that does not depend on exact position of the stimulus. Convolutional sparse coding permits this, and as a result produces a more varied range of basis elements than simple Gabor filters when coding natural images. The sparsity pattern of convolutional coefficients also has a mapping onto the receptive fields of active neurons in V1.[3]

1.6.3 Example 3 - Structure from Motion

Trajectory basis Non-Rigid Structure from Motion (NRSfM) refers to the process of reconstructing the motion of 3D points of a non-rigid object from only their 2D projected trajectories.

Reconstruction relies on two inherently conflicting factors: (i) the condition of the composed camera and trajectory basis matrix, and (ii) whether the trajectory basis has enough degrees of freedom to model the 3D point trajectory. Typically, (i) is improved with a low-rank basis, and (ii) is improved with a higher-rank basis.

The Discrete Cosine Transform (DCT) basis has traditionally been used as a generic basis for encoding motion trajectories, however choosing the correct rank has been a difficult problem. [106] proposed the use of convolutional sparse coding to learn a compact basis that could model the trajectories taken from a corpus of training data.

[3] Unlike many higher regions within the visual cortex, V1 is retinotopic – the spatial location of stimulus in the visual world is highly correlated with the spatial location of active neurons.

Learning the basis proceeds as per usual,

$$\begin{aligned} \arg\min_{\mathbf{d},\mathbf{z}} \quad & \frac{1}{2}\sum_{i=1}^{M}||\big(\mathbf{x}_i - \sum_{j=1}^{N}(\mathbf{d}_j * \mathbf{z}_{i,j})\big)||_2^2 \\ & + \beta\sum_{i=1}^{M}\sum_{j=1}^{N}||\mathbf{z}_{i,j}||_1 \\ \text{subject to} \quad & ||\mathbf{d}_j||_2^2 \leq 1 \;\; \forall\, j \in 1\ldots N \end{aligned} \tag{1.56}$$

where each $\mathbf{x}_i$ is a 1D trajectory of arbitrary length, $\mathbf{d}$ the trajectory basis being learned, and $\mathbf{z}$ the sparse reconstruction coefficients.

Given the convolutional trajectory basis $\mathbf{d}$, reconstructing the sparse coefficients for the $3D$ trajectory from $2D$ observations involves,

$$\begin{aligned} \mathbf{z}^* = \arg\min_{\mathbf{z}} \quad & ||\mathbf{z}||_1 \\ \text{subject to } \underbrace{\mathbf{Q}\mathbf{x}}_{\mathbf{u}} = \mathbf{Q}\sum_{j=1}^{N}\mathbf{d}_j * \mathbf{z}_j & \end{aligned} \tag{1.57}$$

where $\mathbf{u}$ are the $2D$ observations of the $3D$ points $\mathbf{x}$ that have been imaged by the camera matrices $\mathbf{Q}$, one for each frame in the trajectory,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & & \\ & \ddots & \\ & & \mathbf{Q}_F \end{bmatrix} \tag{1.58}$$

In single view reconstruction, back-projection is typically enforced as a constraint, and the objective is to minimize the number of non-zero coefficients in the reconstructed $3D$ trajectory that satisfy this constraint.

A convolutional sparse coded basis produces less $3D$ reconstruction error than previously explored bases, including one learned from patch-based sparse coding, and a generic DCT basis. This illustrates convolutional sparse coding's ability to learn low rank structure from misaligned trajectories stemming from the same underlying dynamics (*e.g.* articulated human motion).

1.6.4 Example 4 - Mid-level Generative Parts

Zeiler *et al.* show how a cascade of convolutional sparse coders can be used to build robust, unsupervised mid-level representations, beyond the edge primi-

tives of Section 1.6.2.

The convolutional sparse coder at each level of the hierarchy can be defined as,

$$C_l(\mathbf{d}^l, \mathbf{z}^l) = \frac{1}{2}\sum_{i=1}^{M} || \underbrace{f_s(\mathbf{z}_i^{l-1})}_{\mathbf{x}_i} - \sum_{j=1}^{M}(\mathbf{d}_j^l * \mathbf{z}_{i,j}^l)||_2^2 + \beta \sum_{i=1}^{M}\sum_{j=1}^{N} ||\mathbf{z}_{i,j}^l||_1 \tag{1.59}$$

where the extra superscripts on each element indicate the layer to which they are native.

The inputs $\mathbf{x}$ to each layer are the sparse coefficients of the previous layer, $\mathbf{z}^{l-1}$ after being passed through a pooling/subsampling operation $f_s(\cdot)$. For the first layer, $\mathbf{z}^{l-1} = \mathbf{x}$, *i.e.* the input image.

The idea behind this coding strategy is that structure within the signal is progressively gathered at a higher and higher level, initially with edge primitives, then mergers between these primitives into line-segments, and eventually into recurrent object parts.

Layers of convolutional sparse coders have also been used to produce high quality latent representations for convolutional neural networks [50, 44, 19], though fully-supervised back-propagation across layers has become popular more recently [47].

1.6.5 Example 5 - Single Image Super-Resolution

Single-Image Super Resolution (SISR) is the process of reconstructing a high-resolution image from an observed low-resolution image. SISR can be cast as the inverse problem,

$$\mathbf{y} = \mathbf{DBx} \tag{1.60}$$

where $\mathbf{x}$ is the latent high-resolution image that we wish to recover, $\mathbf{B}$ is an anti-aliasing filter, $\mathbf{D}$ is a downsampling matrix and $\mathbf{y}$ is the observed low-resolution image. The system is underdetermined, so there exist infinitely many solutions to $\mathbf{x}$. A strategy for performing SISR is via a straightforward convolutional

extension of [98],

$$\begin{aligned}\arg\min_{\mathbf{d}_L,\mathbf{d}_H,\mathbf{z}} & \sum_{i=1}^{M}||\mathbf{DBx}_i - \sum_{j=1}^{N}(\mathbf{d}_{L,j} * \mathbf{Dz}_{i,j})||_2^2 \\ & + ||\mathbf{x}_i - \sum_{j=1}^{N}(\mathbf{d}_{H,j} * \mathbf{z}_{i,j})||_2^2 \\ & + \beta \sum_{i=1}^{M}\sum_{j=1}^{M}||\mathbf{z}_{i,j}||_1 \\ \text{subject to} & ||\mathbf{d}_{L,j}||_2^2 \leq 1 \quad \forall\, j \in 1 \ldots M \\ & ||\mathbf{d}_{H,j}||_2^2 \leq 1 \quad \forall\, j \in 1 \ldots M \end{aligned} \tag{1.61}$$

where $\mathbf{x}_i$ and $\mathbf{DBx}_i$ are a high-resolution and derived low-resolution training pair, $\mathbf{D}$ is the downsampling filter as before and $\mathbf{z}$ are a common set of coefficients that tie the two representations together. The dictionaries $\mathbf{d}_L$ and $\mathbf{d}_H$ learn a mapping between low- and high-resolution image features.

Given a new low-resolution input image $\mathbf{x}_L$, the sparse coefficients are first inferred with respect to the low-resolution basis,

$$\mathbf{z}^\star = \arg\min_{\mathbf{z}} ||\mathbf{x}_L - \sum_{j=1}^{N}(\mathbf{d}_{L,j} * \mathbf{Dz}_j)||_2^2 + \beta||\mathbf{z}||_1 \tag{1.62}$$

and then convolved with the high-resolution basis to reconstruct the high-resolution image,

$$\mathbf{x}_H = \sum_{j=1}^{N}(\mathbf{d}_{H,j} * \mathbf{z}_j) \tag{1.63}$$

1.6.6 Example 6 - Visualizing Object Detection Features

[90] presented a method for visualizing HOG features via the inverse mapping,

$$\phi^{-1}(\mathbf{y}) = \arg\min_{\mathbf{x}} ||\phi(\mathbf{x}) - \mathbf{y}||_2^2 \tag{1.64}$$

where $\mathbf{x}$ is the image to recover, and $\mathbf{y} = \phi(\mathbf{x})$ is the mapping of the image into HOG space. Direct optimization of this objective is difficult, since it is highly nonlinear through the HOG operator $\phi()$, *i.e.* multiple distinct images can map to the same HOG representation.

One possible approach to approximating this objective is through paired dictionary learning, in a similar manner to Section 1.6.5.

Given an image $\mathbf{x}$ and its representation $\mathbf{y} = \phi(\mathbf{x})$ in the HOG domain, we wish to find two basis sets, $\mathbf{d_I}$ in the image domain and $\mathbf{d}_\phi$ in the HOG domain, and a common set of sparse reconstruction coefficients $\mathbf{z}$, such that,

$$\mathbf{x} = \sum_{j=1}^{N} (\mathbf{d}_{\mathbf{I},j} * \mathbf{z}_j) \qquad \mathbf{y} = \sum_{j=1}^{N} (\mathbf{d}_{\phi,j} * \mathbf{z}_j) \tag{1.65}$$

Intuitively, the common reconstruction coefficients force the basis sets to represent the same appearance information albeit in different domains. The basis pairs thus provide *a* mapping between the domains.

Optimizing this objective is a straightforward extension of the patch based sparse coding used by [90],

$$\begin{aligned} \arg\min_{\mathbf{d_I},\mathbf{d}_\phi,\mathbf{z}} \; & \sum_{i=1}^{M} ||\mathbf{x}_i - \sum_{j=1}^{N} (\mathbf{d}_{\mathbf{I},j} * \mathbf{z}_{i,j})||_2^2 \\ & + ||\phi(\mathbf{x}_i) - \sum_{j=1}^{N} (\mathbf{d}_{\phi,j} * \mathbf{z}_{i,j})||_2^2 \\ & + \beta \sum_{i=1}^{M} \sum_{j=1}^{M} ||\mathbf{z}_{i,j}||_1 \\ \text{subject to} \; & ||\mathbf{d}_{\mathbf{I},j}||_2^2 \leq 1 \quad \forall\, j \in 1 \ldots M \\ & ||\mathbf{d}_{\phi,j}||_2^2 \leq 1 \quad \forall\, j \in 1 \ldots M \end{aligned} \tag{1.66}$$

Because we are optimizing over entire images rather than independently sampled patches, the bases learned will (i) produce a more unique mapping between pixel features and HOG features (since translations of features are not represented), and (ii) be more expressive for any given basis set size as a direct result of (i).

Image-scale optimization also reduces blocking artefacts in the image reconstructions, leading to more faithful/plausible representations, with potentially finer-grained detail.

1.7 Discussion

A majority of this chapter was written in 2013, immediately following the original AlexNet paper. At that time, significant work was still focussed on unsupervised training of convolutional networks. That is, networks that learned generic high-level representations of images, independent of task. Indeed, this work was heavily influenced by two works in that domain: *Learning Convolutional Feature Hierarchies for Visual Recognition*, and *Deconvolutional Networks*. Since convolutional network literature has largely transitioned to fully supervised methods, this convolutional sparse coding work is largely defunct. Interesting piexelwise regression-type problems that can be solved by convolutional sparse coding (outlined in the Examples section) can be solved better with Hourglass networks.

Whilst sparsity still plays an important regularization role in convolutional networks – dropout was a significant contribution in the original AlexNet paper – it is more stochastic and less structural than what appears in the convolutional sparse coding problem.

Sparse and convolutional constraints will likely still exist in problems that require exact and online optimization, however not in the form presented in this thesis, but rather as part of a specific objective (such as NRSfM).

1.8 Conclusion

This chapter has focussed on the learning of feature representations in an unsupervised manner from natural imagery. While many sparse coding applications treat patches of images separately, convolution is a natural way of embedding invariance to geometry. This comes at a significant computational cost, however. We introduced a method for solving the convolutional sparse coding problem efficiently, by decomposing the original problem into subproblems using a strategy based on ADMMs, and posing the convolution operations in the Fourier domain.

In the next chapter, we focus on complete image representations inspired by the primary visual cortex (V1). We show how the process of coding, rectification and pooling can be represented as a margin weighting in a maximum margin classifier learned over pairwise interactions of pixels. Preserving only local interactions, we further show that the V1 prior can be replaced with a strategy for sampling geometric perturbations of the training set.

Chapter 2

Locality and Capacity

IMAGE REPRESENTATIONS derived from simplified models of the primary visual cortex (V1), such as HOG and SIFT, elicit good performance in a myriad of visual perception tasks. Image representations and classifiers are intrinsically related, since complexity in one can be traded for simplicity in the other. The choice of representation imparts two properties on a classifier: prior and capacity. Understanding how the classifier is influenced by these properties is central to improving both classification techniques, and the types of priors to encode for visual perception tasks.

It is well understood that for visual recognition tasks, a nearest-neighbour classifier is optimal given infinite training data. Classification reduces to a simple lookup operation that indexes into the perfect world knowledge. Of course, this ignores the very real time and space constraints that actual recognition systems must deal with. A goal of visual recognition, therefore, is to learn from imperfect and finite amounts of training data to generalize to new and unfamiliar scenes.

The role of features has been studied largely in isolation to the learning architecture used for classification. One criticism of ignoring the learning strategy when studying features is that structure unimportant to the classifier may be preserved, resulting in (i) additional computational burden, and (ii) ambiguity in representation.

In this chapter we focus on understanding how feature representations and linear classifiers interact. In particular, we are interested in the statistical properties of natural imagery which should be preserved to maximize classification performance. As discussed in Chapter 1, the mammalian visual system has found a particularly efficient representation of the visual world, but many of the computational objectives or principles that brought this about are still

unknown.

Vast bodies of work have been devoted to understanding the mammalian visual system, in particular the primary visual cortex, V1. The canonical V1 model was first proposed by Hubel and Wiesel in their seminal 1962 work on the cat's visual cortex. By probing cortical neurons with electrodes, the cells fired only when a bar of light of a particular orientation was observed in their receptive field.

Image representations derived from simplified models of the primary visual cortex are all built on the notion that local object appearance can be well categorized by the distribution of local features, without precise knowledge of their spatial location. They typically involve three types of operations: (i) convolution with a bank of filters to produce a set of activations, (ii) non-linear rectification, and (iii) pooling of the responses over a small spatial region. This is the basis for many features in computer vision including HOG, SIFT, LBP and convolutional networks (which compose layers of these simple operations).

We focus primarily on the HOG representation used in conjunction with a linear SVM. This combination is interesting because it makes effective use of small amounts of training data, is fast to apply, and is still the foundation of many time sensitive applications.

This chapter addresses the following concepts:

- We show that a particular class of V1-inspired features can be rewritten as a linear function of the Kronecker expansion of image pixels. This linear transform can be viewed as a data-independent matrix which induces a weighted margin in max-margin learning

- We demonstrate that reinterpreting the role of V1-inspired features as a weighted margin reveals some valuable insights into (i) the uniqueness of the filters commonly used in these architectures, and (ii) the capacity of a linear SVM using V1-inspired features tending towards a quadratic kernel SVM.

- We show that an equivalent classifier can be learned by replacing the feature representation with a quadratic kernel classifier learned on pixels alone, with a parametric model for creating synthetic data

- We show that a *local* quadratic classifier - one that preserves only local pixel interactions, performs equivalently to one preserving all pairwise interactions.

2.1 Related Work

Ashraf *et al.* [2] originally explored the link between feature extraction and a weighted margin for visual classification tasks. By restricting their scope to linear features, they view filtering as a weighted margin on the data in the Fourier domain. We instead explore an inherently nonlinear embedding, more akin to current models of early biological vision. Due to the high dimensionality of the resulting problem (not encountered by Ashraf *et al.* by virtue of the convolution theorem), we seek to explicitly represent the feature maps in a lower dimensional space.

Vedaldi and Zisserman [89] and Bo *et al.* [11] both proposed methods for explicitly representing kernels so lower dimensional approximations can be found, independent of data. The appeal of both approaches is the speedup in training and evaluation time that can be enjoyed by learning a linear rather than kernel SVM. Vedaldi considered the case of approximately representing the implicit feature associated with additive kernels (*i.e.* kernels useful for matching histograms) whilst Bo considered the case of incorporating preprocessed oriented edge energies, along with spatial position and colour directly into the kernel function. Our method, by contrast, relates the raw image pixel intensities directly with the feature pipeline. By having this direct relationship we can gain fundamental insights into the importance of particular architectures and redundancies in V1-inspired features to actual classification performance within a linear SVM.

2.2 V1-Inspired Features

Handcrafted feature representations such as HOG and SIFT, and learned architectures such as convolutional networks, crudely approximate the function of V1 complex cells. A generalized description of these representations involves edge orientation detection, nonlinear rectification, contrast normalization to remove local photometric variation, and pooling to introduce tolerance to geometric variation. This type of representation has proven particularly successful at being tolerant to non-rigid changes in object geometry whilst maintaining high selectivity [23]. To begin our analysis, we compose a canonical form of this representation.

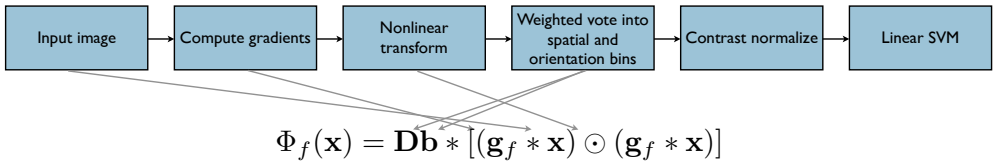


Figure 2.1: An illustration of the HOG feature extraction process and how each component maps to our reformulation. Gradient computation is achieved through convolution with a bank of oriented edge filters. The nonlinear transform is the pointwise squaring of the gradient responses which removes sensitivity to edge contrast and increases edge bandwidth. Histogramming can be expressed as blurring with a box filter followed by downsampling.

2.2.1 Canonical Form

Given a vectorized input image of intensities $\mathbf{x} \in \mathcal{R}^D$, the representation can be computed via convolution with a bank of oriented edge filters, $\{\mathbf{g}_f\}_{f=1}^F$, followed by rectification with a pointwise quadratic function expressed as a Hadamard product ($\odot$), and spatial sum pooling with a constant or Gaussian filter $\mathbf{b}$ with optional downsampling with a decimation matrix. The feature map $\Phi(\mathbf{x})$ can thus be expressed as,

$$\Phi(\mathbf{x}) = [\Phi_1(\mathbf{x}), \Phi_2(\mathbf{x}), \ldots, \Phi_F(\mathbf{x})]^T \tag{2.1}$$

where,

$$\Phi_f(\mathbf{x}) = \mathbf{b} * [(\mathbf{g}_f * \mathbf{x}) \odot (\mathbf{h}_f * \mathbf{x})] \, . \tag{2.2}$$

This particular architecture has been termed "convolutional square pooling" and has shown good performance across a range of tasks [9]. Variations on this feature pipeline have been advocated in the literature, such as the use of max rather than sum pooling, sigmoidal or hinge activation functions, and edge orientation approximations using trigonometric functions or learned filters.

Our choices for the specific canonical form described throughout this chapter are that it is similar in philosophy to other variants, has greater flexibility in manipulation which will become apparent in our later reformulation, has proponents in convolutional network literature[9, 43] and a good basis in statistical models of the primary visual cortex [41].

In Chapter 1, we discussed the orientation and frequency selectivity of visual cortical neurons, and how it can arise from a strategy for sparsely encoding visual data. Convolutional networks also have a tendency to learn similar orientation selective filters in their early layers. In this chapter, we construct the

filter banks $\mathbf{g}, \mathbf{h}$ from Gabor filters to mimic these observed properties.

2.2.2 Kronecker Form

Manipulation of the form in Equation 2.2 is difficult due to the limited properties of the Hadamard product ($\odot$). By defining a relation between the Hadamard and Kronecker product ($\otimes$) however, we can exploit properties of the latter.

Theorem 2.2.1 *The Hadamard product between any two equal size vectors* $\mathbf{x}_i \in \mathcal{R}^D$ *and* $\mathbf{x}_j \in \mathcal{R}^D$ *can be written as,*

$$\mathbf{x}_i \odot \mathbf{x}_j = \mathbf{M}(\mathbf{x}_i \otimes \mathbf{x}_j) \tag{2.3}$$

such that $\mathbf{M} \in \mathcal{R}^{D \times D^2}$. We can explicitly define $\mathbf{M}$ as,

$$\mathbf{M} = \begin{bmatrix} \mathrm{e}_1^T \otimes \mathrm{e}_1^T \\ \vdots \\ \mathrm{e}_D^T \otimes \mathrm{e}_D^T \end{bmatrix} \tag{2.4}$$

given that $\mathrm{e}_i \in \mathcal{R}^D$ is a vector of zeros with 1 at the i-th element. Intuitively, Equation 2.3 forms *all* pairwise products of elements in $\mathbf{x}_i$ and $\mathbf{x}_j$, then preserves only the interactions whose indices are equal. As a result, $\mathbf{x}_i \otimes \mathbf{x}_j$ is highly redundant, and $\mathbf{M}$ is highly sparse.

Replacing 2D convolution operations (*e.g.* $\mathbf{h} * \mathbf{x}$) with Toeplitz convolution matrices (*e.g.* $\mathbf{Hx}$) and applying Theorem 2.2.1 to Equation 2.2, the response to a single filter can be written as,

$$\begin{aligned} \Phi_f(\mathbf{x}) &= \mathbf{BM}[(\mathbf{G}_f\mathbf{x}) \otimes (\mathbf{H}_f\mathbf{x})] \\ &= \mathbf{BM}(\mathbf{G}_f \otimes \mathbf{H}_f)(\mathbf{x} \otimes \mathbf{x}) \,. \end{aligned} \tag{2.5}$$

The full response to a bank of filters can be written as,

$$\Phi(\mathbf{x}) = \mathbf{L}(\mathbf{x} \otimes \mathbf{x}) \tag{2.6}$$

where,

$$\mathbf{L} = \begin{bmatrix} \mathbf{BM}(\mathbf{G}_1 \otimes \mathbf{H}_1) \\ \vdots \\ \mathbf{BM}(\mathbf{G}_F \otimes \mathbf{H}_F) \end{bmatrix} \,. \tag{2.7}$$

For two V1-inspired feature maps $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$, the kernel is defined as the inner product of the maps,

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \ . \tag{2.8}$$

Since the feature maps have a closed form expression, the kernel can be written explicitly as,

$$\begin{aligned} \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) &= (\mathbf{x}_i \otimes \mathbf{x}_i)^T \mathbf{L}^T \mathbf{L} (\mathbf{x}_j \otimes \mathbf{x}_j) & (2.9) \\ &= (\mathbf{x}_i \otimes \mathbf{x}_i)^T \mathbf{S} (\mathbf{x}_j \otimes \mathbf{x}_j) & (2.10) \end{aligned}$$

where $\mathbf{L} \in \mathcal{R}^{DF \times D^2}$ implies that the rank of $\mathbf{S}$ is at most DF. Thus after some manipulation, the form of V1-inspired features can be rearranged with the filter and data terms isolated. This suggests that the kernel is only dependent on the *joint* response from the filters and blur kernels, and that the weighting matrix $\mathbf{S}$ can be completely precomputed in the absence of data.

2.3 Computational Efficiency

Whilst $\mathbf{S}$ is rank deficient, its high dimensionality (*i.e.* $D^2 \times D^2$) makes it infeasible to work with directly. In practice, can find a matrix of rank $K \ll DF$ that makes a good approximation to $\mathbf{S}$ whilst never explicitly computing $\mathbf{S}$ or its eigenvectors.

2.3.1 Indirectly Computing the Eigenvectors

From the thin singular value decomposition (SVD) of $\mathbf{L}$,

$$\mathbf{L} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{2.11}$$

the right singular vectors $\mathbf{V} \in \mathcal{R}^{DF \times D^2}$ correspond to the eigenvectors of $\mathbf{L}^T\mathbf{L} = \mathbf{S} \in \mathcal{R}^{D^2 \times D^2}$, and the left singular vectors $\mathbf{U} \in \mathcal{R}^{DF \times DF}$ to the eigenvectors of $\mathbf{L}\mathbf{L}^T$ which we denote $\mathbf{S}^* \in \mathcal{R}^{DF \times DF}$. The eigenvectors $\mathbf{V}$ of $\mathbf{S}$ can be found efficiently by first computing the eigenvectors $\mathbf{U}$ of $\mathbf{S}^*$, then from Equation 2.11,

$$\begin{aligned} \mathbf{V}^T &= (\mathbf{\Sigma}^T \mathbf{U}^T \mathbf{U} \mathbf{\Sigma})^{-1} (\mathbf{U}\mathbf{\Sigma})^T \mathbf{L} & (2.12) \\ &= \mathbf{\Sigma}^{-1} \mathbf{U}^T \mathbf{L} \ . & (2.13) \end{aligned}$$

Letting $\hat{\mathbf{U}}, \hat{\mathbf{\Sigma}}, \hat{\mathbf{V}}$ be components of the SVD of $\mathbf{L}$ with the K largest magnitude singular values preserved, and $\hat{\Phi}(.)$ the corresponding low dimensional feature map, then

$$\mathbf{S} \approx \hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^2\hat{\mathbf{V}}^T \ . \tag{2.14}$$

The distribution of singular values in $\mathbf{S}^*$ suggests how well a rank reduction will preserve the information in $\mathbf{S}$. Figure 2.2 shows the eigenspectra of typical $\mathbf{S}$ matrices constructed from a number of filter representations. The spectra hint at the significant redundancies that can be exploited to reduce storage and computational costs associated with computing the low rank feature map. The $\sim \frac{1}{f}$ slope of the spectra correlates well with the statistical structure observed in natural images.

2.3.2 Applying the Eigenvectors

An explicit representation of $\mathbf{S}$ is unnecessary since the goal is to find an efficient closed-form expression for the feature maps. Thus,

$$\Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j) \approx (\mathbf{x}_i \otimes \mathbf{x}_i)^T\hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^2\hat{\mathbf{V}}^T(\mathbf{x}_j \otimes \mathbf{x}_j) \tag{2.15}$$

and since the kernel is imbued with an inner product, such that the computation is separable, a single feature map in isolation becomes,

$$\hat{\Phi}(\mathbf{x}_i) = \hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^T(\mathbf{x}_i \otimes \mathbf{x}_i) \ . \tag{2.16}$$

Substituting Equation 2.13 into Equation 2.16 gives,

$$\hat{\Phi}(\mathbf{x}_i) = \hat{\mathbf{U}}^T\mathbf{L}(\mathbf{x}_i \otimes \mathbf{x}_i) \ . \tag{2.17}$$

Whilst $\mathbf{L}$ is sparse for compact support filters, storage in memory quickly becomes prohibitive with increasing image size. For a 50×50 pixel input and 40 filters with 20×20 pixel support, storing the full $\mathbf{L}$ matrix will require on the order of 657 GB. We know however, that the joint portion $\mathbf{L}(\mathbf{x} \otimes \mathbf{x})$ can be efficiently computed using the original method of convolutions via,

$$\mathbf{L}(\mathbf{x} \otimes \mathbf{x}) = \begin{bmatrix} \mathbf{b} * [(\mathbf{g}_1 * \mathbf{x}) \odot (\mathbf{h}_1 * \mathbf{x})] \\ \vdots \\ \mathbf{b} * [(\mathbf{g}_F * \mathbf{x}) \odot (\mathbf{h}_F * \mathbf{x})] \end{bmatrix} \ . \tag{2.18}$$

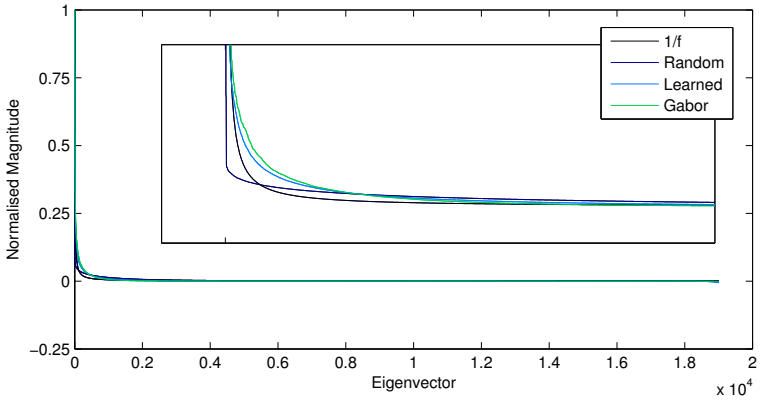


Figure 2.2: The eigenspectrum of $\mathbf{S}$ for a number of filter representations. The $\sim \frac{1}{f}$ distribution suggests a low rank approximation to $\mathbf{S}$ would preserve a significant portion of the variance. The magnified region shows in greater detail the energy distribution of the largest eigenvalues. The learned filters (using the method of [49]) have the most compact energy spectrum, followed by the Gabor filters, with the random filters having the broadest spectrum.

By taking this approach, only $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{U}}$ ever need be explicitly computed. For the example above, storing $\hat{\mathbf{U}}$ of rank $K = D$ will consume only 1.86 GB of memory. This is an amortized model setup cost, actual imagery is still transient.

Computing the feature map of Equation 2.1 incurs a cost of $O(DF \log D)$ operations and storage $O(DF)$. Computing the proposed feature map of Equation 2.17 incurs an added $O(KDF)$ operations but storage is only $O(K)$ where $K \ll DF$. Our feature map therefore realises a tradeoff between computational complexity and storage complexity, and results in a representation that is manageable for large amounts of high dimensional data, and as shown following, tractable in time when learning an SVM.

2.4 Support Vector Classification

Support vector machines have seen extensive use in visual classification tasks, and have proved particularly successful in tasks involving V1-inspired features [22]. Linear SVMs have a number of inherent advantages over kernel SVMs: faster learning times, the ability to learn from larger datasets, low computation cost during evaluation as the summation over support weights and vectors can be pre-computed, and most importantly, for some applications identical if not superior performance to nonlinear kernels (*e.g.*, RBF, polynomial, tanh) [25].

Given a set of training features and labels $\{\Phi(\mathbf{x}), y_i\}_{i=1}^{l}$, $\Phi(\mathbf{x}) \in \mathcal{R}^{DF}$, $y_i \in \{+1, -1\}$ a linear SVM attempts to find the solution to the following optimization problem,

$$\begin{aligned} \min_{\mathbf{w}, \xi_i \geq 0} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i\mathbf{w}^T\Phi(\mathbf{x}_i) \geqslant 1 - \xi_i, \quad i = 1 \ldots l \end{aligned} \tag{2.19}$$

where C is a penalty parameter and ξ_i are the slack variables introduced to offset the effects of outliers in the final solution.[1]

It is well understood in SVM literature that the $\mathbf{w}^T\mathbf{w}$ term in Equation 2.19 is inversely proportional to the margin of the solution. Maximizing this margin is central to the generalization properties of SVMs. The type of margin being maximized in this feature space is based on an unweighted (i.e. Euclidean) distance. Inspired by [2], however, we can demonstrate that an equivalent form of Equation 2.19 can be obtained by solving,

$$\begin{aligned} \min_{\mathbf{v}, \xi_i \geq 0} \quad & \frac{1}{2}\mathbf{v}^T\mathbf{S}^{-1}\mathbf{v} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i\mathbf{v}^T(\mathbf{x}_i \otimes \mathbf{x}_i) \geqslant 1 - \xi_i, \quad i = 1 \ldots l \end{aligned} \tag{2.20}$$

where the role of features has been completely subsumed into the weighted margin term $\mathbf{v}^T\mathbf{S}^{-1}\mathbf{v}$. The solutions to Equation 2.19 ($\mathbf{w} \in \mathcal{R}^{DF}$) and 2.20 ($\mathbf{v} \in \mathcal{R}^{D^2}$) are related by $\mathbf{w} = \mathbf{L}\mathbf{v}$ where $\mathbf{L} \in \mathcal{R}^{DF \times D^2}$ is previously defined in Equation 2.10. A key realisation here is that the role of the features is completely described as a margin manipulation – the weighting term is only applied to the margin term and not the data term.

2.4.1 Capacity of the Classifier

This result reflects previous work of Shivaswamy and Jebara [79] concerning what "type" of margin should be maximized during the estimation of a max margin classifier such as an SVM. In their work, Shivaswamy and Jebara discussed the importance of selecting the "correct" kind of margin when learning an SVM and how maximizing a margin based on Euclidean distance might not always be the best choice in terms of classifier generalization.

[1] The bias b is accounted for in $\mathbf{w} \leftarrow [\mathbf{w}^T, b]$ by $\Phi(\mathbf{x}) \leftarrow [\Phi(\mathbf{x})^T, 1]^T$ but is omitted here for brevity.

In fact, when one sets $\mathbf{S} = \mathbf{I}$ then the solution to the objective in Equation 2.20 reverts to a classical homogeneous second-order polynomial kernel SVM,

$$(\mathbf{x}_i \otimes \mathbf{x}_i)^T \mathbf{I} (\mathbf{x}_j \otimes \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j)^2 \tag{2.21}$$

A question that naturally arises is whether this induced kernel alone is sufficient for good performance, and we address this question later in the piece.

2.4.2 Complexity in SVM Training and Prediction

When training an SVM classifier, we modify Equation 2.19 to instead use our low dimensional feature map $\hat{\Phi}(\mathbf{x})$, which yields an optimisation over a lower (K) dimensional $\hat{\mathbf{w}}$,

$$\begin{aligned} \min_{\hat{\mathbf{w}}, \xi_i \geq 0} \quad & \frac{1}{2} \hat{\mathbf{w}}^T \hat{\mathbf{w}} + C \sum_{i=1}^{l} \xi_i \\ \text{subject to} \quad & y_i \hat{\mathbf{w}}^T \hat{\Phi}(\mathbf{x}_i) \geqslant 1 - \xi_i, \quad i = 1 \dots l \ . \end{aligned} \tag{2.22}$$

Training complexity remains $O(\max(N, D) \min(N, D)^2$, where N is the number of training examples, and D is the dimensionality of the data, except now $D = K$ [18].

Further to the space-time tradeoffs of §2.3.2, our method also realises a preprocessing learning tradeoff, which has benefits when training large datasets and enumerating over different training schemes. During prediction, however, we can take advantage of the form of $\hat{\Phi}(\mathbf{x})$ from Equation 2.13, to promote $\hat{\mathbf{w}}$ from a K dimensional space to a DF dimensional space through $\mathbf{w} = \mathbf{U}\hat{\mathbf{w}}$, such that for a vectorised test image $\mathbf{x}_i$,

$$\mathbf{w}^T \Phi(\mathbf{x}_i) \equiv \hat{\mathbf{w}}^T \hat{\Phi}(\mathbf{x}_i) \tag{2.23}$$

where $\Phi(\mathbf{x})$ is the original feature map of Equation 2.1.

2.4.3 Uniqueness of Filters

The structured form of the $\mathbf{S}$ matrix gives us an insight into the role of filters in the margin manipulation, specifically the uniqueness of the filter responses and their joint contribution to the invariant representation. The matrix $\mathbf{S} = \mathbf{L}\mathbf{L}^T$

can be represented as a concatenation of $F \times F$ sub-matrices,

$$\begin{aligned} \mathbf{L}_i\mathbf{L}_j^T &= \mathbf{B}\mathbf{M}(\mathbf{G}_i \otimes \mathbf{H}_i)(\mathbf{G}_j \otimes \mathbf{H}_j)^T\mathbf{M}^T\mathbf{B}^T \\ &= \mathbf{B}\mathbf{M}(\mathbf{G}_i\mathbf{G}_j^T) \otimes (\mathbf{H}_i\mathbf{H}_j^T)\mathbf{M}^T\mathbf{B}^T \ . \end{aligned} \tag{2.24}$$

From this form one can see that the role of the individual filters in this form is not unique since $\mathbf{G}_i\mathbf{A}\mathbf{A}^{-1}\mathbf{G}_j^T = \mathbf{G}_i\mathbf{G}_j^T$ where $\mathbf{A}$ is any arbitrary full rank transform matrix. Further, it is possible to show that the interaction of these filters $\mathbf{G}_i\mathbf{G}_j^T$ is unique up to a sign ambiguity.[2] Finally, it is possible to see where spatial invariance stems from in the weighting matrix $\mathbf{S}$ since for $i = j$ local phase is lost, and when $i \neq j$ only relative phase is preserved.

2.5 Second-Order Interactions

The term $(\mathbf{x} \otimes \mathbf{x})$ introduced in Equation 2.6 can alternatively be written as,

$$(\mathbf{x} \otimes \mathbf{x}) = \text{vec}(\mathbf{x}\mathbf{x}^T) \ , \tag{2.25}$$

which is the vectorized covariance matrix of all pixel interactions. Tuzel *et al.* [88] showed that the covariance of a local image distribution is often enough to discriminate it from other distributions. However, when dealing with high-dimensional distributions, computing a full-rank covariance matrix is often difficult. Hariharan *et al.* [34] circumvent this problem by assuming stationarity of background image statistics (a translated image is still an image), as well as limiting the bandwidth of interactions between pixels. Simoncelli [81] showed that these assumptions are reasonable, since correlations between pixels fall quickly with distance (see Figure 2.3).

To improve conditioning and prevent overfitting the classifier to hallucinated interactions, we consider the most general set of *local* second-order features: the set of all local unary second-order interactions in an image,

$$\Psi(\mathbf{x}) = [\text{vec}\{\Psi_1(\mathbf{x})\}^T, \ldots, \text{vec}\{\Psi_D(\mathbf{x})\}^T]^T \tag{2.26}$$

[2] Since $\mathbf{x} \otimes \mathbf{x} = \text{vec}(\mathbf{x}\mathbf{x}^T)$ where we know through the SVD that one can recover $\mathbf{x}$ up to a sign. Here we assume $\mathbf{x} = \text{vec}(\mathbf{G}_i\mathbf{G}_j^T)$ from Equation 2.24.

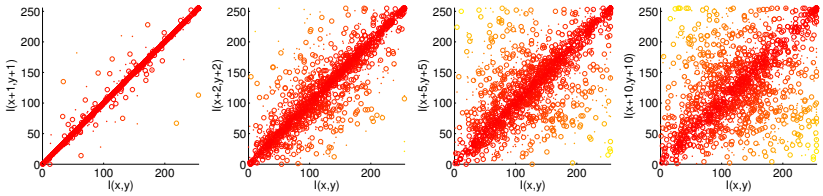


Figure 2.3: An illustration of the locality of pixel correlations in natural images. Whilst a single pixel displacment exhibits strong correlations in intensity, there are few discernible correlations beyond a 5 pixel displacement. Locality is also observed in the human visual system, where cortical cells have finite spatial receptive fields.

where,

$$\Psi_i(\mathbf{x}) = \mathbf{P}_i \mathbf{x}\mathbf{x}^T \mathbf{P}_i^T \ , \tag{2.27}$$

$\mathbf{P}_i$ is simply an $M \times D$ matrix that extracts an M pixel local region centred around the ith pixel of the image $\mathbf{x}$. By retaining local second-order interactions, the feature length grows from D for raw pixels to M^2D.

Fortunately, inspection of Equation 2.26 reveals a large amount of redundant information. This redundancy stems from the re-use of pixel interactions in surrounding local pixel locations. Taking this into account, and without loss of information, one can compact the local second-order feature to MD elements, so that Equation 2.26 becomes,

$$\Psi^*(\mathbf{x}) = \begin{bmatrix} (\mathrm{e}_1 * \mathbf{x})^T \circ \mathbf{x}^T \\ \vdots \\ (\mathrm{e}_M * \mathbf{x})^T \circ \mathbf{x}^T \end{bmatrix} \ . \tag{2.28}$$

where $\{\mathrm{e}_m\}_{m=1}^M$ is the set of M impulse filters that encode the local interactions in the signal.

2.5.1 Local Second-Order Interactions

To illustrate the importance of local second-order interactions, consider a simple thought experiment involving two classes, A and B. Class A represents the distribution of all natural images. Class B represents a noise distribution which has the same frequency spectrum as natural images, namely $\frac{1}{f}$ [81]. Both distributions are power normalized. We sample 25000 training and testing examples

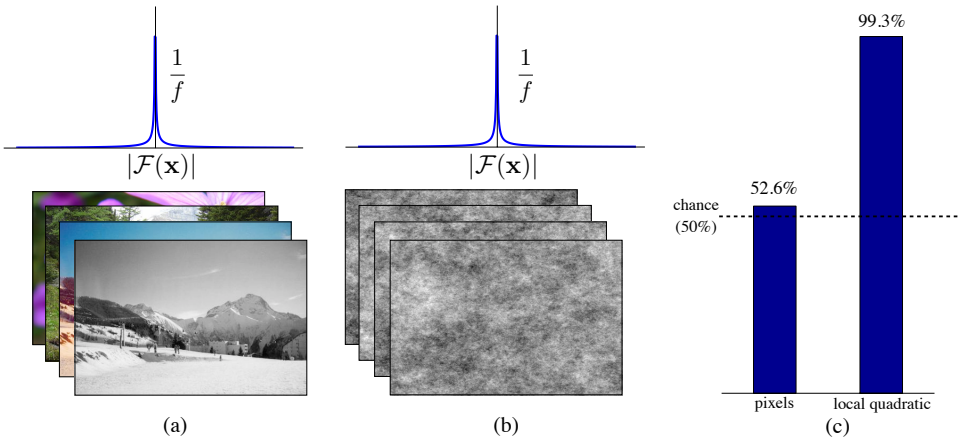


Figure 2.4: Thought-experiment setup. (a) contains an ensemble of samples drawn from the space of natural images with a $\frac{1}{f}$ frequency spectrum, (b) contains an ensemble of samples drawn from a random noise distribution *with the same* $\frac{1}{f}$ *frequency spectrum*. (c) We train two linear classifiers to distinguish between "natural" or "noise." The pixel-based classifier does not have the capacity to discriminate between the distributions. The classifier which preserves local quadratic pixel interactions almost perfectly separates the two distributions.

from each class, and train two classifiers: one preserving the raw pixel information and one preserving *local second-order interactions* of the pixels. The goal of the classifiers is to predict "natural" or "noise." An illustration of the experimental setup and the results are presented in Figure 2.4. The pixel classifier fails to discriminate between the two distributions. There is no information in either the spatial or Fourier domain to linearly separate the classes (*i.e.* the distributions overlap). By preserving local quadratic interactions of the pixels, however, the classifier can discriminate natural from synthetic almost perfectly.

Whilst the natural image and noise distributions have the same frequency spectra, natural images are not random: they contain structure such as lines, edges and contours. This experiment suggests that image structure is inherently local, and more importantly, that local second-order interactions of pixels can exploit this structure. Without encoding an explicit prior on edges, pooling, histogramming or blurring, local quadratic interactions have sufficient capacity to exploit the statistics of natural images, and separate them from noise.

2.5.2 Replacing prior with posterior: learning over pixels

Quadratic kernel SVMs trained on pixels have not historically performed well on recognition tasks when learned using pixel information. The image prior that

HOG encodes, and the affine weighting that it can be distilled into, is integral to obtaining good generalization performance. We know, however, that a prior is simply used to reflect a belief in the posterior distribution in the absence of actual data. In the case of HOG, the prior encodes insensitivity to local non-rigid deformations so that the entire space of deformation does not need to be sampled to make informed decisions.

This is usually a reasonable assumption to make, since sampling the posterior sufficiently may be infeasible. Take, for example, the task of pedestrian detection. The full posterior comprises all possible combinations of pose, clothing, lighting, race, gender, identity, background and any other attribute that manifests in a change to the visual appearance of a person. Multi-scale sliding window HOG detectors work to project out as much of this intra-class variation as possible.

Is it possible to learn a performant detector using only the assumptions that underlie HOG features: the preservation of local second-order interactions? How much data is required to render the HOG prior unnecessary, and what sort of data is required? Can the data just be perturbations of the training data? Does the resulting classifier learn anything more specialized than one learned on HOG features?

Learned representations such as convolutional networks are quickly surpassing the performance of hand-crafted features in many large scale visual object recognition tasks, and feature learning has a strong backing [36]. Convolutional networks model the stationarity and locality of image interactions efficiently using the convolution operator. In our experimental section, we remove the stationarity assumption and preserve only the locality of interactions in a single layer representation, and transfer the burden on learning the distribution to the classifier.

2.6 Experiments

We evaluate several aspects of our reformulation on the MNIST, Caltech 101, Cohn Kanade+ and INRIA Person datasets. Through our thought experiments we illuminated some surprising properties of our reformulation. Here, we illustrate how it remains competitive on established benchmarks. We mimic the experimental setup of other authors who have used similar V1-inspired features.

Where we say $\text{rank}(\mathbf{S}) = D$, we take D to be the dimensionality of the

vectorised input image. In the case of frequency and orientation selective filters, we use a bank of log Gabor filters. In the case of random filters, we use the same number of filters as the Gabor case, and ensure that each filter has zero mean and unit norm. For each convolution, we only keep the central area that is the same size as the input image.

2.6.1 Reintroducing Photometric Normalisation

Jarrett *et al.* [42] show that rectification and photometric normalisation are the single most important factors in improving the performance of a recognition system, especially in images exhibiting large photometric variation, as observed in natural images (*e.g.* Caltech 101).

Given a pointwise processing stage $\Psi(.)$ that maps $\mathcal{R}^{DF} \rightarrow \mathcal{R}^{DF}$ Equation 2.17 can be extended to

$$\Phi(\mathbf{x}) = \mathbf{U}^T \Psi\big(\mathbf{L}(\mathbf{x} \otimes \mathbf{x})\big) \; . \tag{2.29}$$

This allows us to include mid-processing such as photometric normalisation without loss of generality.

2.6.2 MNIST

MNIST is a handwritten character recognition dataset containing 60000 training examples and 10000 test examples of the characters $0-9$. Each character is roughly centred in a 28×28 window and quantised to 8-bit grayscale. Although MNIST is an ageing dataset, LeCun's convolutional network architecture – which for a single layer closely follows our parametric form – has shown particularly impressive performance at the task [42].

We use 48 Gabor filters at 12 orientations and 4 frequencies each of size 28×28, and a boxcar filter of size 3×3. We remove the photometric normalisation step from our model, but preprocess each image by power normalisation. Due to the large number of training data and the resulting descriptor dimensionality of 37632, we opt to train the resulting linear SVM in the prime. Average classification performance is shown in Figure 2.5(a).

2.6.3 Caltech101

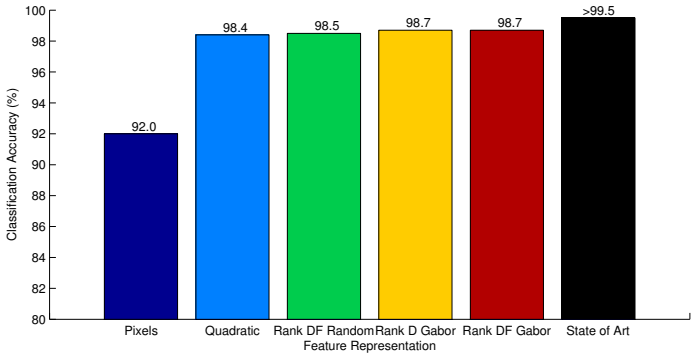
Caltech 101 is a "natural" object recognition dataset containing 101 object classes, each with $40 - 800$ instances. The objects are roughly centred and in similar poses, though vary in appearance. Pinto has pointed to a number of flaws in the dataset and argues that it lacks true real-world variability, and supports his claims by achieving good performance with a simple biologically motivated feature representation [71]. We mimic his setup and achieve similar performance whilst illustrating some advantages of our method.

We use 92 Gabor filters at 16 orientations and 6 scales each of size 43×43, and a boxcar filter of size 17×17. We preprocess the images by resizing and cropping each to fit a 150×150 pixel box. We modify our model to include a downsampling matrix which subsamples each filter response by a factor of 5 (to a 30×30 image). Average classification performance is shown in Figure 2.5(b).

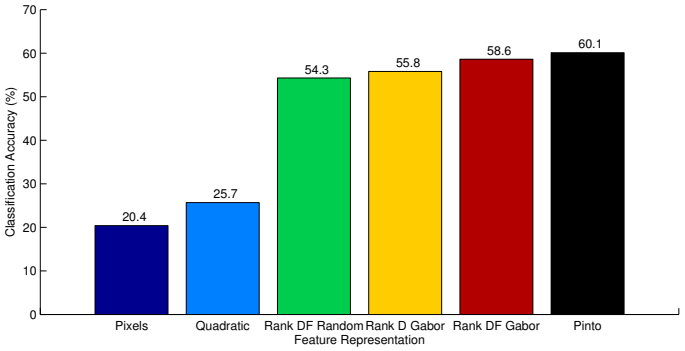
2.6.4 PCA on Responses

To deal with the "curse of dimensionality", many papers have been devoted to finding low dimensional approximations to descriptors using PCA, LDA or nonlinear dimensionality reduction methods [54, 99, 30]. These methods have two inherent problems: the reduction is data dependent and needs to be recomputed for each new set of data, and the reduction must occur in the original dimensionality and may not be feasible in time or space.

Equation 2.17 suggests that the matrix $\mathbf{U}$ acts to transform the feature onto a low rank orthonormal basis which preserves the highest modes of variance. The advantages of this approach are twofold: the reduction can be precomputed in the absence of data and the reduction is based on the filter components that are likely to be discriminative rather than the observed modes of deformation specific to each training set. Figure 2.6 shows the classification performance of our method as a function of feature length, using the Caltech 101 setup with Gabor filters. A number of PCA schemas are shown for comparison. PCA Matched (10%) and PCA Mismatched show how PCA fails to generalise when the data used to calculate the loadings either does not span the full extent of geometric variability in the training and testing sets, or is from a different domain entirely. Our method suffers neither of these drawbacks, yet approaches the performance of PCA with loadings calculated from the full training set (PCA Matched (100%)).



(a) MNIST



(b) Caltech 101

Figure 2.5: Average classification performance across all classes of the (a) MNIST, and (b) Caltech 101 dataset for different feature descriptor representations. (Pixels) raw pixels, (Quadratic) quadratic kernel on raw pixels, (Rank DF Random) the full $\mathbf{S}$ matrix constructed from random filters, (Rank D Gabor) a low rank approximation to the full $\mathbf{S}$ matrix constructed from Gabor filters, (Rank DF Gabor) the full $\mathbf{S}$ matrix constructed from Gabor filters, (State of Art) State of the Art benchmark for MNIST taken from a survey of 60 algorithms, (Pinto) the reference method of Pinto [71].

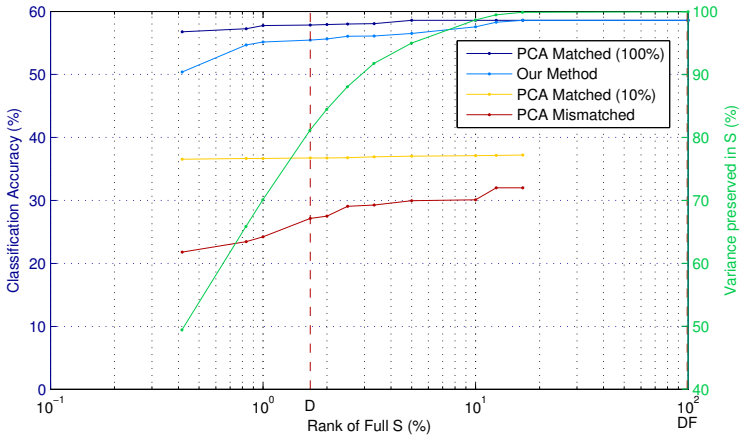


Figure 2.6: A comparison of dimensionality reduction techniques on the Caltech 101 dataset. Performance is measured as average classification accuracy across classes as a function of descriptor dimensionality. (PCA Matched 100%) PCA loadings calculated from the entire training set. (Our Method) Dimensionality reduction using a low rank approximation to $\mathbf{S}$. (PCA Matched 10%) PCA loadings calculated from 10% of the training set, with equal class representation. (PCA Mismatched) PCA loadings calculated from Cohn Kanade+ dataset. The green curve shows the variance of $\mathbf{S}$ preserved as a function of the rank. A descriptor of rank D not only models 80% of the variance in the original DF representation, but achieves similar classification performance. PCA consistently performs $\sim 4\%$ better, but only in well-matched conditions.

2.6.5 Cohn Kanade+

Cohn Kanade+ is an expression recognition dataset consisting of 68-point landmark, broad expression and FACS labels across 123 subjects and 593 sequences. Each sequence varies in length and captures the neutral expression in the first frame and the peak formation of facial expression in the last. We follow the experimental setup of Lucey *et al.* [58], however we consider only the broad expressions and discard the AU labels.

We register each face to a canonical geometric template then measure classification accuracy across all expressions with increasing registration error. Results are shown in Figure 2.7.

We designed an experiment where we could control the amount of geometric misalignment observed between the training and testing examples. We used the Cohn Kanade+ expression recognition dataset, consisting of 68-point landmark, broad expression and FACS labels across 123 subjects and 593 sequences. Each sequence varies in length and captures the neutral and peak formation of facial expression. In this chapter we consider only the task of broad expression

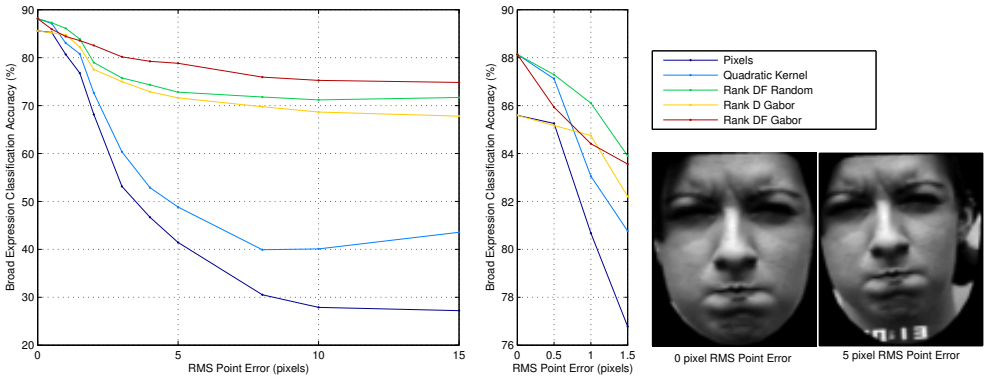


Figure 2.7: Classification performance on Cohn Kanade+ broad expressions as a function of increasing registration error. Feature representations have better robustness to registration error. The central magnified panel shows that with perfect registration, the rank DF representations converge to a quadratic kernel and the rank D representations converge to (a linear kernel on) raw pixels. A quadratic kernel represents the inherent capacity of our V1-like feature parameterisation in a linear SVM learning scheme.

classification (*i.e.* we discard FACS encodings). To test the invariance of different types of features to geometric misalignment, we first register each training example to a canonical pose, then synthesize similarity warps of the examples with increasing RMS point error.

2.6.6 Why Faces?

HOG features have been used across a broad range of visual recognition tasks, including object recognition, scene recognition, pose estimation, *etc.* Faces are unique, however, since they are a heavily studied domain with many datasets containing subjects photographed under controlled lighting and pose conditions, and labelled with ground-truth facial landmarks. This enables a great degree of flexibility in experimental design, since we can programatically set the amount of geometric misalignment observed while controlling for pose, lighting, expression and identity.

We synthesize sets with 300, 1500, 15000 and 150000 training examples. The larger the synthesized set, the greater the coverage of geometric variation. We use HOG features according to Felzenszwalb *et al.* [27] with 18 orientations and a spatial aggregation size of 4. For the reformulation of Equation 2.1, we use Gabor filters with 18 orientations at 4 scales, and a 4×4 blur kernel. The local quadratic features have a spatial support equal to the amount of RMS point

(a) 0-pixel error

(b) 10-pixel error

Figure 2.8: Illustrative examples of subjects from the Cohn Kanade+ dataset with (a) zero registration error, and (b) 10 pixels of registration error.

error (*i.e.* at 10 pixels error, correlations are collected over 10×10 regions). All training images are 80×80 pixels and cropped around only the faces. Figure 2.8 illustrates the degree of geometric misalignment introduced.

2.6.7 Learning

The storage requirements of local quadratic features quickly explode with increasing geometric error and synthesized examples. At 10 pixels RMS error, 150000 training examples using local quadratic features takes 715 GB of storage. To train on such a large amount of data, we implemented a parallel support vector machine [12] with a dual coordinate descent method as the main solver [38]. Training on a Xeon server using 4 cores and 24 GB of RAM took between 1 – 5 days, depending on problem size. We used multiple machines to grid search parameters and run different problem sizes.

Figure 2.9 shows a breakdown of the results for synthesized sets of geometric variation. Pixels (shown in shades of green) perform consistently poorly, even with large amounts of data. HOG features (in blue, and reformulation in aqua) consistently perform well. The performance of HOG saturates after just 1500 training examples. Zhu *et al.* talk about the saturation of HOG at length, noting that more data sometimes *decreases* its performance [105].

Local quadratic features (shown in red) have a marked improvement in performance with increasing amounts of data (roughly 10% per order of magnitude of training data). Synthesizing variation can be an efficient means of augmenting the amount of labelled training data available, and this result illustrates how geometric perturbations of the training data can be an effective replacement for encoding explicit prior in the image representation.

Only when the dataset contains $\geq$ 100000 examples do the local quadratic

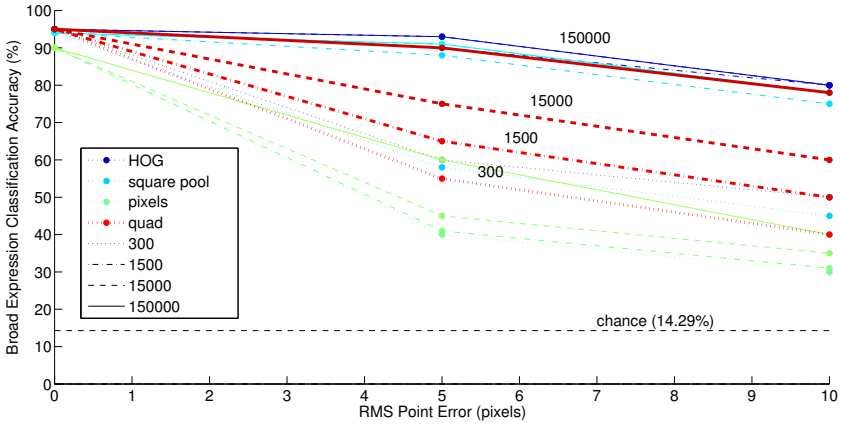


Figure 2.9: Broad expression classification accuracy for different feature representations as a function of alignment error and amount of training data. For each feature representation we synthesized 300, 1500, 15000 and 150000 training examples. The held out examples used for testing *always* come from an unseen identity. HOG features quickly saturate as the amount of training data increases. Quadratic features, shown in red, have poor performance with only a small number of synthesized examples, but converge towards the performance of HOG as the space of geometric variation is better spanned. Quadratic features appear to improve by roughly 10% per order of magnitude of training data, until saturation.

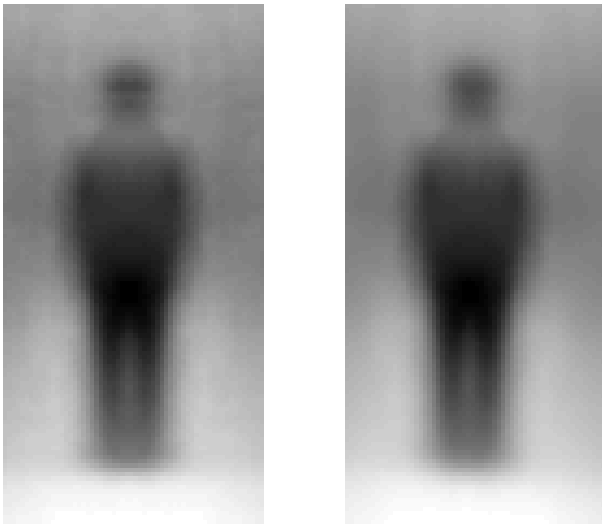


Figure 2.10: The pixel mean of positive examples from the INRIA training set, (a) only, and (b) with 20 synthesized warps per example. The mean is virtually the same, suggesting that the synthesized examples are not adding ridig transforms that could be accounted for by a multi-scale sliding-window classifier.

features begin to model non-trivial correlations correctly. With 150000 training samples, local quadratic features perform within 3% of HOG features.

2.6.8 Pedestrian Detection

We close with an example showing how the ideas of *locality* and *second-order* interactions can be used to learn a pedestrian detector. We don't intend to outperform HOG features. Instead we show that preserving higher-order interactions between pixels are critical for good performance on geometrically diverse object categories.

We follow a similar setup to our earlier expression recognition experiment on INRIA person. We generate synthetic similarity warps of each image, making sure they remain aligned with respect to translation. Figure 2.10 illustrates how the addition of synthesized examples does not change the dataset mean appreciably (misalignment would manifest as blur). We train the SVM on 40,000 positive examples and 80,000 negative examples, without hard negative mining.

The results are striking. Unsurprisingly, the pixel-based classifier has high

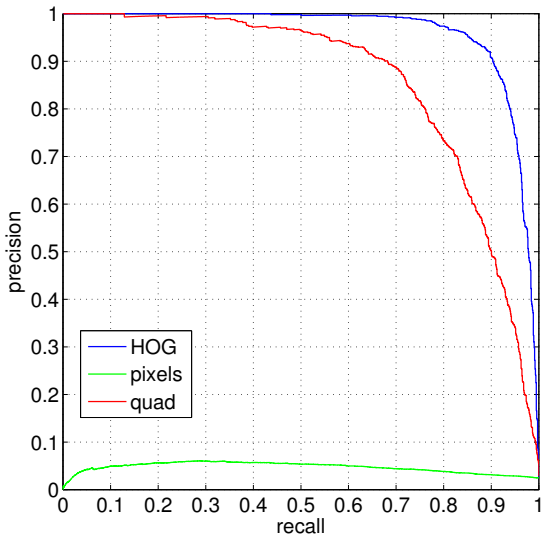


Figure 2.11: Precision recall for different detectors on the INRIA person dataset. HOG performs well and pixels perform poorly, as expected. Local second-order interactions of the pixels (quad) perform surprisingly well, considering the lack of image prior and contrast normalization. The added capacity and locality constraints go a long way to achieving HOG-like performance.

detection error, whilst the HOG classifier performs well. The local-quadratic classifier falls between the two, with an equal error rate of 22%. The improved performance can be attributed solely to the added classifier capacity and its ability to learn from geometric pertubations of the training set.

2.7 Discussion

The application of V1-inspired features can be reinterpreted as a weighted margin on the Kronecker basis expansion of an image. This insight becomes clearer in Equation Equation 2.20 when viewed in the context of training a linear SVM. The prior on the margin is a global spatial weighting on the responses to oriented edge filters, which appear to encode some phase invariance along with relationships between frequency and orientation bands. The Cohn Kanade+ dataset was used to explore the weighted margin insight under known ground-truth geometric distortions. The results of Figure 2.7 reveal a pervasive insight. Image features give better robustness to registration error than raw pixel representations. With perfect registration however, the performance of rank DF

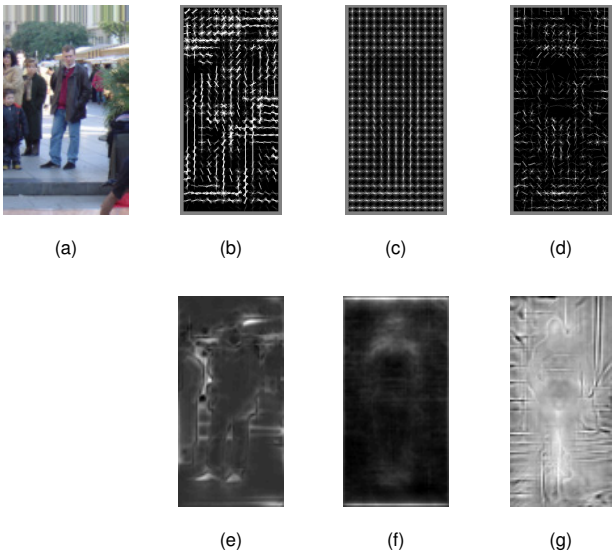


Figure 2.12: Visualisations of HOG and local quadratic classifiers on the INRIA person dataset. (a) A sample image from the set and its equivalent representation in (b) HOG space, and (e) local quadratic space. Lighter shades represent strong edges and correlations respectively. The positive training set mean in (c) HOG space and (f) local quadratic space. Positive weights of the final classifiers in (d) HOG space and (g) local quadratic space. In both HOG and local quadratic space, the visualization of individual images shows significant extraneous informative, however the final classifiers are clearer. Interpreting the local quadratic classifier in (g) is difficult since the correlations cannot be interpreted as edges, however the distribution of positive weights is similar to HOG, especially around the head and shoulders and between the legs.

representations converge to a quadratic kernel on the raw pixels, whilst the performance of the rank D representations converge to (a linear kernel on) raw pixels.

This suggests that in the absence of geometric noise, the filter prior over the data has no influence. The process of *gaining invariance* importantly does not improve performance outright; but rather only in the face of geometric mismatch. With perfect registration the class separation is sufficiently large that a prior on the margin has no effect on the discriminability of the decision hyperplane. It is only with increasing registration error and increasing nonlinearity of the true decision boundary that the prior helps guide the separating hyperplane to a good solution.

The crux of the rank reduction lies in its relationship to and advantage over regular PCA. Because PCA is data dependent, it relies on an explicit representation of the entire training set, and a strong affinity between the observed geometric variability in the training and testing sets. The $\mathbf{S}$ matrix is data agnostic and a rank reduction on this matrix is equivalent to an optimisation over the most important frequency components and their spatial support. Further, we show in Figure 2.6 that this approach is comparable with PCA. In essence, our choice of filters conveys our intuition about what spatial and frequency content is semantically important in images. A rank reduction on $\mathbf{S}$ acts to preserve the most important parts of this prior.

The choice of filters is an important consideration in the design of image representations as they constitute an assumed prior over the image statistics. However, rather than expressing the prior explicitly through filters, we should consider instead a regularization on the optimization procedure, such as a weighting on the margin on a support vector machine which encodes the same prior, *is* unique, and moves the computational burden from feature construction to training.

2.8 Conclusion

This chapter has focused on exploring how the choice of representation and classification are intrinsically linked in visual perception tasks. In particular, we showed that a recognition system based on a V1-inspired feature extraction process and a linear SVM hallucinates a classifier with quadratic capacity. We further showed that by removing the V1 prior and replacing it with a parametric model for generating geometrically perturbed versions of the training data,

we could achieve equivalent classification performance. This is in line with the hypothesis that a principal role of V1-inspired features is to encode some invariance to geometry.

As a supporting result, we showed that as geometric alignment of the dataset improves, the effect of the V1 prior becomes much less pronounced. With near-perfect alignment, a quadratic SVM learned on pixels performs as well as a linear SVM learned on a V1-inspired representation, illustrating that capacity of the classifier is important, even when matching well-aligned signals.

Since this work was originally performed, convolutional networks have made significant strides into improving performance on detection and classification tasks. One of the main contributors to their success has been to learn layers of task-specific filters in a fully supervised manner that provide better invariance higher-level geometrical structures and deformations. Their basic motivation is to compose multiple layers of V1-like representations, with each layer introducing more capacity and geometric tolerance whilst controlling the total number of parameters to learn.

In the following chapter, we explore how V1-inspired representations can be effectively used for the alignment and correspondence of general object classes. Since this is a primarily unsupervised task, we demonstrate a number of ways in which making better use of prior information can improve upon existing approaches to these problems.

Chapter 3

Stationarity and Correspondence

In this chapter, we consider the problem of performing unsupervised image registration on general object classes. This is a particularly compelling problem for dense labeling objectives, or as a preprocessing step for higher-level problems such as $3D$ model building and object recognition.

Unsupervised image registration problems usually consider images that are spatially adjacent (stereo), temporally adjacent (optical flow), or linearly correlated (congealing, RASL). We instead consider the problem of aligning images which are related only by their visual class, which we term *semantic* correspondence. This is an under-constrained problem, since the objects being aligned can vary significantly in appearance and geometry. At the same time, the absence of any training data makes unsupervised alignment of general object classes highly challenging.

As a result, this raises a number of interesting questions about how to best utilize the information available - in this case, the space of all natural images. For example, what representation of images is tolerant to geometric and photometric variation, whilst remaining selective to the semantic content? What descent method can reliably form a basin of attraction between misaligned images from the same object class, without prior knowledge of that class? What class-agnostic statistics of images can be used to improve class-specific matching?

In the first and second chapter, we concentrated on the first question: what interactions should be preserved when representing images, especially in the face of geometric uncertainty? In this chapter, we focus on *recovering* geometry via alignment, and the priors that help make this feasible.

For example, given two images of elephants, we would like to bring them into "alignment", so that semantically related features of the elephants correspond.

More formally, given two images and a discrete set of points $\mathbf{x}$, we pose the semantic correspondence problem as the inverse fitting optimization,

$$\mathbf{x}^* = \arg\min \sum_{i=1}^{MN} f_i(\mathbf{x}_i) + \lambda g(\mathbf{x}) \tag{3.1}$$

where f is the matching function that evaluates the likelihood of a particular assignment for each $\mathbf{x}_i$ based on the image content, and g is a regularizer which enforces constraints on the joint configuration of the points.

Under this umbrella definition of alignment, we can instantiate particular models of optical flow, pose estimation, facial landmark fitting, deformable parts modeling and unsupervised alignment. In semantic alignment, the matching function must be robust to significant intra-class variation in appearance and geometry.

The regularizer typically forms either an explicit parametrization of the modes of deformation, or pixel-wise motion constraints. These approaches manifest in two different optimization procedures: continuous gradient based methods and discrete graphical model based methods. In this chapter, we consider gradient and discrete methods and show how representational prior can be effectively leveraged in both.

3.1 Related Work

Gradient-based search strategies are attractive due to their ability to handle parametric warps and subspace constraints. Gradient-based strategies have a long history in alignment literature [20, 26, 37, 100]. The most notable application of this concept is the classic Lucas & Kanade (LK) algorithm [57], which has been used primarily for image registration. Many variations upon this idea now exist in computer vision literature [10, 20, 3] for applying gradient search to object registration. A traditional drawback of gradient-based methods is that the gradients must be computed on the image function directly, which places some restrictions on the form of the image function. In particular, the image function is assumed to be smooth and differentiable, which precludes the use of many feature representations or classifier responses.

Graphical models, on the other hand, can entertain arbitrarily complex matching functions. They need be smooth or differentiable. However, inference in graphical models is difficult and inexact in all but the simplest models

such as tree- or star-structured graphs. For example, in the application of graph-based search to optical flow – termed SIFT Flow [55] – the regularization on the $2D$ smoothness of the flow prevents the allowable warps from being factored into a tree structure. Instead, the authors employ an alternation strategy of enforcing the smoothness constraint in the $x-$ and then $y-$ directions (each of which independently can be represented as a tree structure) using dual-layer belief-propagation to find an approximate global solution. In many other cases, simplified tree- or star-structured models are unable to capture important dependencies between parts, so are not representative of the underlying structure or modes of deformation of the object being modelled [104]. The limited expressiveness of these simple models prevents many interesting constraints from being explored, which has led to the study of discrete but non-graphical models [74].

Canonical correspondence problems such as stereo and optical flow typically rely on simple (dis-)similarity metrics to describe the likelihood of two pixels matching. In the original work of Horn and Schunck [37] and Lucas and Kanade [57] this was Euclidean distance on raw pixel intensities, which manifested a brightness constancy assumption.

Since then, significant literature has focused on determining robust metrics under increasingly adverse conditions - from non-rigid deformations and occlusions, to non-global intensity, constrast and colorimetric changes [13, 63, 78, 84]. Importantly, however, all of these works assume the images being observed stem from the same underlying scene.

SIFT Flow first introduced the idea of semantic correspondence *across* scenes [55]. While the method uses a simple L_1 metric, the images are represented in dense SIFT space typically associated with sparse keypoint matching.

A number of dense correspondence methods have made use of discriminative pre-training [53, 75, 84], with the recent work of Ladicky *et al.* [48] being particularly relevant to our discussion. In this work, a classifier of the form $f(\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2))$ is trained to predict a (binary) likelihood of two pixels matching. Intuitively, the classifier learns the modes and scale of variation in the underlying feature space Φ that are important and those that are distractors. Training is fully supervised from groundtruth optical flow data.

Like SIFT Flow,Ladicky *et al.* [48] formulate the correspondence objective as a graphical model ([45, 46] respectively). This has the distinct advantage over variational methods of permitting very large displacements and arbitrarily complex data terms, at the expense of requiring simple regularizers to keep inference tractable. More recently, a number of variational optical flow methods

have used sparse descriptor matching to anchor larger displacements [14, 95]. While both methods use robust SIFT descriptors for keypoint matching, in a semantic correspondence setting the best match is infrequently the true correspondence, leading to poor initialization of the densification stage.

3.2 Gradient Based Alignment

For gradient-based alignment, we adopt a squared loss similarity metric for the matching function,

$$f(\mathbf{x}) = ||\mathcal{I}_A - \mathcal{I}_B(\mathbf{x})||_2^2 \tag{3.2}$$

with a gradient update defined as the Lucas and Kanade optimization procedure,

$$\Delta\mathbf{x}^* = \arg\min_{\Delta\mathbf{x}} ||\mathcal{I}_A - \mathcal{I}_B(\mathbf{x} + \Delta\mathbf{x})||_2^2 + g(\Delta\mathbf{x}) \tag{3.3}$$

where $\mathcal{I} : \mathbb{R}^{D\times 2} \rightarrow \mathbb{R}^D$ is the image function which samples the (sub-)pixel values at the given locations, where $\mathbf{x}_i = [x_i, y_i]^T$ is the ith $x-$ and $y-$ discrete coordinates sampled on a regular grid at integer pixel locations within the continuous image function. $g : \mathbb{R}^{2D} \rightarrow \mathbb{R}$ is the regularization function that penalizes the likelihoods of each possible deformation vector $\Delta\mathbf{x}$. In the case of parametric warps (affine, similarity, homography, *etc.*) the regularization function acts as an indicator function which has zero cost if the deformation vector adheres to the desired parametric warp or infinite cost if it does not.

Since pixel intensities are known to be poor estimators of semantic object or part similarity, one can instead consider a feature mapping function,

$$f(\mathbf{x}) = ||\Phi_A - \Phi_B(\mathbf{x})||_2^2 \tag{3.4}$$

where $\Phi_A(\mathbf{x}) = \Phi(\mathbf{x}; \mathcal{I}_A)$ is a nonlinear feature representation of the image $\mathcal{I}_A$ evaluated at $\mathbf{x}$. As per the previous chapters, we consider representations derived from densely sampled sparse V1-inspired features, such as HOG or SIFT.

An important factor for gradient-based search strategies is the accuracy of the linearization matrix function of the representation (whether raw pixel intensities or densely sampled sparse features) with respect to the deformation vector. The linearization matrix function, or gradient as it is often referred to

in computer vision, estimates an approximate linear relationship between the representation function and the deformation vector $\Delta\mathbf{x}$ over a restricted set of deformations. However, does the accuracy of this linearization reflect its utility in gradient search alignment?

We argue that gradient search alignment strategies are often needlessly dismissed, as the linearization of $\Phi(\mathbf{x})$ of most natural images is poor in comparison to that obtained from $\mathcal{I}(\mathbf{x})$. We demonstrate empirically that in spite of the poor linearization approximations of sparse features like SIFT and HOG, they actually enjoy superior gradient search alignment performance in comparison to raw pixel representations.

3.2.1 The Lucas & Kanade Algorithm

Recollect our formulation of the alignment problem in Equation 3.3. A common substitution within the LK algorithm is,

$$\mathbf{p}^* = \arg\min_{\mathbf{p}} ||\Phi_A - \Phi_B(\mathbf{p})||_2^2 \tag{3.5}$$

where $\mathbf{p}$ is a set of warp parameters that model the deformation vector $\Delta\mathbf{x}$ by proxy of a warp function,

$$\Phi(\mathbf{p}) = \begin{bmatrix} \Phi\{\mathcal{W}(\mathbf{x}_1;\mathbf{p})\} \\ \vdots \\ \Phi\{\mathcal{W}(\mathbf{x}_D;\mathbf{p})\} \end{bmatrix} \tag{3.6}$$

and $\mathcal{W}(\mathbf{x};\mathbf{p}) : \mathbb{R}^{2D} \rightarrow \mathbb{R}^P$. The warp function conditions the deformation vector on the warp parameters such that $\mathbf{x} + \Delta\mathbf{x} = \mathcal{W}(\mathbf{x};\mathbf{p})$. In most instances the dimensionality of $\mathbf{p} \in \mathbb{R}^P$ is substantially less than the canonical deformation vector $\Delta\mathbf{x} \in \mathbb{R}^{2D}$ (e.g. for a 2D affine warp $P = 6$). This is equivalent to setting g to be an indicator function, which has zero cost when the parameters fall within the feasible set of warps, and infinity otherwise. The LK algorithm takes successive first-order Taylor expansions about the current estimate of the warp parameters, and solves for the local update,

$$\Delta\mathbf{p}^* = \arg\min_{\Delta\mathbf{p}} ||\Phi_B(\mathbf{p}) + \nabla\Phi_B(\mathbf{p})\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\Delta\mathbf{p} - \Phi_A||_2^2 \tag{3.7}$$

where $\nabla\Phi_B(\mathbf{p})$ is the gradient estimator, and $\frac{\partial\mathcal{W}}{\partial\mathbf{p}}$ is the Jacobian of the warp function which can be found deterministically or learned offline. Here we have

presented the LK algorithm using the canonical L_2 loss function. In reality there are a number of possible variations on this classical LK form. Baker *et al.* [4, 5] provide a thorough reference for choosing an appropriate update strategy and loss function. We present LK in this manner to avoid introducing unnecessary and distracting detail for the unfamiliar reader. Regardless of the matching function, the choice of image representation and method of gradient calculation greatly affect the alignment performance observed.

3.2.2 Linearizing Non-Differentiable Features

As stated earlier, our central focus in this chapter is to first investigate how well sparse features like HOG and SIFT linearize compared to pixel intensities. To do this we first need to review how one estimates the representation's gradient estimate $\nabla\Phi(\mathbf{x}) : \mathbb{R}^{2D} \rightarrow \mathbb{R}^{D\times 2D}$ when performing the linearization,

$$\Phi(\mathbf{x}+\Delta\mathbf{x}) \approx \Phi(\mathbf{x}) + \nabla\Phi(\mathbf{x})\Delta\mathbf{x} \tag{3.8}$$

3.2.3 Gradient Estimation as Regression

One can view the problem of gradient estimation naively as solving the following regression problem,

$$\nabla\Phi(\mathbf{x}) = \arg\min_{\mathbf{J}} \sum_{\Delta\mathbf{x}\in\mathcal{P}} \eta\{\Phi(\mathbf{x}+\Delta\mathbf{x}) - \mathbf{J}\Delta\mathbf{x}\} \tag{3.9}$$

where $\mathcal{P}$ is the set of deformations over which we want to establish an approximately linear relationship between the representation $\Phi(\mathbf{x}+\Delta\mathbf{x})$ and the deformation vector $\Delta\mathbf{x}$. η is the objective function used for performing the regression, for example $\eta\{.\} = ||.||_2^2$ would result in least-squares regression. This gradient estimation step can be performed more efficiently by considering each coordinate in $\mathbf{x} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_D^T]^T$ to be independent of each other. This results in a set of KD regression problems,

$$\nabla\Phi_i^k(\mathbf{x}_i) = \arg\min_{\mathbf{J}} \sum_{\mathbf{d}\in\mathcal{L}} \{\Phi_i^k(\mathbf{x}_i+\mathbf{d}) - \mathbf{J}\mathbf{d}\}, \;\; \forall\, i = 1:D, \;\; k = 1:K \tag{3.10}$$

where $\nabla\Phi_i^k(\mathbf{x}_i) : \mathbb{R}^2 \rightarrow \mathbb{R}^1$, $\mathcal{L}$ is the local translation deformation set for each pixel coordinate (normally a small window of say 3×3 or 5×5 discrete pixel coordinates), D is the number of pixel coordinates and K is the number of

channels in the representation (e.g. for raw pixel intensities $K = 1$). We can then define $\nabla\Phi(\mathbf{x}) : \mathbb{R}^{2D} \rightarrow \mathbb{R}^{DK \times 2D}$ as,

$$\nabla\Phi(\mathbf{x}) = \begin{bmatrix} \nabla\Phi_1^1(\mathbf{x}_1) & & & \\ \vdots & & & \\ \nabla\Phi_1^K(\mathbf{x}_1) & & & \\ & \ddots & & \\ & & \nabla\Phi_D^1(\mathbf{x}_D) & \\ & & \vdots & \\ & & \nabla\Phi_D^K(\mathbf{x}_D) & \end{bmatrix} . \tag{3.11}$$

Of course, linear regression is not the only option for learning the gradient regressor. One could also consider using support vector regression (SVR) [24], which has better robustness to outliers. Intuitively, support vector regression predicts the gradient direction from a different weighted combination of pixels within a local region around the reference pixel. SVR has a clear performance advantage, with a commensurate increase in computation during training.

3.2.4 Gradient Estimation as Filtering

For a least-squares objective $\eta\{.\} = ||.||_2^2$ the solution to each gradient matrix function can be computed in closed form,

$$\nabla\Phi_i^k(\mathbf{x}_i) = \left(\sum_{\mathbf{d} \in \mathcal{L}} \mathbf{d}\mathbf{d}^T \right)^{-1} \left(\sum_{\mathbf{d} \in \mathcal{L}} \mathbf{d}[\Phi_i^k(\mathbf{x}_i) - \Phi_i^k(\mathbf{x}_i + \mathbf{d})] \right) . \tag{3.12}$$

There are a number of interesting things to observe about this formulation. The first term in the solution is independent of the representation – it depends only on the local deformations sampled, and so can be inverted once rather than for each Φ_i^k. The second term is simply a sum of weighted differences between a displaced pixel, and the reference pixel, *i.e.*,

$$\begin{bmatrix} \sum_{\Delta x} \sum_{\Delta y} \Delta x (\Phi_i^k(x_i + \Delta x, y_i + \Delta y) - \Phi_i^k(x, y) \\ \sum_{\Delta x} \sum_{\Delta y} \Delta y (\Phi_i^k(x_i + \Delta x, y_i + \Delta y) - \Phi_i^k(x, y) \end{bmatrix} . \tag{3.13}$$

If $\mathbf{d} = [\Delta x, \Delta y]^T$ is sampled on a regular grid at integer pixel locations, Equation 3.13 can be cast as two filters – one each for horizontal weights Δx, and

vertical weights Δy,

$$f_x = \begin{bmatrix} x_{-n} & \dots & x_n \\ \vdots & & \\ x_{-n} & \dots & x_n \end{bmatrix} \qquad f_y = \begin{bmatrix} y_{-n} & \dots & y_{-n} \\ \vdots & & \\ y_n & \dots & y_n \end{bmatrix} \tag{3.14}$$

Thus, an efficient realization of Equation 3.12 of the gradient at every pixel coordinate is,

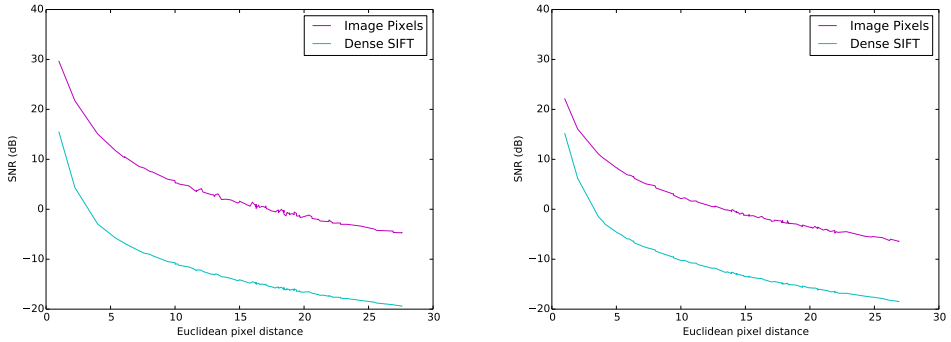
$$\nabla \Phi_i^k(\mathbf{x}_i) = \left(\sum_{\mathbf{d} \in \mathcal{L}} \mathbf{d}\mathbf{d}^T \right)^{-1} \text{diag}\Big(\begin{bmatrix} f_x * \Phi_i^k(\mathbf{x}) \\ f_y * \Phi_i^k(\mathbf{x}) \end{bmatrix} \Big) \tag{3.15}$$

where $*$ is the $2D$ convolution operator. This is equivalent to blurring the image with a clipped quadratic and then taking the derivative. It is also possible to place weights on $\mathbf{d}$ stemming from $\mathcal{L}$ as a function of its distance from the origin. In the case of Gaussian weights this results in the classical approach to estimating image gradients by blurring the representation with a Gaussian and taking central differences. It is surprising that the two formulations make opposing assumptions on the importance of pixels, and as we show in our experiments section the clipped quadratic kernel induced by linear regression is better for alignment.

3.2.5 Pixels versus V1-Inspired Features

Considerable literature has been devoted to finding image features for general object classes that are discriminative of image semantics whilst being tolerant to local image contrast and geometric variation. Recall that many of these encode non-linear combinations of pixels in local support regions, multi-channel outputs, and sparsity. Prominent image features that exhibit these properties include HOG [22] and densely sampled SIFT descriptors [56].

Natural images are known to stem from a $\frac{1}{f}$ frequency spectrum [81]. This means that most of the energy in the image is concentrated in the lower frequencies – the image function is naturally smooth. Sparse multi-channel features follow no such statistics. In fact, they often exhibit highly non-linear properties: small changes in the input can sometimes produce large changes in the output (*e.g.* gradient orientations close to a quantization boundary in HOG/SIFT can cause the output feature to jump channels, pixel differences close to zero in binary features can cause the output feature to swap signs), and other times



(a) (b)

Figure 3.1: An experiment to illustrate the generative ability of pixel and densely sampled sparse features (in this case dense SIFT). We compute the linearization error $\Phi(\mathbf{x}) + \nabla\Phi(\mathbf{x})\Delta\mathbf{x} - \Phi(\mathbf{x} + \Delta\mathbf{x})$ for a range of $\Delta\mathbf{x}$ (x-axis), and look at the resulting signal-to-noise ratio (SNR) on the y-axis. The results are averaged over 10000 random trials across 100 images drawn from a set of (a) faces, and (b) animals. As expected, the generative accuracy of pixels is consistently higher than densely sampled sparse features, and better for face imagery than animal+background imagery (though the sparse representation is largely unchanged).

produce no change in the output (*e.g.* orientations in the center of a bin, pixel differences far from zero).

To evaluate the generative capacity of different representations (*i.e.* how well the tangent approximation predicts the true image function at increasing displacements) we performed a simple experiment. We evaluated the signal-to-noise (SNR) ratio of the linearization function $\nabla\Phi(\mathbf{x})$ for increasing displacements across a number of images,

$$\mathrm{SNR}(\mathbf{x}) = 10\log_{10}\left(\frac{||\Phi(\mathbf{x}+\Delta\mathbf{x})||}{||\Phi(\mathbf{x}) + \nabla\Phi(\mathbf{x})\Delta\mathbf{x} - \Phi(\mathbf{x}+\Delta\mathbf{x})||}\right)^2. \tag{3.16}$$

For simplicity, we restricted the deformation vectors $\Delta\mathbf{x}$ to global translation. Figure 3.1 illustrates the signal-to-noise ratio (SNR) versus Euclidean distance (*i.e.* $||\Delta\mathbf{x}||_2$) for images of (a) faces, and (b) animals.

The tangent to the pixel image is a consistently better predictor of image appearance than the same applied to sparse features (in this case, dense SIFT). This confirms the intuition that pixel images are smoothly varying, whereas non-linear multi-channel features are not. Intuitively, this would suggest that sparse features would not be appropriate for gradient-based alignment search strategies. Unsurprisingly, graph-based optimization have become the strategy

of choice for alignment when using sparse features, with some notable exceptions [73, 94, 97]. As a result, the wealth of research into continuous alignment methods [20, 37, 57, 77, 107] has largely been overlooked by the broader vision community for general object alignment.

3.3 Graph Based Alignment

In the previous section, we continued with the tradition of using stationary matching functions (*i.e.* functions of the form $K(x, z) = k(x - z)$) to define the similarity between two pixels or feature vectors. This was useful in terms of being able to define gradient directions on the inputs. In this section, we explore the possibility of instead learning a discriminative detector at every pixel coordinate in an image.

Motivated by object detection literature, we learn a linear classifier per pixel in the reference image and apply it in a sliding-window manner to the target image to produce a match likelihood estimate. Learning a multitude of linear detectors such as exemplar support vector machines (SVMs) has typically had two issues: each detector must parse the negative set, often with hard-negative mining techniques, leading to long training times, which makes training a classifier for every pixel in an image intractable, and since the scale of the outputs depends on the margin, the output confidences of two different SVMs are not directly comparable.

We leverage recent work on learning detectors quickly with linear discriminant analysis (LDA), by collecting negative statistics across a large number of images in a pre-training phase. Learning a new exemplar detector then involves a single matrix-vector multiplication. Since LDA uses a generative model of the class distributions, the posterior probabilities provide a quantity that is comparable between detectors. This allows us to estimate both the likelihood of matches for each pixel individually, and also a global belief of match quality.

SIFT Flow [55] adopts a unary of the form,

$$f_i(\mathbf{x}_i) = \mathbf{h}(i, \mathbf{x}_i) = ||\Phi_A(i) - \Phi_B(\mathbf{x}_i)||_1 \tag{3.17}$$

where $\Phi_A(\mathbf{x}_i) = \Phi(\mathbf{x}_i; \mathcal{I}_A)$ is a feature representation of the image $\mathcal{I}_A$ evaluated at the point $\mathbf{x}_i$.[1]

[1] For our LDA classifiers, we extract features from a window of pixels around $\mathbf{x}_i$, but this detail can

In [48], the L_1 norm on the difference between features is replaced with a more general learned representation,

$$\mathbf{h}(i, \mathbf{x}_i) = H(\Phi_A(i) - \Phi_B(\mathbf{x}_i)) \tag{3.18}$$

In both formulations, however, the unary function is a stationary kernel. Stationary, or translation-invariant, functions define their output only in terms of the *difference* of the inputs. For two features to have high similarity, they must be similarly colocated in space. Finding such a feature embedding is a difficult task in general, and as a result significant object detection literature has focussed on learning classifiers to distinguish classes instead.

The use of classifiers has two distinct advantages over stationary kernels for describing match likelihood. First, linear classifiers define half-spaces in which samples are either classified as positive or negative. Thus two points with dissimilar appearances can still be afforded a high match likelihood. Second, absolute position of points in space can influence the classification decision.

In this section, we advocate a unary function of the form,

$$f_i(\mathbf{x}_i) = \mathbf{h}(i, \mathbf{x}_i) = \mathbf{w}_A(i)^T \Phi_B(\mathbf{x}_i) \tag{3.19}$$

where $\mathbf{w}_A(i)$ is a linear classifier trained to predict correspondences to pixel i in $\mathcal{I}_A$, with ideal response,

$$\mathbf{w}_A(i)^T \Phi_B(\mathbf{x}_i) = \begin{cases} 1 & \mathbf{x}_i = \mathbf{x}_i^* \\ -1 & \text{otherwise} \end{cases} \tag{3.20}$$

This is traditional binary classification, where the positive class contains the reference pixel, and its true correspondence in the target image, and the negative class contains all other pixels. Since the correspondence in the target image is not known *a priori* however, we rely on the classifier $\mathbf{w}_A(i)$ to generalize from a single training example: $\Phi_A(i)$. This is known as exemplar-based classification [60].

The challenge is how to rapidly estimate thousands of exemplar classifiers per image in reasonable time. The remainder of this section focuses on addressing that challenge, and a number of interesting properties that arise from our approach.

---

be subsumed into the feature transform Φ.

3.3.1 Learning Detectors Rapidly Using Structured Covariance Matrices

Linear classifiers have a rich history in computer vision, not least because of their interpretation and efficient implementation as a convolution operation. Support vector machines (SVMs) have proven particularly popular, due to their elegant theoretical interpretation, and impressive real-world performance, especially on object and part detection tasks. A challenge for any object detection problem is how to treat the potentially infinite negative set (comprising all incorrect correspondences in our case). Object detection methods using support vector machines employ hard negative mining strategies to search the negative set for difficult examples, which can be represented parametrically in terms of the decision hyperplane. This feature is also their limitation for rapid estimation of many classifiers, since each classifier must reparse the negative set looking for hard examples – knowing one classifier does not help in estimating another.[2]

Linear Discriminant Analysis (LDA), on the other hand, summarizes the negative set into its mean and covariance. The parameters $\mathbf{w}$ of the decision hyperplane $\mathbf{w}^T\mathbf{x} = c$ are learned by solving the system of equations,

$$\mathbf{S}\mathbf{w} = \mathbf{b} \tag{3.21}$$

where $\mathbf{S}$ is the joint covariance of both classes and $\mathbf{b} = \mathbf{u}_{\text{pos}} - \mathbf{u}_{\text{neg}}$ is the difference between class means. Hariharan *et al.* [34] made two key observations about LDA: first, if the number of positive examples is small compared to the number of negative examples, the joint covariance $\mathbf{S}$ can be approximated by the covariance of the negative distribution alone, and reused for all positive classes, and second, gathering and storing the covariance can be performed efficiently if the negative class is shift invariant (*i.e.* a translated negative example is still a negative example).

This second fact implies stationarity of the negative distribution, where the covariance of two pixels is defined entirely by their relative displacement. Importantly, both [34] and [35] showed that the performance of linear detectors learned by exploiting the stationarity of the negative set is comparable to SVM training with hard negative mining.

[2] This is not strictly true. Warm starting an SVM from a previous solution, especially in exemplar SVMs where only a single positive example changes, can induce a significant empirical speedup, however is unlikely to change the $O()$ complexity of the algorithm.

The covariance $\mathbf{S}$ can be constructed from a relative displacement tensor, according to,

$$\mathbf{S}_{(u,v,p),(i,j,q)} = g[i-u, j-v, p, q] \tag{3.22}$$

where i, j, u, v index spatial coordinates, and p, q index channels. We call the maximum displacement observed $\text{abs}(i-u)$, $\text{abs}(j-v)$ the bandwidth of the tensor. Also note that stationarity only exists spatially – cross-channel correlations are stored explicitly. The storage of g thus scales quadratically in both bandwidth and channels, though since the detectors we consider are typically small-support, we can entertain feature representations with large numbers of channels (*i.e.* SIFT).

In order to compute g, we gather statistics across a random subset of $50,000$ images from ImageNet. We precompute the covariance matrix of the chosen detector size (typically 5×5) and factor it with either a Cholesky decomposition, or its explicit inverse. Since the sample covariance is estimated from missing data, we make sure it is positive-definite by adding the minimum of zero and the minimum eigenvalue to the diagonal, *i.e.* $(\mathbf{S} - \min(0, \lambda_{\min}) \cdot \mathbf{I})^{-1}$.

For each pixel in the reference image, we compute,

$$\mathbf{w}_A(i) = \mathbf{S}^{-1}(\mathbf{u}_{\text{pos}} - \mathbf{u}_{\text{neg}}) \tag{3.23}$$

which involves a single substraction and matrix-vector multiplication, where,

$$\mathbf{u}_{\text{pos}} = \Phi_A(\mathbf{x}_i) \tag{3.24}$$

The likelihood estimate for the i-th reference point across the target image can be performed via convolution over the discretize pixel grid,

$$f_i(\mathbf{x}) = \mathbf{w}_A(i) * \Phi_B(\mathbf{x}) \tag{3.25}$$

Since storing the full unary is quadratic in the number of image pixels (quartic in the dimension), we perform coarse-to-fine or windowed search as per SIFT Flow [55].

3.3.2 Posterior Probability Estimation

Linear Discriminant Analysis (LDA) has the attractive property of generatively modelling classes as Gaussian distributions with equal (co-)variance. This per-

mits direct computation of posterior probabilities via application of Bayes' Rule:

$$p(C_{\text{pos}}|\mathbf{x}) = \frac{p(\mathbf{x}|C_{\text{pos}})\ p(C_{\text{pos}})}{\sum_{n\in\{\text{pos,neg}\}} p(\mathbf{x}|C_n)\ p(C_n)} \tag{3.26}$$

where,

$$p(\mathbf{x}|C_n) = \frac{1}{(2\pi)|\mathbf{S}|^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\mathbf{u}_n)^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{u}_n)} \tag{3.27}$$

With some manipulation, the posterior of Equation 3.26 can be expressed as,

$$p(C_{\text{pos}}|\mathbf{x}) = \frac{1}{1+\mathrm{e}^{-\mathbf{y}}} \tag{3.28}$$

$$\mathbf{y} = \mathbf{x}^T\mathbf{S}^{-1}(\mathbf{u}_{\text{pos}} - \mathbf{u}_{\text{neg}}) \tag{3.29}$$

$$+ \tfrac{1}{2}\mathbf{u}_{\text{pos}}^T\mathbf{S}^{-1}\mathbf{u}_{\text{pos}} - \tfrac{1}{2}\mathbf{u}_{\text{neg}}^T\mathbf{S}^{-1}\mathbf{u}_{\text{neg}} \tag{3.30}$$

$$+ \ln\left(\frac{p(C_{\text{pos}})}{p(C_{\text{neg}})}\right) \tag{3.31}$$

Equation 3.28 takes the form of a logistic function, which maps the domain $(-\infty \ldots \infty)$ to the range $(0 \ldots 1)$.

The logistic function is typically used to convert SVM outputs to probabilistic estimates, however a "calibration" phase is required to learn the bias and variance of each SVM in the ensemble so their outputs are comparable. With LDA, these parameters are derived directly from the underlying distributions.

Equation 3.29 is the canonical response to the LDA classifier, Equation 3.30 represents the bias of the distributions, and Equation 3.31 is the ratio of prior probabilities of the classes. This must be determined by cross-validation (once, not for each classifier), based on the desired sensitivity to true versus false positives.

By completing the squares in Equation 3.30, we yield the final expression for computing the posterior probability,

$$\begin{aligned}\mathbf{y} &= (\mathbf{x} - \tfrac{1}{2}(\mathbf{u}_{\text{pos}} + \mathbf{u}_{\text{neg}}))^T\mathbf{S}^{-1}(\mathbf{u}_{\text{pos}} - \mathbf{u}_{\text{neg}}) + \mu \\ &= (\mathbf{x} - \tfrac{1}{2}(\mathbf{u}_{\text{pos}} + \mathbf{u}_{\text{neg}}))^T\mathbf{w}_i + \mu \end{aligned} \tag{3.32}$$

The implication of Equation 3.32 is that it is no more expensive to compute probability estimates than to just evaluate the classifier – the computation is still dominated by the single matrix-vector product required to learn the

Figure 3.2: From left to right: (a) reference image with reference point labelled in red, and posterior estimates for (b) LDA and (c) L_1 norm. We present a range of points, from distinctive to indistinctive or background. LDA and L_1 norm have similar likelihood quality for distinctive points, but LDA consistently offers better rejection of incorrect matches and background content.

classifier.

Figure 3.2 illustrates a representative set of likelihood estimates output by our method and SIFT Flow respectively. LDA typically has tighter responses around the true correspondence, and better suppression of false positives, especially on background content that has no clear correspondence.

3.4 Experiments

In order to evaluate the efficacy of our method, we first wanted to understand how well human annotators perform at semantic labelling tasks. Since we are primarily interested in estimating correspondences for reconstruction-type objectives, we gathered 20 pairs of images from visual object categories which exhibit anatomical correspondence, including an assortment of animals, trucks, faces and people. Given a set of sparsely selected keypoints in the first image of each pair, 8 human annotators were tasked with labelling the correspond-

Figure 3.3: A representative subset of the groundtruth dataset. From top to bottom: (a) the source images, (b) the target images, and (c) the distribution of points selected by the human annotators on the target images. The structure of the object is often clearly discernible from the annotations alone.

ing points in the second image. A representative subset of the data is shown in Figure 3.3.

A similar experiment was performed in [55], however they focussed on correspondences across *scenes*, which often have no clear correspondence, even for human annotators. In contrast, the agreement on our dataset is high, with a natural increase in uncertainty from corner features, to edges and textureless regions.

In recognizing that not all features are equally distinctive, we measure distance from estimated points $\mathbf{x}_i$ to the groundtruth using Mahalanobis distance,

$$d_i(\mathbf{x}_i) = \sqrt{(\mathbf{x}_i - \mu_i)^T \mathbf{S}_i^{-1} (\mathbf{x}_i - \mu_i)} \tag{3.33}$$

where μ_i and $\mathbf{S}_i$ are the $2D$ mean and covariance of the groundtruth labellings across annotators. Tompson *et al.* motivate a similar procedure for human pose estimation [86]. This metric has two advantages over Euclidean distance: (i) it takes into account spatial and directional uncertainty (*e.g.* correspondences are afforded some slack along an edge, but not perpendicular to it), and (ii) it is resolution independent, since distance is measured in standard deviations.

Our dataset and metric therefore sets a higher standard for what is considered a good correspondence, both empirically and qualitatively (since readers can accurately discriminate good from poor results). All results presented in the following section are measured under this metric.

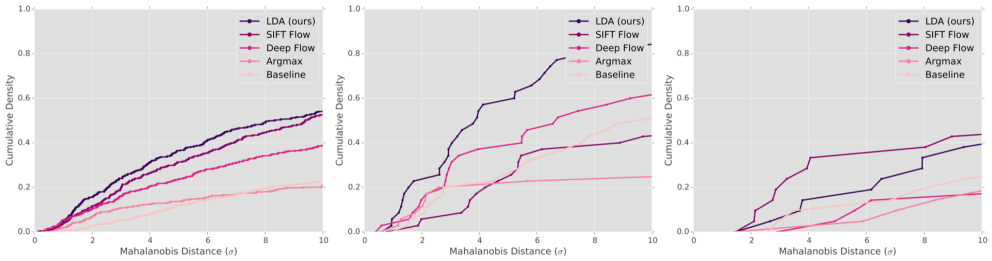


Figure 3.4: Comparison of sparse keypoint localization for our method, SIFT Flow [55] and Deep Flow [95]. The baseline measures the global alignment bias of the dataset (how well one would perform by simply assuming no flow). The argmax considers taking the single best match without regularization. The graphs measure the fraction of correspondences which fall within an increasing distance from groundtruth. Three standard deviations is inperceptible from human annotator accuracy. From left to right: (a) aggregate results across all images, (b) the truck pair which our method localizes well, and (c) the biking pair for which our method fails to produce any meaningful correspondences.

3.4.1 Pairwise Graph Based Alignment

In all of our experiments we resize the source (A) and target (B) image so $\max(M, N) = 150$, preserving the aspect ratio, and extract densely sampled SIFT features.

The stationary distribution (mean and covariance) of SIFT features is estimated from $50,000$ randomly sampled images from ImageNet. Classifiers with spatial support 1×1, 3×3, 5×5, 7×7 and 9×9 were evaluated. The different sizes tradeoff speed, localization accuracy and generalization. We found 5×5 classifiers provided a good balance between these tradeoffs, and the results throughout this chapter use this support.

While the LDA likelihoods are more computationally demanding to compute than L_1-norm likelihoods, the construction and application of the classifiers can be accelerated with BLAS. Estimating $10,000$ 5×5 classifiers and applying them in a sliding window fashion to a 80×125 SIFT image (with 128 channels) takes approximately 6 seconds.

We apply our LDA-based correspondence method in the same graphical model framework as SIFT Flow. We use a coarse-to-fine scheme to handle inference over larger images, and grid searched the hyperparameters for both LDA and L_1 based unary functions. Results are shown in Figure 3.4.

We display the cumulative density for increasing number of standard deviations from groundtruth (*i.e.* fraction of points falling within an increasing

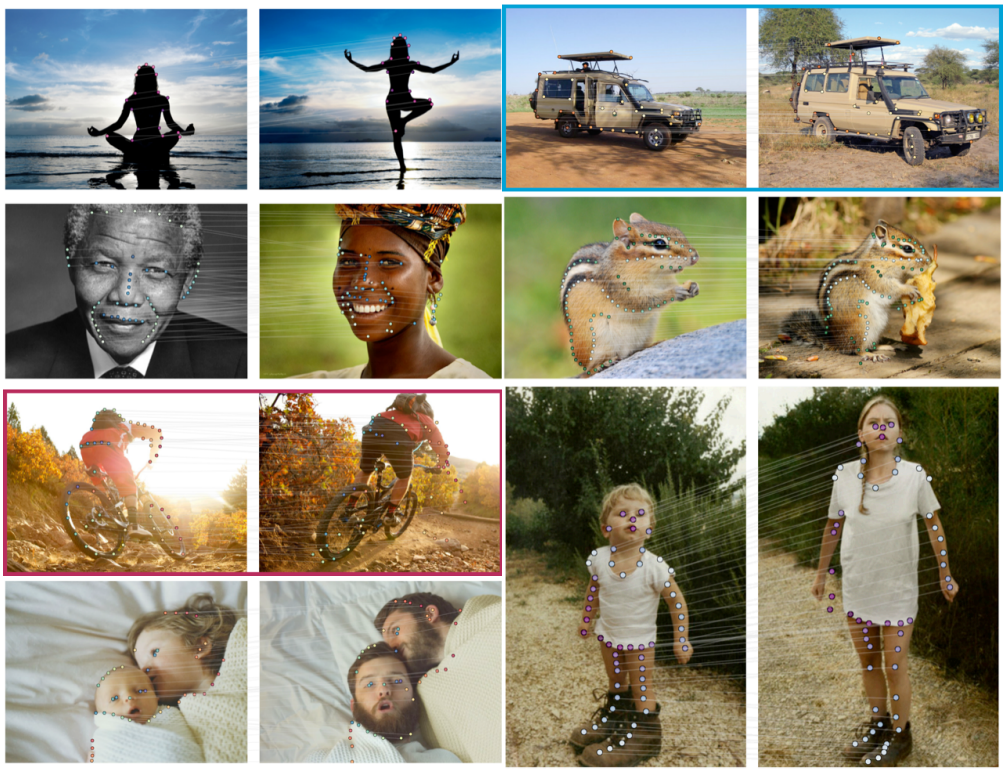


Figure 3.5: Example correspondences discovered by our method, across a broad range of image pairs from our dataset. The truck pair produces good localization of points (see Figure 3.4b), whilst the biking pair shows a failure to produce anything meaningful (see Figure 3.4c).

radius from groundtruth). As a baseline, we simply set $\mathbf{x}_i = i$,[3] which acts as a proxy to the global alignment bias of the dataset (small flow assumption). In addition to SIFT Flow, we also compare our method to a leading optical flow method, Deep Flow [95].

We truncate the CDF due to the long tails for all methods compared. This is an artefact of the non-global regularization schemes, which allow some points to be arbitrarily far from groundtruth without affecting others. Finally, in Figure 3.5 we illustrate a number of exemplar correspondences to show the visual quality of matches produced by our method.

[3] For images of different sizes, we set $\mathbf{x}_i = \mathcal{W}(i)$ where $\mathcal{W}$ is a function that maps the span of $\mathcal{I}_A$ to $\mathcal{I}_B$.

3.4.2 Pairwise Gradient Based Alignment

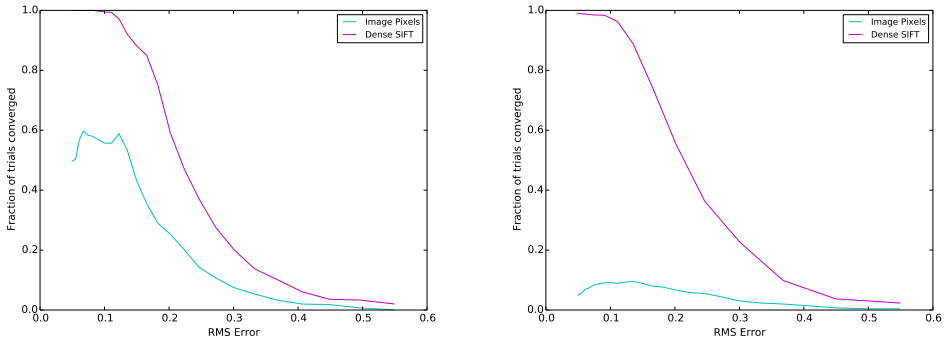
Earlier in Figure 3.1 we performed a synthetic experiment showing the linearization error as a function of displacement for different image representations. Here we perform the sequel to that experiment, showing the frequency of convergence of the LK algorithm as a function of initial misalignment.

We initialize a bounding box template within an image, then perturb its location by a given RMS point error (measured from the vertices of the bounding box) and measure the frequency with which the perturbed patch converges back to the initialization after running LK. The results are show in Figure 3.6. We perform two variants of the experiment, (a) *intra*-image alignment, where the template and perturbation are sampled from the same image, and (b) *inter*-image alignment, where the perturbation is sampled from a different image of the same object class, with known ground-truth alignment. The task of inter-image alignment is markedly more difficult, since the objects within the template and the perturbation may have different non-rigid geometry, scene lighting and background clutter.

Even in the intra-image scenario, dense SIFT consistently converges more frequently than pixel intensities. In the inter-image scenario, the difference is even more pronounced. Figure 3.7 shows a more comprehensive view of the inter-image scenario, with a comparison of the different gradient estimation techniques we have discussed. In general, there is a gradual degradation in performance from support vector regression (SVR) to least squares regression to central differences. The *domain* in the legend specifies the blur kernel size in the case of central differences, or the support region over which training examples are gathered for regression. Figure 3.8 illustrates the type of imagery on which we evaluated the different methods – animal classes drawn from the ImageNet dataset, often exhibiting large variations in pose, rotation, scale and translation.

3.4.3 Ensemble Alignment

We finish the piece with the challenging real-world application of ensemble alignment. The task of ensemble alignment is to discover the appearance of an object of interest in a corpus of images in an unsupervised manner. Discrete approaches are currently unsuitable to this problem because searching over translation and scale alone is insufficient for good alignment, and exploring higher-dimensional warps using discrete methods is either infeasible or compu-



(a) (b)

Figure 3.6: An experiment to illustrate the alignment performance of pixel intensities versus densely sampled sparse features (in this case densely extracted SIFT descriptors). In both scenarios, we initialize a bounding box within an image, then perturb its location by a given RMS point error (x-axis) and measure the frequency with which the perturbed patch converges back to the initialization (y-axis). In (a) we perform *intra*-image alignment, where the template and perturbation are sampled from the same image. In (b) we perform *inter*-image alignment, where the perturbation is sampled from a different image of the same object class with known ground-truth alignment. The task of inter-image alignment is markedly more difficult, since the two objects being aligned may be experiencing different lightning and pose conditions. The drop in pixel performance is more pronounced than dense SIFT when moving to the harder task.

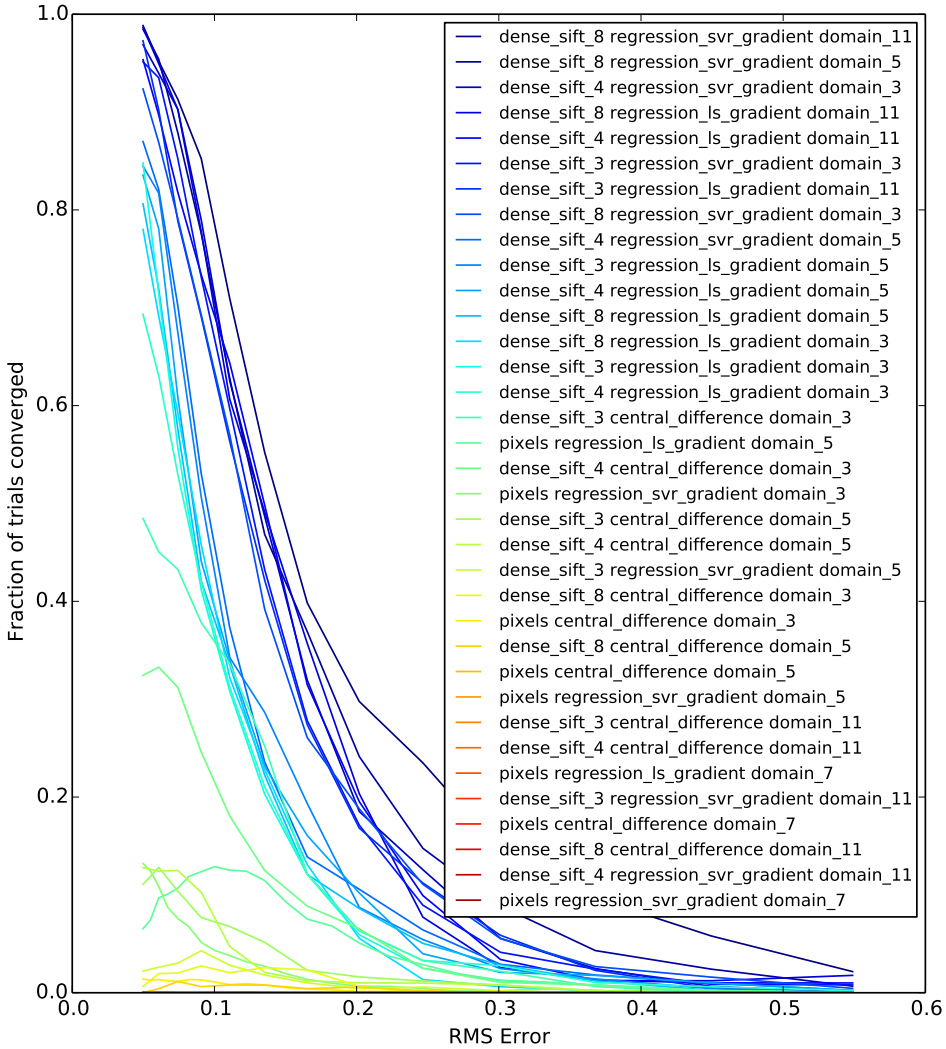


Figure 3.7: *Inter*-image alignment performance. We initialize a bounding box within the image, then perturb its location by a given RMS point error (x axis), run Lucas Kanade on the resulting patch, and measure the frequency with which the patch converges back to the initialization (y axis). The *domain* specifies the Gaussian standard deviation in the case of central differences, or the maximum displacement from which training examples are gathered for regression. On dense SIFT, there is a progressive degradation in performance from SVR to least-squares regression to central differences. Pixel intensities (using any gradient representation) perform significantly worse than the top dense SIFT based approaches.

(a) (b)

Figure 3.8: Representative pairwise alignments. (a) is the template region of interest, and (b) is the predicted region that best aligns the image to the template. The exemplars shown here all used dense SIFT features and least squares regression to learn the descent directions. The four examples exhibit robustness to changes in pose, rotation, scale and translation, respectively.

tationally challenging.

We present results using a gradient-based approach called least squares congealing [21]. The details of the algorithm are not essential to our discussion, however it features the same linearization as the LK algorithm, and as such is subject to the same properties we have discussed throughout this chapter.

Figure 3.9 show the results of aligning a subset of 170 elephants drawn from the ImageNet dataset,[4] using dense SIFT features and least squares regression, parametrized on a similarity warp. The same set-up using pixel intensities failed to produce any meaningful alignment. Figure 3.10 shows the mean of the image stack before and after congealing. Even though individual elephants appear in different poses, the aligned mean clearly elicits an elephant silhouette.

3.5 Discussion

So far in this chapter we have presented the somewhat paradoxical result that densely sampled sparse features perform well in real-world alignment applications (Figure 3.6, Figure 3.7) whilst sporting poor tangent approximations (Figure 3.1). Here we try to offer some insight into why this might be the case.

Consider first the effect of convolving a sparse signal with a low-pass filter. We know from compressive-sensing that observed blurred signals can be recovered almost exactly if the underlying signal is sparse [87]. Unlike traditional dense pixel representations whose high-frequency information is attenuated when convolved with a low-pass filter, sparse signals can be blurred to a much larger extent without any information loss before reaching the limits of sampling theory. Figure 3.11 illustrates the effect of comparing dense and sparse signals as the degree of misalignment and blur increases.

The immediate implication of this for image alignment is that a sparse multi-channel representation can be blurred to dilate the convergent region whilst preserving information content. The encoding of local pixel interactions ensures this information content contains high-frequency detail required for good alignment.

We would be remiss not to mention the rise of convolutional network approaches in the context of this work. The work of DeepFlow [95] aimed to

[4] We removed those elephants whose out-of-plane rotation from the mean image could not be reasonably captured by an affine warp. The requirement of a single basis is a known limitation of the congealing algorithm.

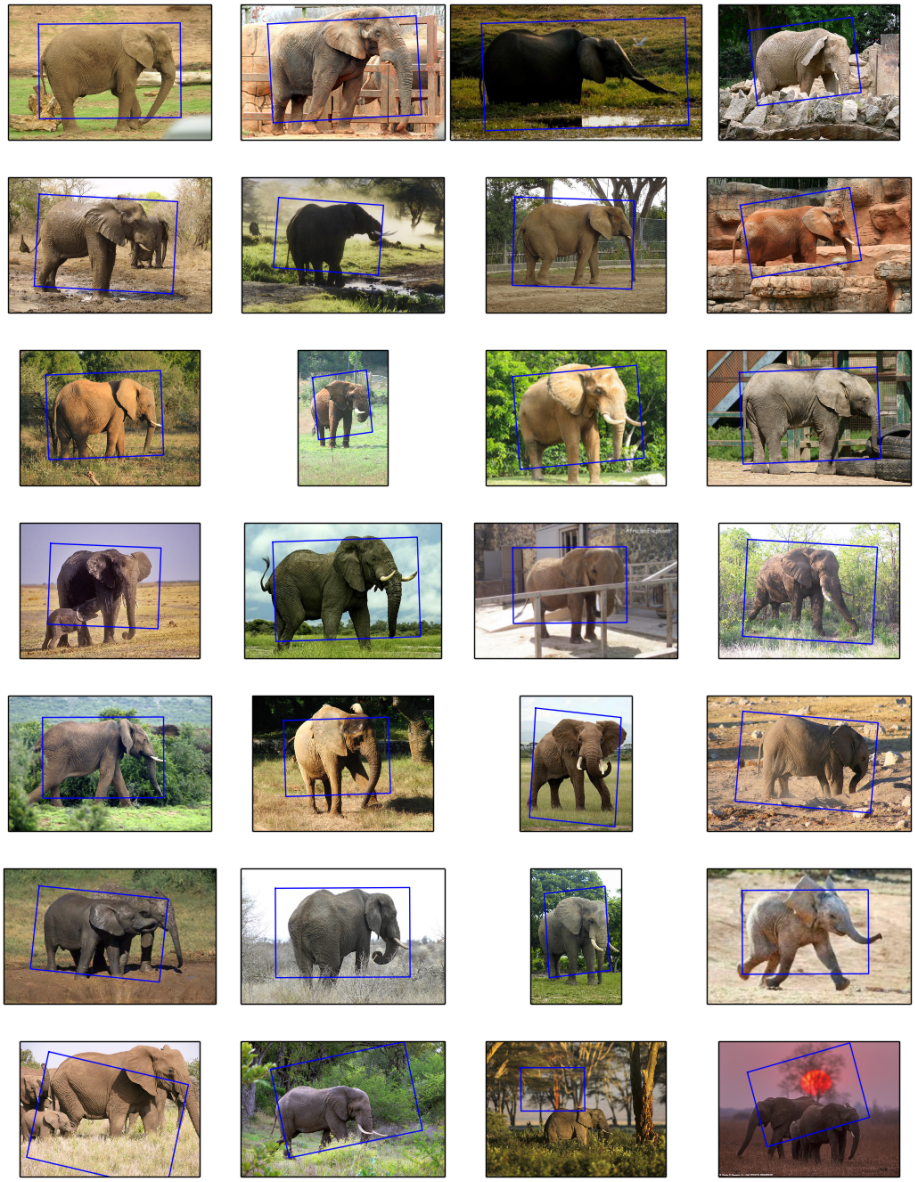
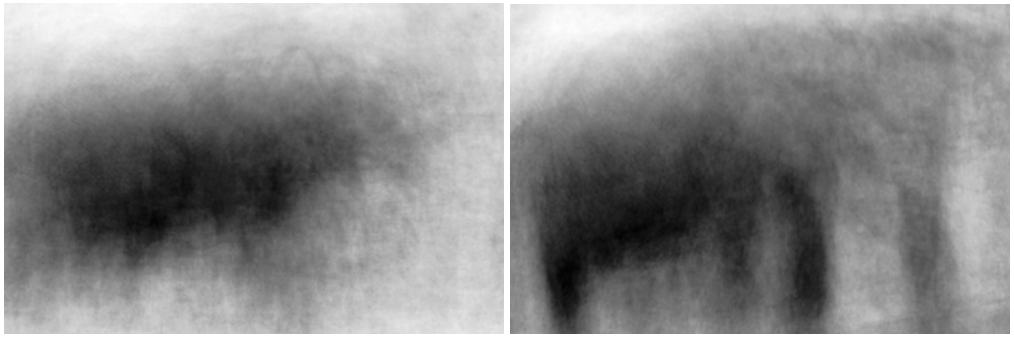
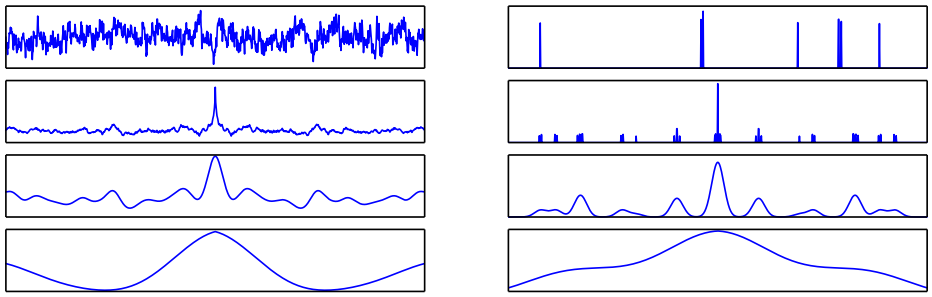


Figure 3.9: Unsupervised ensemble alignment (congealing) on a set of 170 elephants taken from ImageNet. The objective is to jointly minimize the appearance difference between all of the images in a least-squares sense – no prior appearance or geometric information is used. The first 6 rows present exemplar images from the set that converged. The final row presents a number of failure cases.



(a) (b)

Figure 3.10: The mean image of Figure 3.9 (a) before alignment, and (b) after alignment with respect to a similarity warp. Although individual elephants undergo different non-rigid deformations, one can make out an elephant silhouette in the aligned mean.



(a) (b)

Figure 3.11: A $1D$ alignment thought experiment. The first row shows two signals: a dense signal with a $\frac{1}{f}$ frequency spectrum, and a sparse positive signal. The second, third and fourth rows show the negative auto-correlation of the signals to simulate the expected loss for varying degrees of misalignment (x-axis) with increasing amounts of Gaussian blur applied to the original signals (row-wise). The red circles represent a hypothetical initialization of the parameters (in this case x-translation), the green squares represent the global optima, and the arrows indicate the direction of steepest descent. For the given initialization, gradient-based alignment on the dense signal will never converge to the true optima. Even with a large amount of blur applied, the solution is divergent (the gradient of the cross-correlation is moving away from the optima). The sparse signal, on the other hand, can tolerate a larger amount of blur and still maintain the location of the optima, in this case converging with the greatest amount of blur applied. This illustrates the importance of sparse, positive representations when matching misaligned signals. In order to retain discriminative appearance information, modern features use *multi-channel*, sparse, positive representations – but the basic concept remains.

apply convolutional networks to the problem of large displacement optical flow. They apply their method to the Sintel dataset – a synthetic rendered dataset for which flow vectors are known exactly. A subset of this dataset is used for training. Results are state of the art.

True semantic correspondence like that in SIFT Flow [55], FlowWeb [103] or the work presented here is still incredibly challenging for convolutional networks due to the dearth of training data available. Convolutional network literature in the vision community seems to remain firmly planted in strong supervision, and the types of problems that afford large datasets. In Figure 3.4 we compare our method to the DeepFlow work, and it understandably fails to generalize from the optical flow problem to the significantly less constrained semantic correspondence problem.

In regards to features, both SIFT features in my own work and convolutional networks pool statistics over local spatial regions. Preliminary work on replacing SIFT features with Hypercolumn features from a pretrained network suggests that long-range matching performance improves at the expense of fine-grained localization (as might be expected for a network with large receptive field). Regardless of underlying feature representation, using LDA to effectively "tune" the relative importance of each dimension is superior to using a uniform metric such as truncated L1.

3.6 Conclusion

Image alignment underlies many important computer vision and learning tasks. Recently, there has been significant interest on *semantic* alignment, which considers images that stem from the same visual class. In this chapter, we considered two approaches to the problem of semantic correspondence, and illustrated the strengths and weaknesses of both.

Following our theme of representation in computer vision, we illustrated how the use of V1-inspired representations alone can boost the performance of the matching function. We also showed, however, that such representations can be put to even better use by learning their statistical structure from a large unlabelled training set.

While a large body of research has focussed on gradient-based alignment strategies in the facial domain, they have rarely been applied to broader object categories. For general objects, alignment in pixel space performs poorly because low frequency information in the signal is dominated by lighting varia-

tion. Densely sampled sparse features provide tolerance to local image contrast variation, at the expense of reducing the range over which tangent approximations to the image function are accurate. As a result, graphical models have become the preferred approach to alignment when using densely sampled sparse features. We showed the surprising result that although the tangent approximation is poor, the real-world results when using image features are impressive. We offered some insights into why this may be the case, along with a number of approaches for estimating the descent directions.

Secondly, in contrast to existing correspondence methods, which typically use similarity kernels, we proposed using exemplar classifiers for describing the likelihood of two points matching. We showed that LDA classifiers exhibit 3 desirable properties: (i) higher average precision than simple measures of image similarity such as the L_1 norm, (ii) significantly faster training than exemplar SVMs, and (iii) estimates of match confidence that are directly comparable across pixels.

We presented a small semantic correspondence dataset and metric in a bid to measure the performance of different methods in a quantifiable manner, and showed that under this metric our classifier-based approach offered improvements over the L_1 norm, within the same SIFT Flow optimization framework. The qualitative results illustrate our method's ability to estimate high-quality dense semantic correspondences.

3.7 Future Work

A longstanding drawback of non-rigid structure from motion (NRS*f*M) methods has been their reliance on synthetic data. Unannotated real-world data is difficult to use because traditional correspondence methods based on an underlying rigid assumption of the world breakdown when applied to non-rigid geometry. This work on pairwise semantic correspondence was an initial foray into understanding whether non-rigid correspondences could be improved by applying a fast learning-based matching function. We found that the learned matching function had better recall than truncated $L1$ used by SIFT Flow, however overall correspondence performance was limited by the weak regularization and pairwise constraint. Around the same time, FlowWeb [103] was published, which aims to solve the semantic correspondence problem across a graph of images, however they rely on an underlying off-the-shelf optical flow method to seed correspondences in the graph.

Professor Simon Lucey has since received grant funding to further pursue his interests in NRSfM. A component of this project will be to solve for correspondences on a temporal graph, constrained directly by the structure from motion objective. This "solve it all at once" approach is designed to improve upon the weak regularization of my own work, whilst maintaining the advantages of the discriminative patch matching function, and adding structural regularization to a collection of images.

Conclusion

Visual recognition remains one of the core problems of computer vision. Despite its challenges, the community has made significant progress towards both practical domain-specific and general recognition algorithms. Much of this progress can be attributed to larger datasets, more computational resources and a better understanding of visual mechanics.

This thesis has looked primarily at that last aspect and how an understanding of image formation and statistical structure can be used to make better use of unlabelled training data. We focussed on core concepts involved in handling geometric variation in object classes, including *convolution*, *stationarity*, *sparsity*, *locality* and *capacity*.

In Chapter 1, we looked at the initial stages of image representation in the visual cortex, and how the behaviour observed is well explained by the sparse coding problem. Accounting for the stationarity of natural imagery and modelling this via the convolution operator, we demonstrated how *convolutional* sparse coding can be performed empirically fast, with numerous natural coding applications.

In Chapter 2, we considered the interaction of image representation with the choice of classifier, and characterized the type of pixel interactions required for learning geometric tolerance. We showed how the image representation can be partioned into the prior it encodes and the capacity it induces in the classifier.

Finally in Chapter 3, we showed how image representations that encode geometric tolerance are important for recovering geometry in unsupervised general object alignment tasks. Whilst the representations are typically non-smooth and non-differentiable, their sparse properties make them well-suited to gradient-based alignment. In the alignment objective, stationarity can once again be leveraged to summarize the statistics of natural images in order to

learn more robust matching functions based on image classification rather than traditional image similarity.

The scale of labelled training data now available has led to the re-emergence of high-capacity classification techniques, namely convolutional networks. These networks typically eschew complex priors or handcrafted image representations in favour of a fully-learned procedure, in order to capture latent structure in the problem that may be difficult to analytically describe.

The issue with this approach is that it requires significant amounts of *labelled* training data - current methods make poor or no use of unlabelled data. There also exists a gulf between the function that is representable by a given network, and the function that can actually be learned with current optimization techniques.

Developing a theory and understanding of visual mechanics will therefore continue to play an important role in efficiently using unlabelled data and steering optimization procedures to good solutions as the size and complexity of visual recognition algorithms continue to increase.

Bibliography

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast Retina Keypoint. *International Conference of Computer Vision and Pattern Recognition (CVPR)*, 2012. 7

[2] A. B. Ashraf, S. Lucey, and T. Chen. Reinterpreting the application of Gabor filters as a manipulation of the margin in linear support vector machines. *Pattern Analysis and Machine Learning*, 32(7):1335–41, jul 2010. 41, 47

[3] S. Avidan. Support vector tracking. *Pattern Analysis and Machine Intelligence (PAMI)*, 26(8):1064–72, aug 2004. 66

[4] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework: Part 1. *International Journal of Computer Vision (IJCV)*, 56(3):221–255, feb 2004. 70

[5] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework: Part 2. *International Journal of Computer Vision (IJCV)*, 2004. 70

[6] H. Barlow. Redundancy reduction revisited. *History*, 12:241–253, 2001. 4

[7] H. Bay, A. Ess, T. Tuytelaars, and L. Vangool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, jun 2008. 7

[8] A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 17, 22

[9] J. Bergstra, G. Desjardins, P. Lamblin, and Y. Bengio. Quadratic polynomials learn better image features. Technical report, Universite de Montreal, 2009. 42

[10] M. J. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision (IJCV)*, 26(1):63–84, 1998. 66

[11] L. Bo, X. Ren, and D. Fox. Kernel Descriptors for Visual Recognition. *Advances in Neural Information Processing Systems*, pages 1–9, 2010. 41

[12] S. Boyd. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010. 17, 20, 58

[13] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *European Conference on Computer Vision (ECCV)*, 4(May):25–36, 2004. 67

[14] T. Brox, J. Malik, and C. Bregler. Large displacement optical flow. *International Conference of Computer Vision and Pattern Recognition (CVPR)*, pages 41–48, 2009. 68

[15] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF : Binary Robust Independent Elementary Features. *European Conference on Computer Vision (ECCV)*, pages 778–792, 2010. 7

[16] E. J. Candes and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 2008. 7

[17] R. Chalasani and J. Principe. A fast proximal method for convolutional sparse coding. *International Joint Conference on Neural Networks (IJCNN)*, 2012. 17

[18] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007. 48

[19] B. Chen, G. Polatkan, G. Sapiro, and D. Blei. Deep Learning with Hierarchical Convolutional Factor Analysis. *Pattern Analysis and Machine Intelligence (PAMI)*, pages 1–30, 2013. 34

[20] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence (PAMI)*, 2001. 66, 74

[21] M. Cox, S. Sridharan, and S. Lucey. Least-squares congealing for large numbers of images. *International Conference on Computer Vision (ICCV)*, 2009. 87

[22] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *International Conference of Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005. 7, 46, 72

[23] J. J. DiCarlo, D. Zoccolan, and N. C. Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–34, mar 2012. 41

[24] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support Vector Regression Machines. *Advances in Neural Information Processing Systems (NIPS)*, (x):155–161, 1997. 71

[25] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008. 46

[26] P. F. Felzenszwalb. Representation and detection of deformable shapes. *Pattern Analysis and Machine Intelligence (PAMI)*, 27(2):208–20, feb 2005. 66

[27] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–45, sep 2010. 57

[28] D. Field and A. Olmos. Does spatial invariance result from insensitivity to change? *Journal of Vision*, 7(14):1–13, 2007. 2

[29] D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America. A, Optics and image science*, 4(12):2379–94, 1987. 6

[30] I. Fodor. A survey of dimension reduction techniques. Technical Report 1, Lawrence Livermore National Laboratory, 2002. 54

[31] D. Gabay and B. Mercier. A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite Element Approximations. *Computers and Mathematics with Applications*, 1976. 20

[32] R. Glowinski and A. Marroco. Sur L'Approximation, par Elements Finis d'Ordre Un, et la Resolution, par Penalisation-Dualite, d'une Classe de Problemes de Dirichlet non Lineares. *Revue Francaise d'Automatique*, 1975. 20

[33] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariant sparse coding for audio classification. *UAI*, 2007. 16, 17

[34] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. *European Conference on Computer Vision (ECCV)*, 1:1–14, 2012. 49, 76

[35] J. F. Henriques, J. Carreira, R. Caseiro, and J. Batista. Beyond Hard Negative Mining: Efficient Detector Learning via Block-Circulant Decom-

position. *International Conference on Computer Vision (ICCV)*, pages 2760–2767, dec 2013. 76

[36] G. E. Hinton. Learning multiple layers of representation. *Trends in Cognitive Sciences*, 11(10):428–34, oct 2007. 52

[37] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, aug 1981. 66, 67, 74

[38] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *International Conference on Machine Learning (ICML)*, number 2, pages 408–415, New York, New York, USA, 2008. ACM Press. 58

[39] D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106, 1962. 5

[40] A. Hyvärinen, J. Hurri, and P. O. Hoyer. Natural Image Statistics A probabilistic approach to early computational vision. *Computational Imaging and Vision*, 39, 2009. 32

[41] A. Hyvärinen and U. Köster. Complex cell pooling and the statistics of natural images. *Network*, 18(2):81–100, jun 2007. 42

[42] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? *International Conference on Computer Vision*, pages 2146–2153, sep 2009. 53

[43] K. Kavukcuoglu, M. Ranzato, and R. Fergus. Learning invariant features through topographic filter maps. *Computer Vision and Pattern Recognition (CVPR)*, pages 1605–1612, jun 2009. 42

[44] K. Kavukcuoglu, P. Sermanet, Y.-l. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning convolutional feature hierarchies for visual recognition. *Advances in Neural Information Processing Systems (NIPS)*, (1):1–9, 2010. 15, 34

[45] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *Pattern Analysis and Machine Intelligence (PAMI)*, 28(10):1568–1583, 2006. 67

[46] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1–9, 2011. 67

[47] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NIPS)*, pages 1–9, 2012. 34

[48] L. Ladicky, C. Häne, and M. Pollefeys. Learning the Matching Function. *arXiv preprint*, 2015. 67, 75

[49] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. *Advances in Neural Information Processing Systems*, 19:801, 2007. 16, 46

[50] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *International Conference on Machine Learning (ICML)*, pages 1–8, 2009. 17, 34

[51] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust invariant scalable keypoints. *International Conference on Computer Vision (ICCV)*, pages 2548–2555, 2011. 7

[52] M. Lewicki and T. Sejnowski. Coding time-varying signals using sparse, shift-invariant representations. *NIPS*, 1999. 13, 15, 16, 17

[53] Y. Li and D. Huttenlocher. Learning for optical flow using stochastic optimization. *European Conference on Computer Vision (ECCV)*, pages 379–391, 2008. 67

[54] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing*, 11(4):467–76, jan 2002. 54

[55] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *Pattern Analysis and Machine Intelligence (PAMI)*, 33(12):2368–2382, 2011. 67, 74, 77, 80, 81, 90

[56] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, nov 2004. 7, 72

[57] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence (IJCAI)*, 1981. 66, 67, 74

[58] P. Lucey, S. Lucey, and J. Cohn. Registration invariant representations for expression detection. *International Conference on Digital Image Computing: Techniques and Applications*, (i):255–261, 2010. 56

[59] J. Mairal, F. Bach, and J. Ponce. Online dictionary learning for sparse coding. *Conference on Machine Learning*, 2009. 27

[60] T. Malisiewicz, A. Gupta, and A. a. Efros. Ensemble of exemplar-svms for object detection and beyond. *International Conference on Computer Vision (ICCV)*, pages 89–96, nov 2011. 75

[61] D. Marr, S. Ullman, and T. Poggio. *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. MIT Press, 1982. 6

[62] S. Martucci. *Symmetric convolution and the discrete sine and cosine transforms: Principles and applications*. PhD thesis, Georgia Institute of Technology, 1993. 25

[63] Y. Mileva, A. Bruhn, and J. Weickert. Illumination-Robust Variational Optical Flow with Photometric Invariants. *Pattern Recognition*, pages 152–162, 2007. 67

[64] M. Mørup, M. N. Schmidt, and L. K. Hansen. Shift invariant sparse coding of image and music data. *Journal of Machine Learning*, pages 1–14, 2008. 17

[65] Y. Nesterov. A method of solving a convex programming problem with convergence rate O(1/kˆ2). *Soviet Mathematics Doklady*, 1983. 27

[66] T. Ojala. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence (PAMI)*, pages 1–35, 2002. 7

[67] B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. 6, 13, 15, 16, 31, 32

[68] B. A. Olshausen and D. J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Science*, 37(23):3311–3325, 1997. 17

[69] A. V. Oppenheim, A. S. Willsky, and with S. Hamid. *Signals and Systems (2nd Edition)*. Prentice Hall, 1996. 30

[70] N. Parikh and S. Boyd. Proximal Algorithms. 1(3):1–108, 2013. 22

[71] N. Pinto, D. D. Cox, and J. J. DiCarlo. Why is real-world visual object recognition hard? *PLoS computational biology*, 4(1):e27, jan 2008. 54, 55

[72] T. Poggio. Marr's Approach to Vision. Technical report, 1981. 6

[73] L. Rakêt, L. Roholm, M. Nielsen, and F. Lauze. TV-L1 optical flow for vector valued images. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 1–14, 2011. 74

[74] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose Machines: Articulated Pose Estimation via Inference Machines. *European Conference on Computer Vision (ECCV)*, pages 1–15, 2014. 67

[75] S. Roth and M. Black. On the spatial statistics of optical flow. *International Conference on Computer Vision (ICCV)*, pages 42–49 Vol. 1, 2005. 67

[76] O. Russakovsky and L.-j. L. Li. Best of both worlds : human-machine collaboration for object annotation. *International Conference of Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[77] J. Saragih, S. Lucey, and J. Cohn. Face alignment through subspace constrained mean-shifts. In *Computer Vision, 2009 IEEE 12th International Conference on*, number Clm, pages 1034–1041. IEEE, 2009. 74

[78] S. M. Seitz and S. Baker. Filter flow. *International Conference on Computer Vision (ICCV)*, 2009. 67

[79] P. Shivaswamy and T. Jebara. Relative margin machines. *Advances in Neural Information Processing Systems*, 21:1–8, 2008. 47

[80] E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger. Shiftable multiscale transforms. *Information Theory*, 1992. 14

[81] E. Simoncelli and B. Olshausen. Natural Image Statistics and Neural Representation. *Annual Review of Neuroscience*, 2001. 49, 50, 72

[82] K. Sohn and D. Jung. Efficient learning of sparse, distributed, convolutional feature representations for object recognition. *International Conference on Computer Vision (ICCV)*, 2011. 17

[83] G. Sullivan. The H. 264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions. In *Conference on Applications of Digital Image Processing*, pages 1–22, 2004. 31

[84] D. Sun, S. Roth, J. P. Lewis, and M. J. Black. Learning optical flow. *European Conference on Computer Vision (ECCV)*, 2008. 67

[85] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996. 16

[86] J. Tompson, R. Goroshin, A. Jain, Y. Lecun, and C. Bregler. Efficient Object Localization Using Convolutional Networks. *arXiv preprint*, 2015. 80

[87] G. Tsagkatakis, P. Tsakalides, and A. Woiselle. Compressed sensing reconstruction of convolved sparse signals. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014. 87

[88] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. *European Conference on Computer Vision (ECCV)*, pages 589–600, 2006. 49

[89] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *Computer Vision and Pattern Recognition*, (iii):3539–3546, 2010. 41

[90] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. HOGgles: Visualizing Object Detection Features. *International Conference on Computer Vision (ICCV)*, 2013. 35, 36

[91] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently Scaling up Crowdsourced Video Annotation. *International Journal of Computer Vision (IJCV)*, 2012. 1

[92] C. Wah, G. V. Horn, S. Branson, S. Maji, P. Perona, S. Belongie, and U.-s. Diego. Similarity Comparisons for Interactive Fine-Grained Categorization. *International Conference of Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[93] G. Wallace. The JPEG Still Picture Compression Standard. *Communications of the ACM*, pages 1–17, 1991. 31

[94] J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. *International Journal of Computer Vision (IJCV)*, 81:67–81, 1995. 74

[95] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large Displacement Optical Flow with Deep Matching. *International Conference on Computer Vision (ICCV)*, pages 1385–1392, dec 2013. 68, 81, 82, 87

[96] D. Wipf, B. Rao, and S. Nagarajan. Latent Variable Bayesian Models for Promoting Sparsity. *Information Theory*, 57(9):6236–6255, 2011. 19

[97] X. Xiong and F. De la Torre. Supervised Descent Method and Its Applications to Face Alignment. *International Conference of Computer Vision and Pattern Recognition (CVPR)*, pages 532–539, jun 2013. 74

[98] J. Yang, J. Wright, T. Huang, and Y. Ma. Image Super-Resolution via Sparse Representation. *IEEE Transactions on Image Processing*, 19(11):1–13, nov 2010. 35

[99] M. Yang and L. Zhang. Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary. *European Conference on Computer Vision*, pages 448–461, 2010. 54

[100] A. Yuille. Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3(1):59–70, jan 1991. 66

[101] M. Zeiler, D. Krishnan, and G. Taylor. Deconvolutional networks. *Computer Vision and Pattern Recognition (CVPR)*, 2010. 17

[102] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics (arXiv)*, 38(2):894–942, apr 2010. 19

[103] T. Zhou, Y. J. Lee, S. X. Yu, U. C. B. Icsi, and A. A. Efros. FlowWeb: Joint Image Set Alignment by Weaving Consistent, Pixel-wise Correspondences. *International Conference of Computer Vision and Pattern Recognition (CVPR)*, 2015. 90, 91

[104] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. *International Conference of Computer Vision and Pattern Recognition (CVPR)*, pages 2879–2886, jun 2012. 67

[105] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do We Need More Training Data or Better Models for Object Detection? *British Machine Vision Conference (BMVC)*, pages 80.1–80.11, 2012. 58

[106] Y. Zhu and S. Lucey. Convolutional Sparse Coded Filters Nonrigid Structure From Motion. *Pattern Analysis and Machine Intelligence (PAMI)*, 2014. 17, 32

[107] K. Zimmermann, J. Matas, and T. Svoboda. Tracking by an optimal sequence of linear predictors. *Pattern Analysis and Machine Intelligence (PAMI)*, 31(4):677–92, apr 2009. 74

Nomenclature

Operators

conj	Conjugate operator.
min	Minimum function, typically of an objective.
rank	Matrix rank operator.
abs	Absolute value function.
diag	Diagonalization of a vector.
prox	Proximal operator to a function.
sgn	Sign function, removing magnitude.
subject to	An expression used before specification of constraints.
vec	Vectorization operator, flattening a matrix.

Symbols

$*$	Discrete Convolution operator.
$\star$	Discrete Correlation operator.
$\mathcal{F}$	Discrete fourier transform.
$\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$	Partial derivative of warp function wrt parameters.
$\mathbf{I}$	Identity matrix.
f	A model fitting function.
$\mathbf{g}, \mathcal{R}$	A regularization term.
∇	Discrete gradient operator.
$\mathcal{K}$	A Kernel function.
$\otimes$	Kronecker product.
$\mathbb{R}$	Set of real numbers.
$p \in \mathcal{P}$	Set membership.

Definitions

NRSfM	Non-rigid Structure from Motion.
SfM	Structure from Motion.

Basis	A matrix or Euclidean space from which a signal can be reconstructed.
Capacity	The degrees of freedom in the decision boundary of a discriminative classifier.
Convolutional Network	A specialized neural network that maintains dense connections across local spatial neighborhoods of parameters.
Correspondence	Located structural similarites between two signals, esp. where the underlying sources are geometrically or anatomically similar.
Decorrelation	Removing correlations between values in a signal whilst maintaining structure.
Discriminative Model	A model which generates hypotheses from a conditionally dependent distribution.
Fourier Domain	An affine transform of the spatial or temporal domain in which frequencies are explicitly represented.
Generative Model	A Model which generates hypotheses from a joint distribution.
Invariance	In the context of computer vision, a function that remains approximately constant when subjected to a particular type of input variation.
Locality	Interactions in a signal that affect only spatially or temporally adjacent neighbors.
Normalization	Shifting values into a common scale frame.
Pooling	Summarizing values over a local spatial region, by taking the min, max, sum or average.
Rectification	Making a signal non-negative, either through an absolute value operator or truncation.
Registration	Directly solving for a parametrization of a collection of signals that maximizes their similarity.
Representation	The space in which the appearance of an image/object is embedded.
Separable Filter	A 2D filter kernel that can be expressed as the outer product of two vectors.
Sparse	A signal that contains very few non-zero elements.
Stationarity	A distribution whose statistics are invariance to temporal or spatial shifts.
SVM	Support Vector Machine.
V1	The primary visual cortex in the mammalian brain.

Author Biography

Hilton Bristow is affiliated with the Speech, Audio, Image and Video Technology (SAIVT) lab within the Science and Engineering Faculty at the Queensland University of Technology in Australia. Prior to his doctoral studies, he was awarded a Bachelor of Engineering (Mechatronics, Hons I) and University Medal from the University of Queensland. He has a long-standing interest in robotics, computer vision, machine learning and programming, and has multiple publications in those fields.

In 2012, Hilton interned at Willow Garage in San Francisco, and in 2013 participated in Google's Summer of Code, sponsored by the Open Computer Vision (OpenCV) Foundation. He was granted a visiting scholar's position at Carnegie Mellon University in 2014, where he spent time developing the ideas that would form the basis of this dissertation. He defended his thesis in July 2015.