Towards large-scale nonparametric scene parsing of images and video

by

Frederick Tung

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

March 2017

Abstract

In computer vision, scene parsing is the problem of labelling every pixel in an image or video with its semantic category. Its goal is a complete and consistent semantic interpretation of the structure of the real world scene. Scene parsing forms a core component in many emerging technologies such as self-driving vehicles and prosthetic vision, and also informs complementary computer vision tasks such as depth estimation.

This thesis presents a novel nonparametric scene parsing framework for images and video. In contrast to conventional practice, our scene parsing framework is built on nonparametric search-based label transfer instead of discriminative classification. We formulate exemplar-based scene parsing for both 2D (from images) and 3D (from video), and demonstrate accurate labelling on standard benchmarks. Since our framework is nonparametric, it is easily extensible to new categories and examples as the database grows.

Nonparametric scene parsing is computationally demanding at test time, and requires methods for searching large collections of data that are time and memory efficient. This thesis also presents two novel binary encoding algorithms for large-scale approximate nearest neighbor search: the bank of random rotations is data independent and does not require training, while the supervised sparse projections algorithm targets efficient search of high-dimensional labelled data. We evaluate these algorithms on standard retrieval benchmarks, and then demonstrate their integration into our nonparametric scene parsing framework. Using 256-bit codes, binary encoding reduces search times by an order of magnitude and memory requirements by three orders of magnitude, while maintaining a mean per-class accuracy within 1% on the 3D scene parsing task.

Preface

This thesis presents original and independent research conducted under the guidance of my supervisor, Dr. James J. Little. Parts of this thesis have been published as follows.

Chapter 2. An early version of the nonparametric 2D scene parsing algorithm CollageParsing was published in *F. Tung and J. J. Little, CollageParsing: Nonparametric scene parsing by adaptive overlapping windows, European Conference on Computer Vision (ECCV), 2014* [130]. The 2D scene parsing work was later extended to the journal article *F. Tung and J. J. Little, Scene parsing by nonparametric label transfer of content-adaptive windows, Computer Vision and Image Understanding (CVIU), 143:191-200, 2016* [132]. I conceived the initial idea, developed the algorithm, and designed and executed the experiments.

Chapter 3. A version of the nonparametric 3D scene parsing algorithm MF3D is presently in peer review. I conceived the idea of this extension, developed the algorithm, and designed and executed the experiments.

Chapter 4. The bank of random rotations algorithm is my individual contribution to a collaborative publication *F. Tung, J. Martinez, H. H. Hoos, and J. J. Little, Bank of quantization models: A data-specific approach to learning binary codes for large-scale retrieval applications, IEEE Winter Conference on Applications of Computer Vision (WACV), 2015* [133]. I conceived the initial idea, developed the algorithm, and designed and executed the experiments described in this thesis. J. Martinez contributed a training-based extension to the algorithm and additional experiments, which are not included in this thesis. Dr. Holger H. Hoos and Dr. James J. Little contributed to the writing. The supervised sparse projections algorithm will be published in *F. Tung and J. J. Little, SSP: Supervised Sparse Projections* *for large-scale retrieval in high dimensions, Asian Conference on Computer Vision (ACCV), 2016* [131]. I conceived the idea, developed the algorithm, and designed and executed the experiments.

Dr. James J. Little contributed to the planning and writing of all of the above publications, and provided valuable overall guidance.

Table of Contents

List of Tables

List of Figures

Acknowledgments

This research was funded by a postgraduate scholarship from the Natural Sciences and Engineering Research Council of Canada and by a University of British Columbia Four Year Doctoral Fellowship.

Chapter 1

Introduction

Human vision is fast and versatile. We rapidly recognize familiar faces and objects, and process complex scenes around us at a glance. Giving machines the same capacity to interpret still images and moving sequences is an ambitious endeavour, yet one that has seen much progress in recent years. Computer vision as a field encompasses a broad range of tasks, from low-level image acquisition to high-level semantic recognition. Low-level computer vision is concerned with basic image tasks such as image filtering [48, 148], edge or contour detection [2, 25], and 3D reconstruction [89, 121]. High-level vision algorithms analyze images and attempt to draw meaningful inferences about the real-world scenes. The goal of high-level vision is to bridge the semantic gap [120]: the disconnect between the raw pixel data and high-level semantics.

High-level semantic inferences about the scene imaged by a camera can be made at different levels of abstraction. A traditional level of scene inference is scene categorization, or naming [63, 119]. Scene categorization is concerned with determining the general semantic category of a scene (e.g. forest, concert hall), which can be useful for contextual object detection or image retrieval. Instead of a basic category name, some applications such as guided image search [68] perform inference at the level of nameable attributes (e.g. cluttered, rugged) [29, 79, 101]. Scene parsing performs high-level semantic inference at the pixel level [82, 127]: the objective of scene parsing is to assign a semantic category label to every pixel in an image or video.

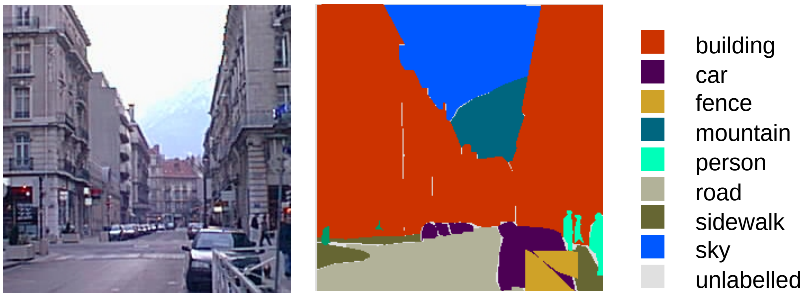


Figure 1.1: Sample output of our nonparametric scene parsing algorithm for images (Section 2) on the SIFT Flow benchmark [82].

1.1 Scene parsing: task and motivation

Scene parsing is a fundamental problem in computer vision. It is concerned with constructing a complete and consistent semantic interpretation of the structure of the real-world scene. In other words, we wish to explain the entire real-world scene in terms of its semantic components. In contrast to object categorization [11, 99, 102], scene parsing does not assume the image contains a single dominant object, as in product images [9] for example. Similar to object detection [31, 41, 107], scene parsing is concerned with identifying spatially well-defined, foreground objects of interest such as vehicles, but unlike object detection, scene parsing is also concerned with recognizing amorphous background categories such as grass or pavement. While the output of an object detection algorithm is typically a set of bounding boxes around detected objects, the output of scene parsing is a dense pixelwise labelling of the scene. Fig. 1.1 shows the 2D scene parsing output for a sample image.

Scene parsing enables many practical applications. It is a core component in emerging technologies such as prosthetic vision [54] and self-driving vehicles and mobile robots [113, 135, 138]. Scene parsing is also used to inform complementary computer vision tasks such as depth estimation [81], 3D scene reconstruction [46, 76], and urban modelling [22, 88].

Many factors make scene parsing a challenging task. Some challenges are related to imaging: important components of a scene may be occluded, imaged under non-uniform lighting conditions, or appear at different orientations and scales.

However, another important factor is scalability: scene parsing is intrinsically challenging due to the breadth of possible foreground and background categories. The operating range of categories depends on the application. For example, standard scene parsing benchmarks for self-driving in urban environments focus on a small set of relevant categories such as road, building, tree, sidewalk, fence, car, and pedestrian [15, 113, 135]; in practice, the complexity of the self-driving task may require finer-grained categories such as police officers directing traffic. For image or video search, a broad vocabulary of dozens or hundreds of categories may be required. In addition, each of these categories can exhibit significant intra-category variation in visual appearance.

1.2 The role of big data

In the past several years, “big data” has transformed how we approach high-level computer vision problems. The emergence and ongoing development of today’s leading architecture for many recognition tasks, deep neural networks [72, 107], has been enabled by the creation of massive labelled databases such as ImageNet [111]. With unprecedented amounts of labelled training examples available for tasks ranging from image classification [111] to 3D reconstruction [122], increasingly complex models have been trained to achieve exciting new recognition results [51].

Most of the contemporary research in solving high-level vision problems has adopted a strategy of improving and applying more sophisticated classification models. For example, most methods for scene parsing learn domain-specific classification models to discriminate between different categories of pixels or voxels [28, 77, 113, 135, 138]. A model may be trained to distinguish a tree voxel from a voxel of a car, road, building, or any other category in a pre-defined set of categories. Such methods are said to be *parametric*: they involve constructing compact models of the categories of interest, in which the parameters of the models are learned from training examples. Inference on new inputs is performed by evaluating them against the learned models. While there has been much work in leveraging big data for training powerful parametric models, another complementary way to leverage big data for computer vision has been largely underexplored. *Nonpara-*

metric methods are data-driven instead of model-driven: inference on new inputs is performed by directly comparing them to similar examples in the data. Leveraging big data for search allows for the development of powerful exemplar-based algorithms for visual recognition.

Why might a nonparametric, search-based approach to scene parsing be useful? Nonparametric approaches are database driven and do not require expensive training of category models. They can be easily extended to new semantic categories as the database grows: a new semantic category can be added to the database without having to learn how to distinguish it from all previously learned categories. Consequently, nonparametric scene parsing is scalable to an online growth in the database. This scalability can be useful in embedded applications, such as the addition of new mapped environments to a self-driving vehicle, as well as in web-scale applications, such as the addition of new user contributions to an open-universe database (e.g. LabelMe [112]). An open-universe database is one in which users can continually add new images, annotations, or other content.

From a theory perspective, there are several reasons why nonparametric approaches are worth studying. Stone [123] proved that, in finite dimensions, the k-nearest neighbour classifier is universally consistent: as the number of data points increases, its risk converges to the Bayes risk, for all probability distributions on the joint input-output space. Hence, nonparametric approaches are sufficient to solve any learning problem given enough data. Fixed-size deep neural networks do not share a similar theoretical guarantee, though they can achieve good practical performance with less data by exploiting the problem structure. In addition, nonparametric approaches are easy to interpret: it is straightforward to trace the effects of changes in the input domain to the output domain. It is also easy to explain *why* nonparametric approaches make the predictions they do. On the other hand, the behaviour of deep neural networks is still not fully understood. For example, Szegedy et al. [125] showed that a small, imperceptible perturbation in the input to a deep neural network can induce the network to output an arbitrary incorrect classification label. Such “adversarial examples” can be constructed for practically any input, suggesting that deep neural networks have “intrinsic blind spots, whose structure is connected to the data distribution in a non-obvious way” [125].

Nonparametric search and deep learning classification are by no means mutually exclusive; rather, the two approaches are complementary and can often be combined to achieve performance superior to either approach by itself. The novel scene parsing algorithms presented in this thesis leverage the strengths of deep learning features within a nonparametric search-based framework. In the other direction, researchers have leveraged advances in large-scale search to accelerate deep neural network frameworks [142].

1.3 Nonparametric scene parsing

Scalable nonparametric approaches to classic computer vision problems have been demonstrated in web-scale studies by Torralba et al. [129] and Hays and Efros [47]. Torralba et al. constructed a database of 80 million tiny (32x32) images from the web following the WordNet lexical dictionary [30], and demonstrated that the massive scale of the database enables effective object and scene classification using simple nonparametric matching techniques, despite the low resolution of the images. Scene completion, or inpainting, refers to the task of synthesizing missing parts of an image such that the output image is semantically consistent and visually plausible. Hays and Efros [47] proposed a nonparametric scene completion algorithm that draws on a database of two million unlabelled scenes to generate potential image completions. Later, Ordonez et al. [98] collected one million human-captioned photos from the web to develop a nonparametric approach for automatic image captioning. These studies demonstrate that high-level vision tasks can be approached by combining principled nonparametric search with large, semantically-annotated databases of images.

Although scene parsing is a well studied problem in computer vision, it is only in recent years that nonparametric solutions have been proposed. Liu et al. [82] were the first to adopt a search-based approach to scene parsing. Given a query image, Liu et al.'s 'label transfer' algorithm finds the image's nearest neighbors in a database of scenes annotated with semantic labels, warps the neighbors to the query image using SIFT Flow [83], and transfers the annotations from the neighbors to the query image using a Markov random field defined over pixels. A Markov random field (MRF) is a probabilistic graphical model that is often used to solve discrete labelling problems. More recent nonparametric scene parsing algorithms per-

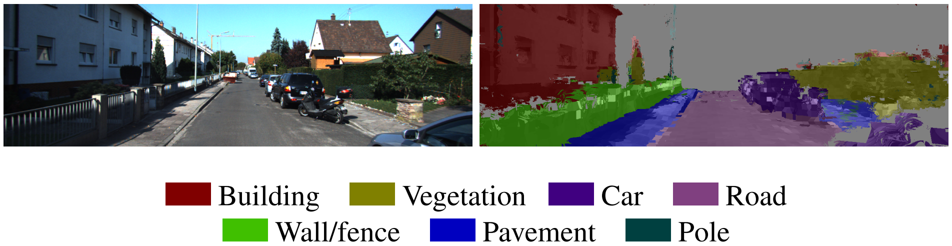


Figure 1.2: Sample output of our nonparametric 3D scene parsing algorithm for video (Section 3) to a depth of 25 metres on the KITTI labelled benchmark [38].

form label transfer at the level of superpixels [26, 91, 118, 127]. Superpixels offer a higher level of perceptual abstraction than pixels, enabling more efficient inference in a random field model. Large, cohesive groups of pixels can be labelled at once. However, while superpixel-based methods tend to perform well on large regions of background categories (sometimes referred to as 'stuff' in the literature [52]), they tend to perform less effectively on foreground object categories ('things'). Though better than pixels, superpixels are still a relatively low-level unit for label transfer, and they tend to fragment objects.

1.4 Contributions

In this thesis, we first demonstrate that nonparametric scene parsing of 2D images can be improved by performing label transfer at a higher level of perceptual organization than superpixels. We propose an algorithm called CollageParsing that performs label transfer at the level of object proposals [1, 151], which unlike superpixels, are designed to preserve entire objects instead of fragmenting them. Performing label transfer using object proposals enables the construction of a more effective Markov random field unary potential than previous approaches. The name 'CollageParsing' refers to the collage-like unary potential computation, which accumulates evidence from overlapping proposal windows, and derives from the name of Tighe and Lazebnik's landmark 'SuperParsing' algorithm [127], which performs superpixel-based scene parsing.

Furthermore, we propose the first nonparametric method for 3D scene parsing of video. In 3D scene parsing of video, we are interested in reconstructing a 3D model of the scene from video data, and densely labelling the reconstructed scene with a set of semantic categories. A densely labelled 3D reconstruction has applications in autonomous navigation [113, 135, 138] as well as in urban modelling [22, 88, 108]. Many applications require the 3D reconstruction and labelling to be performed online: that is, using only sensory data that has been observed so far by the platform, as opposed to offline labelling, in which the entire video sequence may be loaded for post-analysis. We describe a novel method for online 3D scene parsing of video that is nonparametric and easy to scale. Fig. 1.2 visualizes a sample result.

In general, nonparametric algorithms in computer vision are expensive in terms of both memory and test time requirements. Nonparametric approaches are memory intensive due to the need to store the large labelled database in memory, and time intensive since this large database must be searched at test time. To enable accurate nonparametric scene parsing, millions of local feature descriptors may need to be efficiently stored and searched. Hence, an important question demands our attention: how can we search "big data" efficiently while maintaining economy of storage?

One highly successful approach to efficient large-scale retrieval is compact binary encoding (or hashing) [43, 74, 95, 114, 143]. The idea is to encode each database feature vector into a short binary signature (or binary code) using a generated projection matrix. Given a novel feature vector as a query, the query is similarly projected into a binary code and compared to the binary codes in the database. Binary encoding is time-efficient because binary code distances can be computed quickly using bit operations. In particular, the Hamming distance between two binary codes can be computed by taking an XOR and counting the ones, which is an operation supported directly in modern hardware. Binary encoding is also memory-efficient because the generated binary codes are typically at least an order of magnitude more compact to store than the original features. We propose two new methods for large-scale search based on binary encoding. The bank of random rotations method is data-independent and does not require re-training as the database grows over time. The supervised sparse projections method specifically targets search in high dimensions and enables faster encoding at test time.

We demonstrate the effectiveness of these methods on standalone retrieval datasets, and also study their integration into our scene parsing framework.

In summary, this thesis makes the following contributions:

- A novel nonparametric algorithm for 2D scene parsing of images. In contrast to traditional nonparametric scene parsing methods that perform matching on the level of pixels or superpixels, we obtain higher labelling accuracy by integrating information from overlapping object proposals.

- A novel nonparametric algorithm for online 3D scene parsing of video. Our method constructs a semantically labelled 3D reconstruction of the environment in an online manner, building on the principles underlying our 2D scene parsing work.

- A demonstration of how deep learning features can be used in a nonparametric label transfer setting for the first time, and an experiment providing a positive test case that it is the learned features rather than other aspects of deep learning that are most important for scene parsing performance.

- Two novel binary encoding algorithms for large-scale nonparametric search. The first algorithm is data-independent and easily scalable to new examples and categories. The second algorithm is supervised and targets high-dimensional feature vectors such as the proposal descriptors in our scene parsing framework.

Chapter 2

Nonparametric 2D Scene Parsing of Images

In this chapter, we are interested in the task of assigning a semantic class label to every pixel in a single RGB image (Fig. 1.1). We would like to explain the entire image, including both foreground objects of interest and background elements. Foreground objects are typically compact objects with well-defined boundaries, sometimes referred to as ‘things’ in the literature [52]; examples include cars and people. Background elements are often amorphous in shape or defined by texture, sometimes referred to as ‘stuff’; examples include grass and mountain.

With few exceptions, the dominant paradigm today for approaching recognition problems in computer vision is that of parametric, discriminative classification (see Section 1.2 for a discussion). For example, we can learn discriminative classifiers to recognize airplanes or motorbikes in images, or to distinguish a tree pixel from a pixel of a car, road, or building. On the other hand, a nonparametric, search-based approach to scene parsing attempts to label the query image by matching parts of the image to similar parts in a large database of labelled images. Typically, the problem is formulated within a Markov random field framework in which unary potentials are computed by nearest-neighbor search. Intuitively, labels are ‘transferred’ from exemplars in the database to the query image.

State-of-the-art nonparametric scene parsing algorithms perform label transfer at the level of pixels or superpixels [26, 82, 91, 118, 127]. Superpixels offer a higher level of perceptual abstraction than pixels, enabling more efficient inference in a random field model. Large, cohesive groups of pixels can be labelled at once. However, while superpixel-based methods tend to perform well on large regions of background ('stuff') categories, they tend to perform less effectively on foreground object ('thing') categories. We argue that, though better than pixels, superpixels are still a relatively low-level unit for nonparametric label transfer. Superpixels tend to fragment objects. Is it possible to perform label transfer at a higher level of organization?

This chapter presents a novel nonparametric algorithm, called CollageParsing, for 2D scene parsing of images. In contrast to conventional nonparametric scene parsing methods that perform matching on the level of pixels or superpixels, we obtain higher labelling accuracy by integrating information from overlapping object proposals [1, 151]. Object proposals provide a higher level of perceptual organization than superpixels, and unlike superpixels are designed to preserve entire objects instead of fragmenting them. Performing label transfer using object proposals enables the construction of a more effective Markov random field unary potential than previous approaches. On a standard benchmark [82] consisting of outdoor scenes from the LabelMe database [112], CollageParsing obtains state-of-the-art performance with 15 to 19% higher average per-class accuracy than recent nonparametric scene parsing algorithms.

2.1 Related work

2.1.1 Markov random fields in scene parsing

Markov random fields (MRFs) have been applied in a broad range of computer vision tasks, and have seen widespread adoption among methods for scene parsing. An MRF is a probabilistic graphical model that is often used to solve discrete labelling problems. We can think of an MRF as defining an energy (or cost) over a graph of random variables reflecting how good or bad a particular labelling is. Each random variable can be assigned a particular discrete label. The MRF energy is composed of potential functions defined over cliques of conditionally dependent random variables, typically singletons and pairs (unary and pairwise potentials, respectively). A conditional random field (CRF) is an MRF in which the potentials are conditioned on observation variables.

Scene parsing lends itself naturally to formulation in an MRF framework, with random variables corresponding to image pixels or superpixels, and assignments to those random variables producing an image labelling. Shotton et al.'s TextonBoost framework [115] consists of a CRF with shape-texture, color, and location unary potentials and contrast-sensitive pairwise potentials. The shape-texture unary potential uses a boosted classifier of rectangular filter responses on texton maps. Rabinovich et al. [103] incorporated object class co-occurrences in the pairwise potential term of a CRF model. Kohli et al. [66] introduced a higher-order potential that encourages pixels within a segment to share the same label. The authors showed that approximate inference in a CRF augmented with their higher-order potential can be performed efficiently using graph cuts. Ladický et al. [77] proposed a hierarchical CRF model in which potentials at multiple spatial scales are combined; scales in the hierarchy correspond to pixels and superpixels at varying granularities. Later, Ladický et al. [78] also introduced a higher-order potential over label co-occurrences that can be efficiently approximately minimized using graph cuts. The potential is designed to be invariant to the number of pixels assigned to a label and to encourage parsimony. Boix et al. [12] proposed a two-level hierarchical CRF model in which the global node in the hierarchy can be assigned a set of labels instead of a single label. "Harmony" potentials between local nodes and the global node encourage local nodes to take on labels in the global node's set of labels. A thorough treatment of MRFs in computer vision can be found in the survey of Wang et al. [139].

Our proposed CollageParsing algorithm is formulated in an MRF framework, and introduces a novel and effective computation of unary potentials by nonparametric label transfer of overlapping object proposals.

2.1.2 Nonparametric scene parsing

Liu et al. [82] developed the first nonparametric, search-based approach to scene parsing. Given a query image, Liu et al.'s 'label transfer' algorithm finds the image's nearest neighbors in a database of scenes annotated with semantic labels, warps the neighbors to the query image using SIFT Flow [83], and transfers the annotations from the neighbors to the query image using an MRF defined over pixels. Tighe and Lazebnik's landmark SuperParsing algorithm [127] builds an MRF over superpixels instead of pixels. Label transfer is performed by matching superpixels in the query image with similar superpixels in the query's nearest neighbor images. The nonparametric label transfer can be combined with additional parametric geometry classification (sky, vertical, horizontal) to improve labelling consistency.

Since the introduction of SuperParsing, other effective superpixel-based nonparametric scene parsing algorithms have been developed. Eigen and Fergus [26] learned weights for each descriptor in the database in a supervised manner to reduce the influence of distractor superpixels, and augmented the retrieved set of neighbor superpixels with examples from rare classes with similar local context. Myeong et al. [91] applied link prediction techniques to superpixels extracted from the query image and its nearest neighbors to learn more effective pairwise potentials. Singh and Košecká [118] applied a locally adaptive nearest neighbor technique to retrieve similar superpixels, and refined the retrieval set of query image neighbors by comparing spatial pyramids of predicted labels.

Instead of computing the set of nearest neighbor images at test time, the PatchMatchGraph method of Gould and Zhang [44] builds offline a graph of patch correspondences across all database images. Patch correspondences are found using an extended version of the PatchMatch algorithm [8] with additional "move" types for directing the local search for correspondences.

CollageParsing belongs to this general family of nonparametric scene parsing approaches, and we compare to this prior body of work in the experiments.

2.1.3 Parametric scene parsing

Besides the traditional MRF approaches in Section 2.1.1, many recent successful methods employ parametric strategies. Farabet et al. [28] developed a scene parsing algorithm combining several deep learning techniques. Dense multi-scale features are computed at each pixel and input to a trained neural network to obtain feature maps. Feature maps are aggregated over regions in a hierarchical segmentation tree. Regions are classified using a second neural network and pixels are finally labelled by the ancestor region with the highest purity score. Tighe and Lazebnik [128] extended the SuperParsing algorithm [127] with per-exemplar detectors (Exemplar-SVMs [86]). A superpixel-based data term is the same as in SuperParsing. A detector-based data term is obtained by running the per-exemplar detectors of class instances found in the retrieval set and accumulating a weighted sum of the detection masks. The final unary potential over pixels is determined by another SVM trained on the two data terms. Yang et al. [146] also proposed a hybrid parametric superpixel-based algorithm. The retrieved set of neighbor superpixels is augmented with a fixed (query-independent) set of examples of rare classes. Rare class examples are extracted offline in a training phase; a class is considered rare if it falls under the 20% 'tail' in the overall distribution of classes in the database. Superpixel unary potentials are determined by a combination of non-parametric matching and a parametric SVM classifier trained on the rare class examples. Predicted labels are used to refine the retrieval sets of query image neighbors and superpixel neighbors in a second pass through the labelling pipeline.

While the characteristics and assumptions of parametric approaches are different from ours (see discussion in Section 1.2), we include parametric baselines in the experimental comparisons for completeness.

2.1.4 Segmentation methods with overlapping regions

Pantofaru et al. [100] combined multiple image segmentations to create a map of intersections-of-regions. Each intersection-of-regions is classified by averaging over the category predictions of a learned region-based classifier on its constituent regions. Li et al. [80] also formed intersections-of-regions (referred to as superpixels in their formulation) using multiple image segmentations. Intersections-of-regions are aggregated into objects by maximizing a composite likelihood of predicted object overlap with segments. In contrast, CollageParsing is formulated in an MRF framework and operates on high-level object proposals instead of intersections of regions generated by low-level image segmentation. CollageParsing is also a nonparametric method, while Pantofaru et al. [100] and Li et al. [80] are parametric methods.

Kuettel and Ferrari [73] proposed a figure-ground (foreground-background) segmentation algorithm that transfers foreground masks from the training images by matching object proposals. Matching is independent of the global similarity between the test and training images. The transferred foreground masks are used to derive unary potentials in a CRF. CollageParsing performs multi-class scene parsing instead of figure-ground separation, and specifically matches windows from globally similar images, which form the retrieval set. The retrieval set adds context that is important for multi-class labelling and enables scalable label transfer in large databases. The label transfer in CollageParsing is nonparametric, while Kuettel and Ferrari [73] iteratively learns a mixture model of appearance to refine the figure-ground segmentation.

2.1.5 Scene collage

Isola and Liu [58] proposed a similarly named 'scene collage' model and explored applications in image editing, random scene synthesis, and image-to-anaglyph. In contrast to CollageParsing, the scene collage is an image representation: it represents an image by layers of warped segments from a dictionary.

2.1.6 Object proposals

Detecting objects in images can be computationally expensive. A traditional sliding window approach moves a window across an image and evaluates a trained object classifier within each window [23, 31]. The process is repeated at multiple scales and aspect ratios. Unfortunately, the sliding window approach spends many computation cycles evaluating windows that have little chance of containing objects – patches of background texture, for example. As a result, sliding window detectors can be slow, limiting their feasibility for real-time applications or restricting the complexity of the underlying trained classifier.

Object proposal algorithms have been developed to address this inefficiency [1, 3, 16, 20, 27, 70, 87, 136, 151]. Object proposal algorithms attempt to extract a small number of candidate windows (on the order of thousands) to be evaluated by the trained object classifier. Object proposal algorithms are designed such that the proposal windows have high probability of containing objects of any category. Object proposals are a key component in leading large-scale object detection algorithms such as R-CNN [41] and its extensions [40, 107].

Alexe et al. [1] first defined the 'objectness' of an image window as the likelihood that the window contains a foreground object of any kind instead of background texture such as grass, sky, or road. Objects often have a closed boundary, a contrasting appearance from surroundings, and/or are unique (salient) in the image. Alexe et al. proposed an objectness measure integrating multiscale saliency, color contrast, and superpixels straddling cues. The multiscale saliency cue is a multiscale adaptation of Hou and Zhang's visual saliency algorithm [56]. The color contrast cue measures the difference between the color histograms of the window and its surrounding rectangular ring. The superpixels straddling cue measures the extent to which superpixels straddle the window (contain pixels both inside and outside the window). Windows that tightly bound an object are likely to have low straddling.

Since the introduction of 'objectness' many effective methods have been proposed for generating object proposals. Van de Sande et al. [136] proposed selective search, in which object proposals are generated using a hierarchical segmentation process that greedily merges similar segments until a single segment remains. All generated segments, or their bounding boxes, become candidate windows. Multiple segmentations are performed in color spaces with different sensitivities to shadows, shading, and highlights. Arbelaez et al.'s multiscale combinatorial grouping [3] also uses hierarchical segmentation. Multiscale combinatorial grouping generates proposals by considering singletons, pairs, triplets, and 4-tuples of regions generated by a multiscale hierarchical segmentation algorithm. Proposals are further refined using a random forest regressor trained on size, location, shape, and contour strength features. Manen et al. [87] generated object proposals by growing random partial spanning trees on a graph defined over superpixels. Edge weights encode the probability of two adjacent superpixels belonging to the same object. The trees are grown using a randomized version of Prim's algorithm that replaces greedy edge selection with weighted sampling based on edge weights, and adds a randomized stopping condition.

Another interesting approach to generating object proposals involves solving multiple figure-ground (foreground-background) segmentation problems from different initial foreground seeds. Carreira and Sminchisescu [16] performed multiple foreground-background segmentations from initial seeds on a regular grid using color features. A random forest regressor trained on mid-level appearance features is used rank the proposals. Endres and Hoiem [27] selected initial seeds from a hierarchical segmentation and estimated the affinities between superpixels using multiple learned classifiers. The classifiers predict whether the region and seed lie on the same object; whether a region lies on the left, right, top, or bottom of an object; and whether a region is homogeneously foreground or background. A structural learning method is used to rank the proposals based on overlap and appearance features. Krähenbühl and Koltun [70] computed geodesic distances between foreground seeds and background superpixels. Seeds can be chosen heuristically or using a learned classifier. Geodesic distances refer to shortest paths over a weighted graph of superpixels, in which edges are weighted by the probability of boundary between superpixels. Proposals are derived from critical level sets in the geodesic distance map.

Superpixel-based proposal methods are often sensitive to small changes in the image content [55]. Some approaches forego superpixel segmentation entirely. Cheng et al. [20] developed a fast method for generating object proposals based on image gradients. Given an image window, gradient norms are aggregated in an 8x8 grid and the resulting 64-dimensional feature is scored using a learned linear SVM. The computation can be accelerated by binarizing the linear model and normed gradient features, allowing scores to be evaluated via fast bit operations. Zitnick and Dollár [151] introduced the Edge Boxes algorithm, which generates object proposals using edge cues: the objectness score of a window is determined by the number and strength of edges wholly contained within the window. A fast structured forest method [25] is used for edge detection.

Our CollageParsing algorithm makes use of object proposals, and a key novelty is the computation of unary potentials by nonparametric label transfer on overlapping object proposals. As described in Section 2.2.3, we compute object proposals using the Edge Boxes algorithm [151].

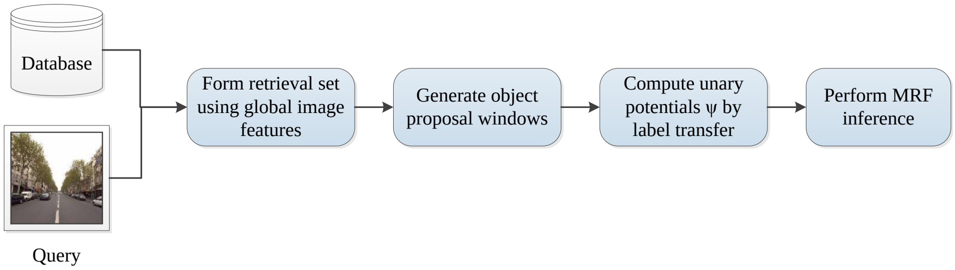

Database
Query
Form retrieval set using global image features
Generate object proposal windows
Compute unary potentials ψ by label transfer
Perform MRF inference

Figure 2.1: High-level overview of the CollageParsing scene parsing pipeline. CollageParsing constructs a Markov random field over image pixels and performs nonparametric label transfer at the level of object proposals. Object proposals are designed to preserve objects instead of fragmenting them. Each of the four components of the pipeline (blue boxes) is described in Sections 2.2.2 to 2.2.5.

2.2 Algorithm description: CollageParsing

CollageParsing follows a common nonparametric scene parsing pipeline, in which an MRF is constructed and nonparametric label transfer is used to determine unary potentials. The CollageParsing pipeline is illustrated in Fig. 2.1. First, a short list of the query image's nearest neighbors in the database is formed using global image features (Section 2.2.2). The short list is referred to as the retrieval set. Next, object proposal windows are extracted from the query image (Section 2.2.3). Unary potentials are then computed by matching these windows to similar object proposal windows in the retrieval set (Section 2.2.4). The unary potentials are combined with pairwise potentials, and approximate MRF inference is performed to obtain a pixelwise labelling of the image (Section 2.2.5). We start by defining the MRF model (Section 2.2.1) and then proceed to explain each stage of the pipeline in turn.

2.2.1 Markov random field model

We adopt a Markov random field $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ defined over a dense pixel lattice, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of undirected edges. Let X_i be the random variable associated with node $i \in \mathcal{V}$. X_i can take on a label from the label set $\mathcal{L} = \{l_1, l_2, ..., l_k\}$, corresponding to the vocabulary of semantic category labels. Denote by x_i a realization or assignment of random variable X_i, and by $\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}$ the assignments of all random variables; that is, $\mathbf{x}$ represents a complete labelling of the image. A clique c is a fully connected set of nodes in $\mathcal{G}$. By the Hammersley-Clifford theorem, the probability distribution $p(\mathbf{x})$ over labellings is a Gibbs distribution and can be expressed as

$$p(\mathbf{x}) = \frac{1}{Z} \exp\Big(- \sum_c \psi_c(\mathbf{x}_c) \Big) \tag{2.1}$$

where $\psi_c(\mathbf{x}_c)$ are potential functions defined over cliques and Z is the normalizing factor or partition function. Let $E(\mathbf{x})$ denote the energy of the MRF, defined as

$$E(\mathbf{x}) = \sum_c \psi_c(\mathbf{x}_c) \tag{2.2}$$

Then the maximum a posteriori (MAP) labelling is given by

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}} p(\mathbf{x}) = \arg\min_{\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}} E(\mathbf{x}) \tag{2.3}$$

MRFs in computer vision are often defined over unary and pairwise cliques because of the availability of efficient approximate inference techniques. Following this modelling practice, we arrive at an MRF energy composed of unary and pairwise potentials:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \tag{2.4}$$

The main novelty of CollageParsing is in the computation of the unary potentials by nonparametric label transfer of overlapping object proposals, which will be described in detail in Section 2.2.4.

2.2.2 Forming the retrieval set

The standard first step in nonparametric scene parsing is the retrieval of database images that are contextually similar to the query image. In addition to filtering out semantically irrelevant database images that are likely to be unhelpful, a small retrieval set makes nearest neighbor based label transfer practical on large datasets.

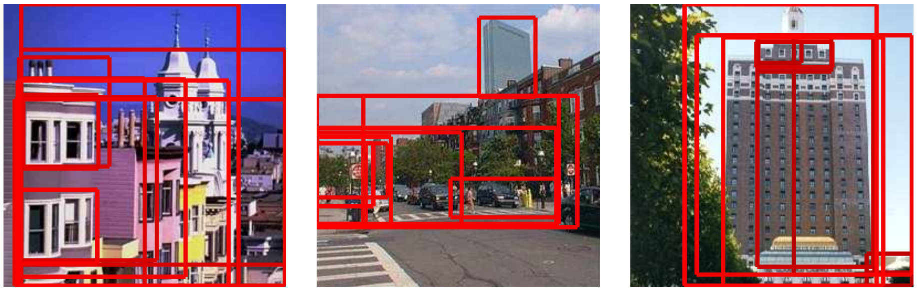


Figure 2.2: Examples of image windows with high ‘objectness’, as computed using Zitnick and Dollár’s Edge Boxes method [151]. In each image, windows with the top ten objectness scores are shown. Images are from the SIFT Flow dataset [82].

This coarse filtering step is typically performed using fast global image descriptors. CollageParsing compares the query image to the database images using Gist [97] and HOG visual words [82, 145]. The database images are sorted by similarity to the query image with respect to these two features, and the K best average ranks are selected as the retrieval set.

An assumption made in this step is that the retrieval set is sufficient for accurate parsing of the query image. However, the coarse filtering could potentially discard relevant exemplars, leading to poor matching during label transfer. Rare categories are at particular risk of being under-represented in the pool of exemplars. We study the effect of retrieval set size on accuracy later in the experiments (Section 2.3).

2.2.3 Computing object proposal windows

CollageParsing performs label transfer at the level of object proposal windows. Since the original ‘objectness’ work of Alexe et al. [1], many algorithms have been developed for generating object proposals (Section 2.1.6). In this chapter, we adopt a state-of-the-art method called Edge Boxes.

Introduced by Zitnick and Dollár [151], the Edge Boxes algorithm generates object window proposals using edge cues alone. The objectness score of a window is determined by the number and strength of edges wholly contained within the window. A fast structured forest method [25] is used for edge detection. Candidate windows for score evaluation are generated by a sliding-window search over position, scale, and aspect ratio, with the step size parameterized by the target intersection over union (IoU). High-scoring windows are further refined by greedy iterative search. Finally, a non-maximal suppression step parameterized by the target IoU is performed. We use the publicly available implementation of Edge Boxes with the default parameter values recommended in Zitnick and Dollár [151]. Fig. 2.2 shows examples of proposals computed using the Edge Boxes algorithm.

2.2.4 Computing unary potentials

CollageParsing computes unary potentials by performing nearest-neighbor matching on object proposal windows and transferring the category labels. Compared with previous nonparametric scene parsing methods, label transfer is performed at a higher level of perceptual organization designed to preserve the integrity of objects.

This matching requires a way to compare object proposal windows in the query image with object proposal windows in the retrieval set. To this end, we generate deep convolutional neural network features using the VGG CNN-M-1024 network, which was demonstrated to provide high-quality features for object classification in the experimental evaluation of Chatfield et al. [19]. To obtain the feature descriptor for an object proposal window, we resize the cropped image window to the canonical size required by the convolutional neural network, perform a forward pass through the network, and take the 1024-dimensional response. We use Chatfield et al.'s CNN-M-1024 network pre-trained on ImageNet as an off-the-shelf feature extractor, without any fine-tuning on our scene parsing images. We append the scaled, normalized spatial coordinates of the window centroid to produce a final 1026-dimensional feature descriptor $\mathbf{f}$, following the common practice of spatial coding in object recognition [11, 90, 134], which encourages matches to come from spatially similar regions in the respective images. To our knowledge, this represents the first use of deep features in a nearest neighbor search setting for scene parsing.

Let $\{W_1, W_2, ...W_m\}$ denote the set of object proposal windows in the query image. For each window, we search for its closest object proposal window in the retrieval set by matching the region descriptors $\mathbf{f}$. Each object proposal window in the database has an associated ground-truth 2D label field L appropriately cropped from the labelled database images. The search therefore produces a set of label fields $\{L_1, L_2, ..., L_m\}$. We warp these retrieved label fields to match the sizes of the query proposal windows, and denote the resized label fields by $\{L'_1, L'_2, ..., L'_m\}$.

Next, let $\{M_1, M_2, ..., M_m\}$ denote label masks constructed from $\{L'_1, L'_2, ..., L'_m\}$. Each mask M_j is the size of the query image and is zero (null) everywhere except within the corresponding object proposal window W_j, where it is populated with the values of L'_j.

The unary potential associated with assigning a category label l to pixel i is then given by

$$\psi_{i,l}(x_i) = -\sum_{j=1}^{m} \delta[M_j(i) = l]\phi(l) \tag{2.5}$$

where $\delta[\cdot]$ is the indicator function and $M_j(i)$ is the value of M_j at location i, which can be a category label or null. The term $\phi(l)$ is a weight that is inversely proportional to the frequency of category label l in the retrieval set:

$$\phi(l) = \frac{1}{\text{freq}(l)^\gamma} \tag{2.6}$$

where freq(l) denotes the number of pixels with category label l in the retrieval set. Eq. 2.6 performs a softened inverse document frequency (IDF) re-weighting to account for differences in the frequency of categories. The constant γ controls the strength of the penalty given to high frequency categories, and as we show later in the experiments, influences the tradeoff between overall per-pixel and average per-class accuracy. After computing $\psi_{i,l}$ for all categories and pixels in the query image, all values are rescaled to be between -1 and 0.

Instead of constructing the label masks $\{M_1, M_2, ..., M_m\}$, the unary potential computation is implemented more efficiently in practice by the accumulation of unary factors [76]. Each of the m unary factors has a spatial extent over pixels corresponding to an object proposal window in the query image. The jth factor adds a single vote to each pixel within its spatial extent, for the corresponding

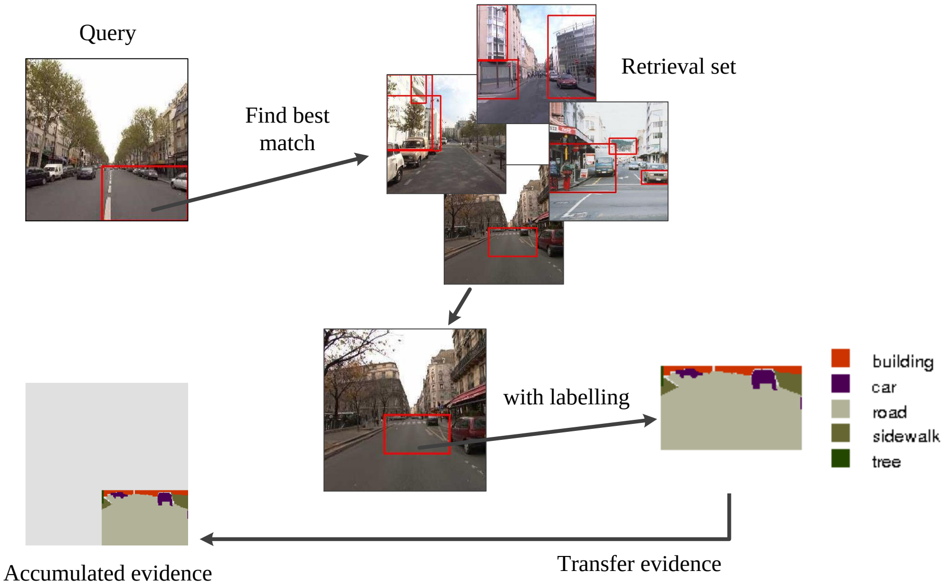


Figure 2.3: Visualization of the computation of the unary potential ψ_i by label transfer. The contribution of one object proposal window in the query image is depicted. The steps are explained in Section 2.2.4.

category label in L'_j. Figure 2.3 illustrates matching and label transfer for a single object proposal window W_j. Conceptually, the unary potential is accumulated in an overlapping, collage-like manner by transferring the category labels from the most similar object proposal windows in the retrieval set.

Fig. 2.4 visualizes the unary potential for three example query images from the SIFT Flow dataset [82] after all m region proposals have been processed. Saturated colors represent higher values of $\psi_{i,l}$.

2.2.5 Performing MRF inference

The unary potential $\psi_i(x_i)$ is combined in (2.4) with a pairwise potential $\psi_{ij}(x_i, x_j)$ defined over pairs of adjacent pixels. In general, the pairwise potential in an MRF imposes a smoothness prior in the labelling. We adopt the same pairwise potential term as in SuperParsing [127], which is based on the co-occurrences of category labels:

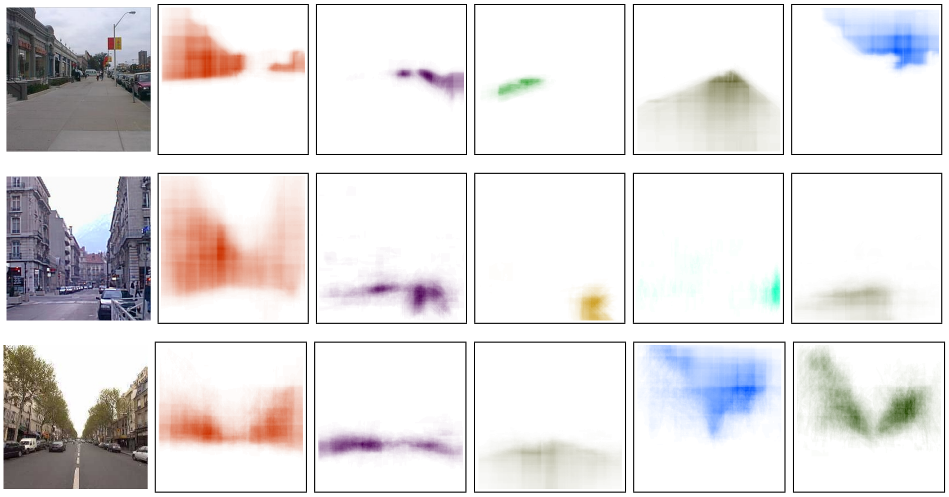


Figure 2.4: Visualization of the unary potential for three sample query images. Top row, from left to right: building, car, plant, sidewalk, and sky categories. Middle row, from left to right: building, car, fence, person, and road categories. Bottom row, from left to right: building, car, road, sky, and tree categories.

$$\psi_{ij}(x_i, x_j) = -\lambda \log[(P(x_i|x_j) + P(x_j|x_i))/2]\delta[x_i \neq x_j] \tag{2.7}$$

where λ is the MRF smoothing constant. Intuitively, this pairwise potential term biases the labelling towards category transitions that are more frequently observed.

We minimize the MRF energy (2.4) approximately using α/β-swap, a standard graph cuts technique [13, 14, 67]. A non-metric version of α-expansion [110] could also be applied for faster inference. We report α/β-swap inference here to be consistent with our journal publication [132], and because we will adopt an entirely different and more powerful random field model and inference algorithm when we extend the CollageParsing framework to 3D scene parsing of video in the next chapter.

Finally, the labelling is spatially refined in a post-processing step by aligning the labels to the query image's superpixels. All pixels within a superpixel are assigned the most common (mode) label in the superpixel. Following SuperParsing [127], superpixels are extracted using the graph-based method of Felzenszwalb and

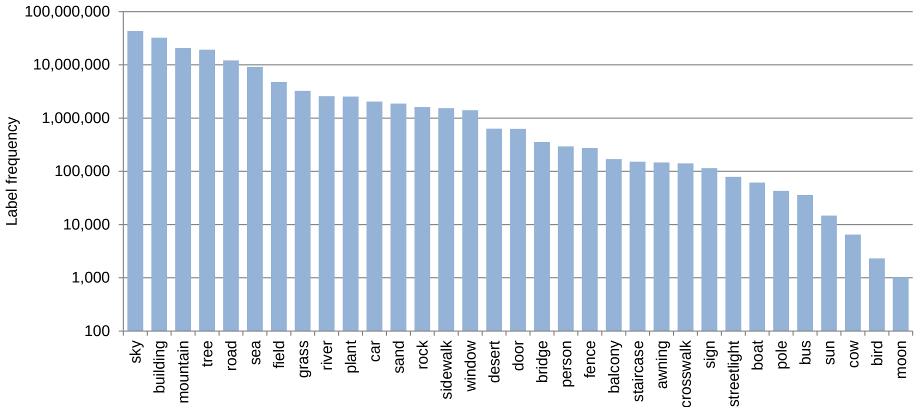


Figure 2.5: Frequency counts of the semantic categories in the SIFT Flow dataset

Huttenlocher [32]. Post-processing produces more visually pleasing results and has only a small effect on labelling accuracy.

2.3 Experiments

We performed experiments on the SIFT Flow dataset [82], a scene parsing benchmark consisting of 200 query images and 2,488 database images from LabelMe. Images span a range of outdoor scene types, from natural to urban. Pixels are labelled with one of 33 semantic categories. The label frequencies are shown in Figure 2.5.

Table 2.1 shows the experimental results and a comparison with recent non-parametric and parametric approaches on this dataset. We set the retrieval set size K = 400, γ = 0.4, and the MRF smoothing parameter λ to 0.01. We investigate the effect of these parameters on the algorithm performance later in this section. Both the overall per-pixel accuracy and the average per-class accuracy are reported. Average per-class accuracy is a more reliable measure of how well the algorithm performs across different categories and not just on the most commonly occurring ones.

By performing label transfer at the level of object proposal windows, CollageParsing obtains state-of-the-art labelling results on the benchmark. An early version of CollageParsing using traditional HOG features and Alexe et al.'s object-

Table 2.1: Overall per-pixel and average per-class labelling accuracy on the SIFT Flow dataset [82]

	Per-pixel	Per-class
Nonparametric scene parsing methods		
Liu et al. [82]	76.7	-
Gould and Zhang [44]	65.2	14.9
Tighe and Lazebnik [127]	77.0	30.1
Myeong et al. [91]	77.1	32.3
Eigen and Fergus [26]	77.1	32.5
Singh and Košecká [118]	79.2	33.8
CollageParsing	**79.9**	**49.3**
Parametric scene parsing methods		
Tighe and Lazebnik [128]	78.6	39.2
Farabet et al. [28], "natural"	78.5	29.6
Farabet et al. [28], "balanced"	74.2	46.0
Yang et al. [146]	79.8	48.7

ness [1] already obtained a 7 to 11% improvement in average per-class accuracy over superpixel based nonparametric approaches [26, 91, 118, 127]. By using off-the-shelf deep features and the Edge Boxes method, the current version of CollageParsing realizes gains of 15 to 19% average per-class accuracy in comparison to recent nonparametric approaches.

Labelling performance is also comparable to or better than recent parametric methods, while not requiring expensive model training. For example, on the LM+SUN dataset (45,000 images with 232 semantic labels), just the per-exemplar detector component of Tighe and Lazebnik [128] requires four days on a 512-node cluster to train. On a dataset of 715 images with eight semantic labels [45], Farabet et al. [28] requires "48h on a regular server" to train. The bulk of our computation time comes from feature extraction. Once features are computed, matching object proposal windows and approximately solving the Markov random field by graph cuts takes 10s in Matlab on a desktop. On average, 855 content-adaptive windows are generated per database image using Edge Boxes, and the nearest-neighbor window matching and label transfer account for 8 of those 10 seconds.

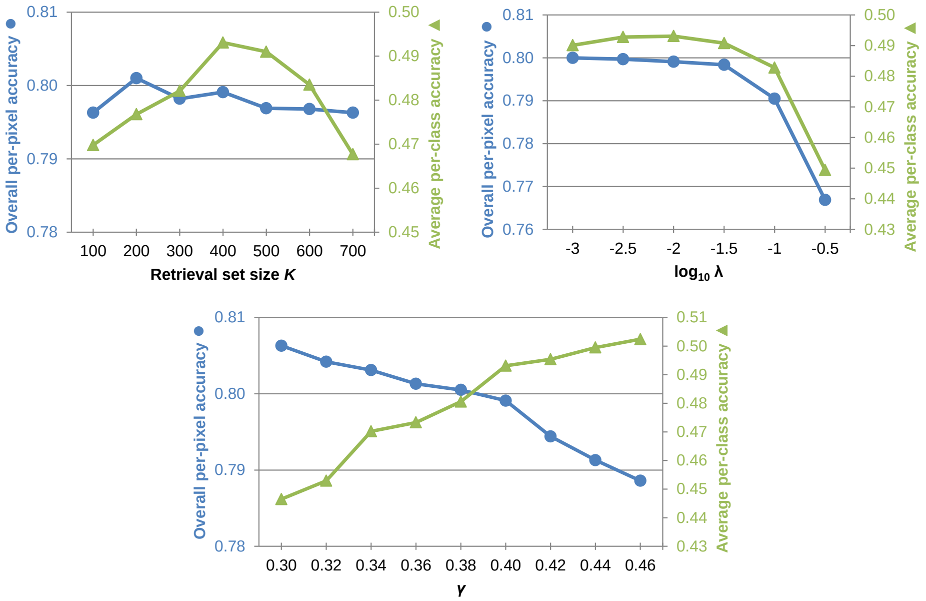


Figure 2.6: Effect of varying the retrieval set size K, the MRF smoothing parameter λ, and γ on overall per-pixel accuracy and average per-class accuracy on the SIFT Flow dataset.

Results are comparable to the hybrid parametric approach recently proposed by Yang et al. [146], which achieves a large increase in average per-class accuracy (nearly 18% compared to the authors' baseline system) by augmenting the retrieval set with examples from rare classes to improve rare class labelling. Interestingly, this idea of 'rare class expansion' is complementary to our work and could be an interesting direction for future investigation. However, from a nonparametric design perspective, 'rare class expansion' would make CollageParsing more difficult to scale because as new images or categories are added to the database, the auxiliary set needs to be re-composed – both the selection of 'rare' categories and the examples themselves.

Figure 2.6 shows the effect of varying γ, the retrieval set size K, and the MRF smoothing parameter λ on the overall per-pixel and average per-class accuracy. Similar to Tighe and Lazebnik [127], we observed that the overall per-pixel accuracy drops when the retrieval set is too small for sufficient matches, but also when

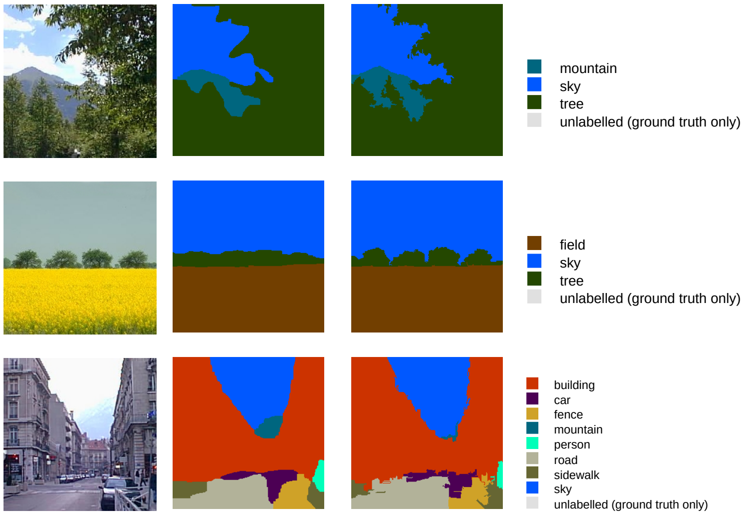

mountain
sky
tree
unlabelled (ground truth only)
field
sky
tree
unlabelled (ground truth only)
building
car
fence
mountain
person
road
sidewalk
sky
unlabelled (ground truth only)

Figure 2.7: Examples of scene parsing results with and without the spatial refinement post-processing step. From left to right: query image, labelling without post-processing, labelling with post-processing.

the retrieval set becomes large, confirming that the retrieval set performs a filtering role in matching query windows to semantically relevant database images. The constant γ controls the strength of the penalty given to high frequency categories. As γ increases, evidence for high frequency categories is discounted more heavily, and labelling is biased towards rarer categories. As reflected in the figure, this tends to increase the average per-class accuracy but at the expense of overall per-pixel accuracy. The post-processing step of spatial refinement using superpixels produces more visually pleasing results and has only a small effect on accuracy: about 1% overall per-pixel accuracy and 0.1% average per-class accuracy. Figure 2.7 shows some examples of scene parsing results with and without superpixel post-processing.

It can be difficult to interpret small differences in per-pixel accuracy between scene parsing alternatives. When ground truth labellings are crowd-sourced from a broad pool of users, with varying experiences and pre-conceptions about what consitutes a correct labelling, it may be unclear if such differences are significant. To address this common limitation in scene parsing evaluation, we additionally explored qualitative scene parsing results. Figure 2.8 shows some qualitative scene

Table 2.2: Overall per-pixel and average per-class labelling accuracy on the LM+SUN dataset [127]

	Per-pixel	Per-class
Nonparametric scene parsing methods		
Tighe and Lazebnik [127]	54.9	7.1
CollageParsing	60.8	**19.3**
Parametric scene parsing methods		
Tighe and Lazebnik [128]	**61.4**	15.2
Yang et al. [146]	60.6	18.0

parsing results, including query images, system predicted labellings for both SuperParsing [127] and CollageParsing, and ground truth labellings. We found CollageParsing to perform well in a wide range of outdoor scenes, from natural (top) to urban (bottom) environments. In comparison to SuperParsing [127], CollageParsing more reliably detects smaller and rarer categories, such as the streetlight and sign in the fourth row, the staircase in the fifth row, and the pedestrians and fence in the sixth row.

We performed further experiments on a second, larger dataset. The LM+SUN dataset [127] consists of 500 query images and 45,176 database images from the LabelMe [112] and SUN datasets [145]. The dataset is complementary to SIFT Flow in that it contains both indoor and outdoor scenes, while indoor scenes are not represented in SIFT Flow. LM+SUN is annotated with a broad vocabulary of 232 semantic categories.

Table 2.2 shows evaluation results and a comparison with recent nonparametric and parametric approaches. Since LM+SUN contains a larger number of images and categories, we increased the retrieval set size K to 1600. The trade-off of γ is shown in Fig. 2.9. LM+SUN is a challenging benchmark but CollageParsing obtains encouraging results. An improvement of 6% per-pixel and 12% average per-class accuracy is obtained relative to SuperParsing. Compared with state-of-the-art parametric methods, a modest gain of 1% average per-class accuracy is achieved (or 7% relative). Inference on LM+SUN is more time consuming than

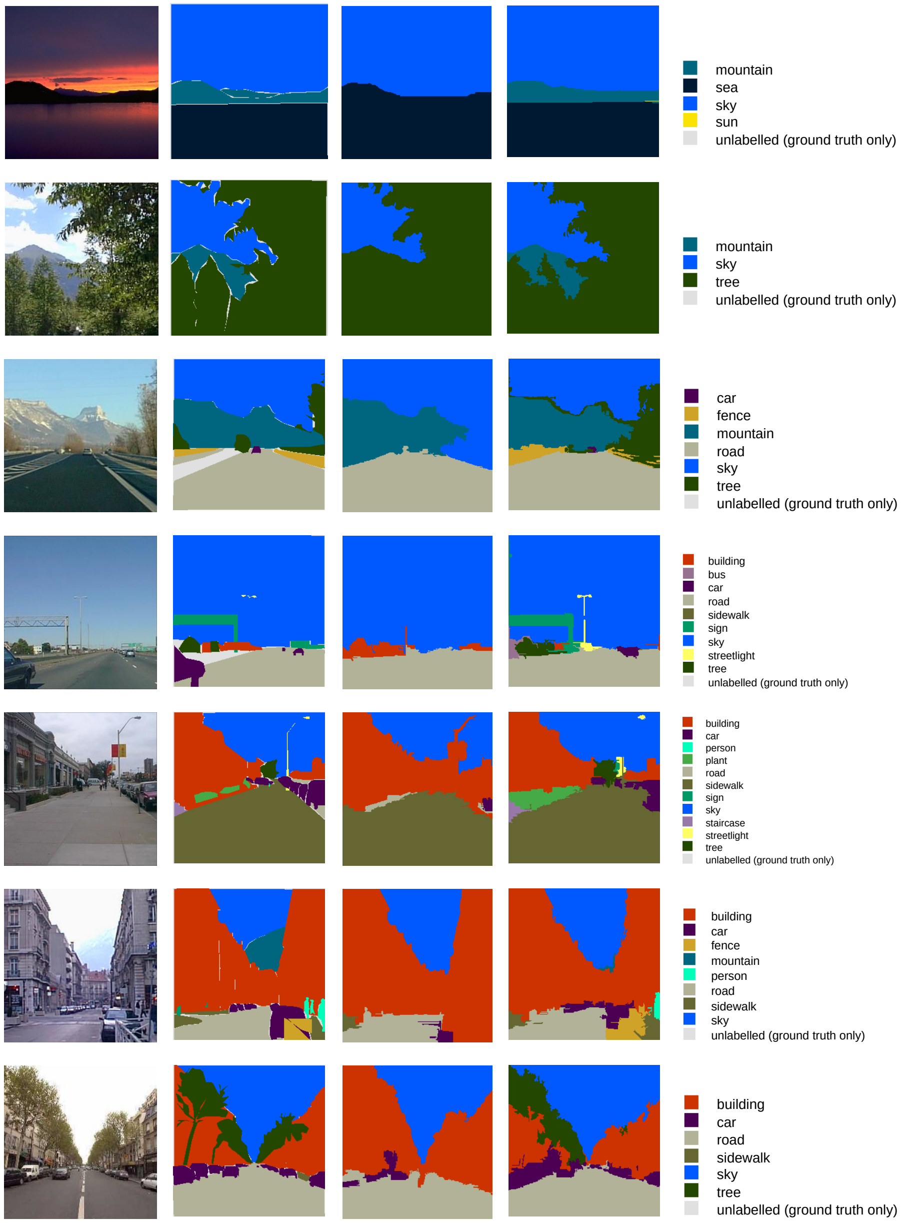


Figure 2.8: Examples of scene parsing results on the SIFT Flow dataset. From left to right: query image, ground truth labelling, SuperParsing [127] predicted labelling, CollageParsing predicted labelling.

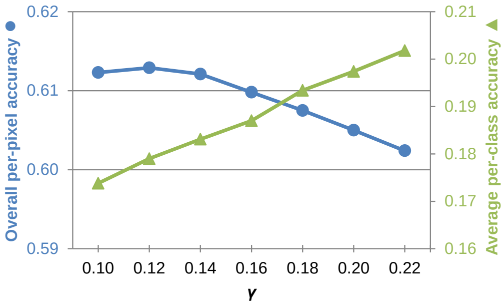


Figure 2.9: Varying γ to trade off overall per-pixel accuracy and average per-class accuracy on the LM+SUN dataset

on SIFT Flow because of the increased retrieval set size and number of categories. Once features are computed, matching and inference take approximately 3 minutes in Matlab on a desktop, which is about half the runtime of Yang et al. [146] on this dataset. No model training is required.

Figure 2.10 shows additional qualitative results on indoor scenes, which are not represented in the SIFT Flow dataset. Similar improvements can be seen on the indoor scenes as on the outdoor scenes presented earlier: in comparison to SuperParsing, CollageParsing is again more accurate for smaller and rarer categories, as well as overall.

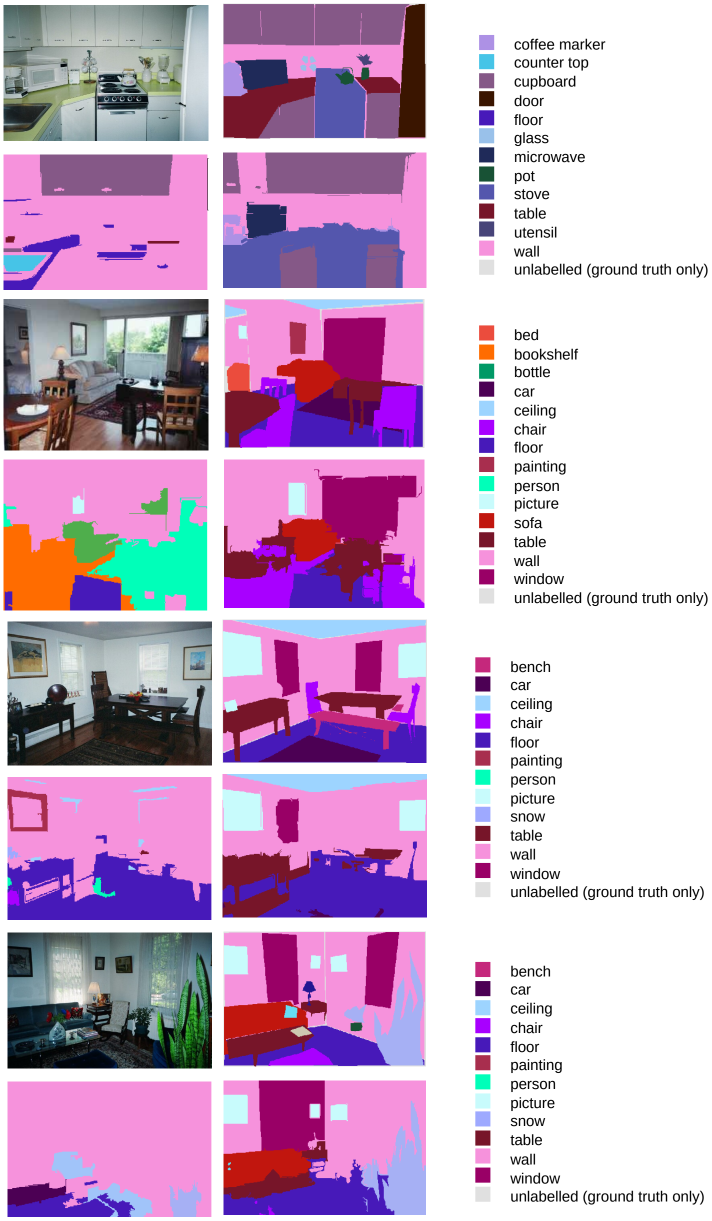


Figure 2.10: Examples of indoor scene parsing results on the LM+SUN dataset. Top left: query image; top right: ground truth labelling; bottom left: SuperParsing [127] predicted labelling; bottom right: CollageParsing predicted labelling.

Chapter 3

Nonparametric 3D Scene Parsing of Video

In this chapter, we extend CollageParsing to 3D scene parsing of video. We are interested in semantically reconstructing the 3D scene imaged by a moving camera in an online manner (Fig. 1.2). Our proposed algorithm is online in the sense that it uses only sensory data that has been observed so far by the platform, as opposed to offline semantic reconstruction, in which the entire video sequence may be loaded for post-analysis. Online operation is required in many applications, including self-driving vehicles and mobile robots [113, 135, 138].

As in 2D scene parsing, the dominant contemporary paradigm for performing 3D semantic reconstruction is that of parametric, discriminative classification (see Section 1.2 for a discussion). This chapter presents, to our knowledge, the first method to approach 3D semantic reconstruction non-parametrically. Our method, called MF3D (Model-Free 3D), approaches 3D voxel labelling via search-based label transfer. There is no explicit appearance model or trained classifier to decide whether a 3D voxel should be labelled a particular class versus others. Instead, MF3D searches over a database of labelled examples to determine how to best label a novel 3D voxel. MF3D uses several state-of-the-art techniques to obtain rich representations of the local scene structure, and applies an intuitive label transfer operation similar to CollageParsing to accumulate evidence for labels represented in the database. This evidence is combined in a principled way via dense conditional random field (CRF) inference [69] to obtain a dense, online semantic labelling of the scene in 3D. We demonstrate that our model-free approach enables accurate online 3D scene parsing while retaining scalability to new categories.

3.1 Related work

The problem of building a dense, semantically-labelled, 3D reconstruction of the environment has generated broad research interest. Brostow et al. [15] were the first to propose using structure from motion to inform scene parsing in video. Structure from motion is used to obtain sparse 3D point clouds, from which several simple features are extracted and input to a trained random forest classifier. The authors demonstrated that these structure and motion features can be combined with appearance features to obtain more accurate results than either feature type by itself. Sturgess et al. [124] combined Brostow et al.'s structure from motion features with appearance features in a higher-order CRF model to perform scene parsing of road scenes in video.

Instead of using structure from motion as a source of complementary features, other techniques integrate 3D information directly into the scene parsing framework. Xiao and Quan [144] developed an offline scene parsing algorithm for video captured by a laterally-moving camera (e.g. curb-facing camera on a moving vehicle). The authors construct an MRF over superpixels in the video, in which superpixels from different frames are connected by pairwise potentials based on structure from motion correspondences. Unary potentials are computed using a boosted classifier trained on 2D appearance, height, and 3D geometry features. Floros and Leibe [33] extended a hierarchical CRF scene parsing model [66, 77] to video by introducing new potentials for temporal consistency. 3D scene information is obtained by a point cloud reconstruction. A higher-order potential is introduced that encourages projections of the same 3D point to be assigned the same label. In addition, a 3D unary potential is added. Häne et al. [46] proposed a joint 3D reconstruction and scene parsing framework that integrates class-specific geometric priors based on surface normals. Sengupta et al. [113] obtained 3D semantic reconstructions of street scenes by fusing the output of 2D scene parsing [77] with 3D surface reconstruction from stereo images. Valentin et al. [135] developed a mesh-based method in which depth estimates from RGB-D cameras or stereo pairs are fused into a 3D mesh model and a CRF is defined over the mesh faces. Unary potentials are computed using a cascaded classifier trained on geometry, appearance, and contextual features. The method is offline: the full 3D mesh model is constructed once and then labelled. Riemenschneider et al. [108] improved the efficiency of offline scene parsing by removing view redundancy and reducing the number of sites requiring unary potential computation. Multi-view reconstruction is used to obtain a 3D mesh model. For each triangle (face) in the mesh, only the single predicted best view is used to compute the unary potential. Triangles are also ranked and unary potentials are computed at only the top triangles. Kundu et al. [76] combined sparse visual SLAM and 2D scene parsing in a 3D CRF framework to infer semantically-labelled 3D scene structure with a monocular camera. However, the projected 2D image labellings tend to be somewhat blocky or voxelated. Vineet et al. [138] proposed an online 3D scene parsing method that combines large-scale 3D reconstruction with a novel handling of moving objects. Voxels corresponding to moving object classes (e.g. cars and people) are allowed to update more quickly. A dense 3D CRF is constructed over voxels in the current view. Unary potentials are computed by projecting 2D unary potentials onto the voxels; 2D unary potentials are computed in image space using a random forest trained on appearance and position features. Martinović et al. [88] parsed building facades in 3D by performing point cloud labelling using a random forest, inferring facade separations based on differences in the label structure, and enforcing weak architectural rules such as alignment and co-occurrence (e.g. a balcony must have at least one window above it).

Similar to this past work, we perform scene parsing in 3D. However, all of these methods for 3D scene parsing involve the training of discriminative classifiers that decide whether a pixel or voxel should be labelled a particular category versus the others in a pre-defined set of categories. For example, a classifier may be trained to distinguish a tree pixel versus a pixel of car, road, or building. To the best of our knowledge, MF3D is the first method for 3D scene parsing of video that is nonparametric. Our experiments confirm that a nonparametric search-based approach can produce accurate 3D labelling results while retaining extensibility to novel categories.

3D semantic scene parsing is distinct from 3D object detection in that the aim of scene parsing is to produce a fully labelled reconstruction, including both foreground objects and background texture categories. However, 3D object detection methods may employ similar reconstruction techniques: for example, Tateno et al. [126] proposed an object recognition framework that builds on top of KinectFusion reconstruction [92] and can localize pre-trained, fully 3D objects in clutter.

While our focus is on video sequences, 3D semantic scene parsing can also be approached by other input modalities. Bláha et al. [10] proposed an adaptive volumetric model for 3D scene parsing at a city scale based on aerial images. Their volumetric model is hierarchical and the resolution is refined only near surfaces. Given the point cloud scan of an entire building, Armeni et al. [4] parse the point cloud into rooms and further parse the rooms into structural elements (e.g. walls, ceiling) and furniture elements.

3.2 Algorithm description: MF3D

Fig. 3.1 illustrates the MF3D processing pipeline. MF3D takes as input a pair of calibrated stereo RGB images. From this stereo pair, a disparity map is calculated and depths are estimated for each imaged scene point. The estimated depths are used to incrementally construct a 3D volumetric representation of the surroundings. The 3D reconstruction is stored and indexed efficiently using voxel hashing. Next, a nonparametric label transfer operation is performed in image space, in which category labels from a database of labelled 2D images are transferred to the current image by nearest neighbor matching. The label evidence is then projected into the 3D scene to form the unary potentials of a 3D CRF defined over voxels. The CRF is solved in 3D using an efficient approximate inference algorithm to obtain the current labelling of the 3D scene. The pipeline updates the labelled 3D reconstruction as new observations arrive (in other words, in an online manner), and does not require knowledge of the entire video sequence. The following sections describe each of these components in detail.

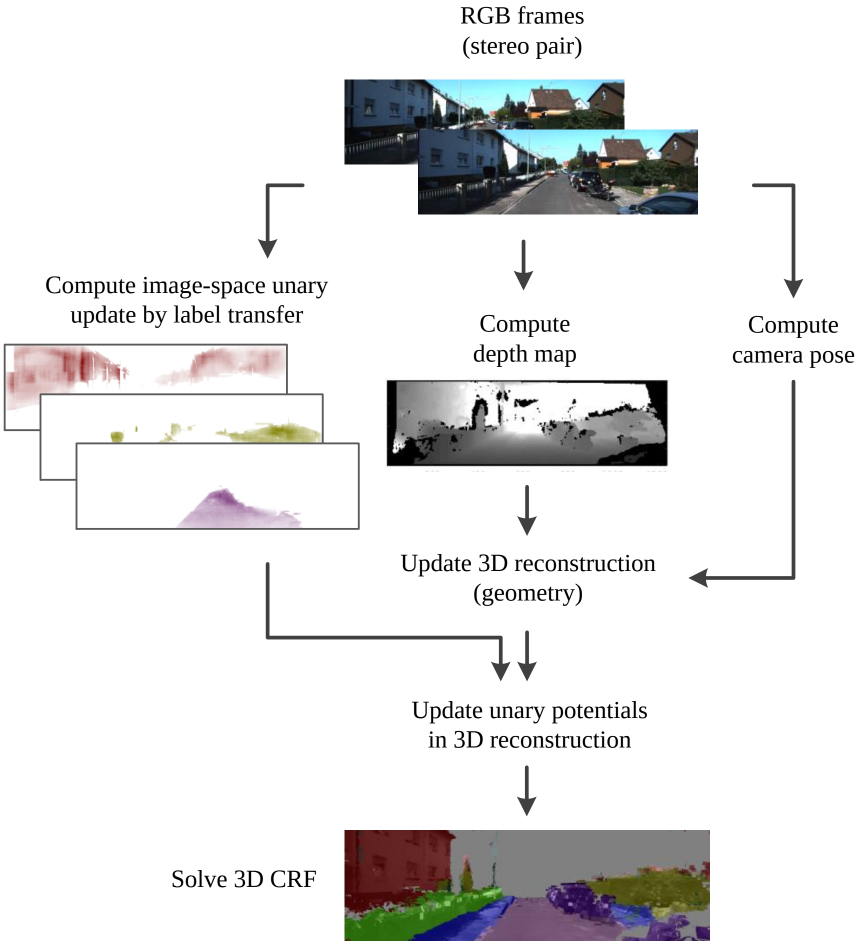


Figure 3.1: Overview of the MF3D processing pipeline for 3D scene parsing. MF3D takes as input a pair of stereo RGB images, from which depth and the current camera pose with respect to the 3D reconstruction are estimated (Section 3.2.1). A label transfer operation is performed in image space and projected into the 3D reconstruction as a unary potentials update. A 3D CRF, defined over visible voxels, is solved to obtain a semantic labelling of the scene in 3D (Sections 3.2.2 and 3.2.3).

3.2.1 Preliminaries: 3D reconstruction

The base representation of our 3D scene model is a sparse volumetric representation in which each allocated voxel stores a truncated signed distance function (TSDF) value [21] indicating the distance to the closest surface. The TSDF is an implicit representation of a 3D surface, which is captured as the zero level set (free space is represented as positive values and occupied space as negative values). Given a camera pose, a view of the model surfaces can be obtained by raycasting. Besides the TSDF value, we also store in each voxel a mean RGB value and a k-dimensional array containing unary potentials (‘evidence’) accumulated so far for the k semantic categories.

Modern 3D reconstruction pipelines commonly consist of tracking and mapping modules. Tracking involves localizing the camera within the 3D reconstruction, and mapping (or fusion) involves integrating new sensor observations into the 3D reconstruction. Klein and Murray [65] were the first to separate tracking and mapping into separate tasks that are run in parallel, in contrast to previous SLAM approaches. Their Parallel Tracking and Mapping (PTAM) method can rapidly estimate the pose of a calibrated hand-held camera and build a sparse 3D reconstruction of a small workspace environment. In the tracking task, map points are projected into the image based on a prior pose estimate given by a motion model, coarse-to-fine matching of features is performed, and the updated camera pose is obtained by iterative minimization of the reprojection error. The mapping task is initialized using a stereo technique and expanded over time with new keyframes, with bundle adjustment as resources permit. Newcombe et al.'s Dense Tracking and Mapping (DTAM) method [93] performs dense image alignment instead of feature matching, enabling more robust tracking under rapid motion.

Newcombe et al.'s KinectFusion method [92] took advantage of the newly introduced Kinect depth sensor and contemporary GPU hardware to obtain real-time dense 3D reconstruction of room-sized environments. Kinect sensor observations are fused into a dense TSDF voxel grid. Tracker drift is significantly reduced by estimating the new camera pose with respect to the complete current reconstruction instead of the previous frame. The camera pose is estimated using coarse-to-fine iterative closest point (ICP) alignment between the live depth image and the raycasted surface prediction.

Scaling a 3D reconstruction to large spaces is challenging as the memory requirements can quickly become impractical. Nießner et al. [94] extended dense TSDF reconstruction to larger scale environments by introducing voxel hashing (not to be confused with binary hashing methods for efficient search in Section 4). Voxel hashing reduces the memory requirements of the 3D reconstruction: instead of allocating a regularly spaced voxel grid, voxels near surfaces are allocated as needed as observations arrive. Empty space remains unallocated. Voxel blocks are indexed by world coordinates in a hash table based data structure.

Our 3D reconstruction framework builds on top of InfiniTAM [64], a highly streamlined implementation of the modern 3D reconstruction pipeline with a dense TSDF 3D reconstruction indexed using a modified version of voxel hashing. Optimizations of voxel hashing, fusion, and raycasting allow real-time 3D reconstruction to be achieved on a tablet computer. The authors of InfiniTAM report an order of magnitude speed-up over KinectFusion [92] and Nießner et al. [94]. We incorporate modifications to InfiniTAM's tracking and mapping (fusion) components.

Instead of using InfiniTAM's default iterative closest point (ICP) tracker, which we found to be unreliable for outdoor driving sequences, we track the camera pose using FOVIS [57]. FOVIS is an algorithm for visual odometry. At each time step, the algorithm accepts a greyscale image and a depth map as input, and returns the integrated pose estimate. The visual odometry pipeline consists of feature extraction, initial rotation estimation, feature matching, inlier detection, and finally motion estimation. First, FAST features [109] are extracted from the image at multiple scales. An initial rotation is then estimated to constrain feature matching across frames. Next, features are matched and consistent inlier matches are detected using a greedy maximum-clique technique. Finally, a motion estimate is obtained by iteratively minimizing the feature reprojection error and updating the inlier set.

For mapping (fusion) in our modified InfiniTAM implementation, the TSDF, RGB, and unary potential values associated with each voxel are updated as a running average in an online manner as new sensor observations arrive. Observations at each time step include the RGB image and a disparity map estimated using stereo. Disparity is converted to depth by $z = bf/d$, where z is the depth, b is the stereo baseline, f is the camera's focal length, and d is the disparity. In particular, we estimate the disparity map from the stereo RGB pair using the Efficient Large-scale Stereo (ELAS) method of Geiger et al. [37]. ELAS is a Bayesian method for stereo matching that forms a prior on the disparities by triangulating a set of robust point correspondences, referred to as support points. Given a disparity for a pixel in the left (right) image, an observation likelihood is computed for the corresponding pixel in the right (left) image by comparing fast Sobel-based features. The disparity map is obtained by maximum a-posteriori (MAP) inference at each pixel, which can be easily parallelized and does not require an expensive global optimization.

3.2.2 3D CRF model

MF3D constructs a dense conditional random field (CRF) over the 3D reconstruction visible in the current frame. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph defined over the visible 3D reconstruction, where $\mathcal{V}$ is a set of nodes corresponding to the visible voxels and $\mathcal{E}$ is a set of undirected edges. Let X_i be the random variable associated with node $i \in \mathcal{V}$. X_i can take on a label from the label set $\mathcal{L} = \{l_1, l_2, ..., l_k\}$, corresponding to the vocabulary of k semantic category labels. Denote by x_i a realization or assignment of random variable X_i, and by $\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}$ the assignments of all random variables; that is, $\mathbf{x}$ represents a complete labelling of the image. The energy (or cost) function associated with a particular labelling of the CRF is given by

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \psi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \tag{3.1}$$

where $\psi_i(x_i)$ are unary potentials and $\psi_{ij}(x_i, x_j)$ are pairwise potentials, both implicitly conditioned on the observed data. The task of inference is to find the labelling $\mathbf{x}^*$ with the lowest energy:

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{L}^{|\mathcal{V}|}} E(\mathbf{x}) \tag{3.2}$$

Unary potentials represent the costs associated with independently assigning a voxel a particular semantic category label. The conventional practice is to compute the unary potentials using a trained discriminative classifier. This classifier is trained to distinguish between instances of the categories in $\mathcal{L}$. In contrast to conventional practice, MF3D computes the unary potentials via the nonparametric, search-based label transfer mechanism described in the next section.

Pairwise potentials represent the costs associated with assigning two connected voxels two particular semantic category labels. For example, the cost of assigning the same semantic label to two voxels that are close in the 3D world and similar in color appearance should be low. Connectivity is defined in terms of the edges $\mathcal{E}$ in graph $\mathcal{G}$ and does not necessarily correspond to spatial adjacency. In our model, $\mathcal{G}$ is densely connected: that is, there exists a connection between every possible pairing of voxels in $\mathcal{V}$. This is in contrast to many traditional labelling methods in which only adjacent neighbours are connected. The dense connectivity enables encoding of long-range patterns. Krähenbühl and Koltun [69] demonstrated that approximate inference in a densely connected CRF, in which the pairwise potentials are defined by a linear combination of Gaussian kernels, can be performed efficiently by optimizing a mean-field approximation to the CRF distribution. Optimizing the mean field approximation normally involves iterative message passing that is quadratic in the number of variables (voxels). Krähenbühl and Koltun's insight is that the message passing can be implemented as Gaussian filtering in feature space. By applying a fast approximate high-dimensional filtering technique, inference becomes linear in the number of variables and sublinear in the number of edges. We adopt Krähenbühl and Koltun's inference method to obtain a labelling over voxels, generalizing their original 2D pairwise appearance kernel to 3D (voxels that are close in the 3D world and similar in color are encouraged to share the same label):

$$\psi_{ij}(x_i, x_j) = \lambda \exp\left(-\frac{|\mathbf{p}_i - \mathbf{p}_j|^2}{\sigma_1^2} - \frac{|\mathbf{I}_i - \mathbf{I}_j|^2}{\sigma_2^2}\right) \delta[x_i \neq x_j] \tag{3.3}$$

where λ is the CRF smoothing constant, $\mathbf{p}_i$ and $\mathbf{p}_j$ are voxel positions, $\mathbf{I}_i$ and $\mathbf{I}_j$ are color vectors, σ_1 and σ_2 are Gaussian bandwidth parameters, and $\delta[\cdot]$ is the indicator function.

The 3D CRF is approximately solved each time step in an online manner, using only sensory data obtained so far, and is constructed only over voxels visible in the current time step. Unary potentials at each voxel are accumulated online as a running average over time (Section 3.2.1).

3.2.3 Unary potentials update by label transfer

Recall that label transfer refers to an accumulation of semantic category evidence by transferring the labels of matching exemplars in a labelled database. This is a search-based strategy that differs from the common strategy of training a discriminative classifier to predict the semantic category label. Its main advantages are flexibility and scalability: the approach is naturally extended to new labelled examples or new categories because no intensive re-learning stage is necessary. A new category can be added to the database without having to learn how to discriminate it from all previously learned categories.

In each frame, MF3D computes a unary potential update in image space. The unary potential update is then projected into the online 3D reconstruction to update the unary potentials stored in each voxel. Two questions remain: how can we compute a unary potential update from a database of labelled examples, and what is the best way to project the unary update into the 3D reconstruction?

In the previous chapter on 2D scene parsing, we demonstrated how using mid-level region proposals (also known as object proposals) for label transfer produces more accurate labellings than using low-level superpixels. Intuitively, region proposals are more likely to preserve object boundaries. For MF3D, we generate a set of region proposals for each frame using a state-of-the-art region proposal network (RPN) [107]. Each region proposal is a rectangular window. The appearance of each region proposal is described using a pre-trained VGG-16 network [117] as an off-the-shelf feature extractor, without any fine-tuning for the 3D scene parsing task. In particular, an appearance-based feature descriptor $\mathbf{f}_a$ is obtained by concatenating the activations of the two fully-connected layers of VGG-16. These descriptors are obtained efficiently using region pooling of the convolutional layers (region pooling evaluates the expensive convolutional layers only once for the image instead of once for each window [40, 50]).

In addition, a depth-based feature descriptor $\mathbf{f}_d$ is computed for each region proposal using the depth map estimated in the 3D reconstruction step (Section 3.2.1). The depth-based descriptor $\mathbf{f}_d$ is obtained by taking multiple binary comparisons

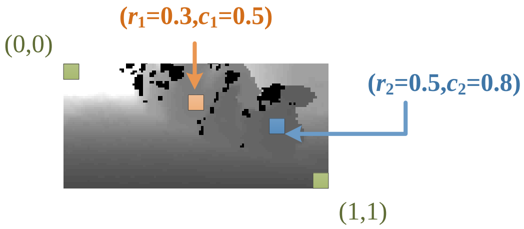


Figure 3.2: The depth feature descriptor $\mathbf{f}_d$ of a window is an n_d-dimensional vector, in which each entry is determined by comparing the estimated depth at two locations (r_1, c_1) and (r_2, c_2). These locations are normalized window coordinates, i.e. $0 \leq r_1, c_1, r_2, c_2 \leq 1$. If the first location is at a greater depth, the entry in $\mathbf{f}_d$ is set to +1; otherwise, the entry is set to -1.

of the estimated depth at random relative locations within the region proposal. Let $R = \{(r_1, c_1), (r_2, c_2)\}^{n_d}$ denote a set of n_d relative location pairs within the region proposal. For the jth pair, we compare the estimated depths at the two locations within the region. If the first location has a greater depth estimate, the jth entry of $\mathbf{f}_d$ is set to +1; otherwise, the jth entry of $\mathbf{f}_d$ is set to -1. Fig. 3.2 illustrates this comparison. R is generated once and applied for each region in the database and in the queries. Each r_1, c_1, r_2, c_2 is drawn uniformly at random from the interval [0,1]. The final region descriptor $\mathbf{f}$ is a concatenation of $\mathbf{f}_a$ and $\mathbf{f}_d$ multiplied by a scaling factor to make the two types of descriptor comparable: $\mathbf{f} = [\mathbf{f}_a, \alpha_d \cdot \mathbf{f}_d]$.

We transfer labels from the database in a manner similar to CollageParsing in the previous chapter. Let $\{W_1, W_2, ...W_m\}$ denote the set of region proposals in the current frame. For each region proposal, we search for its closest region proposal in the database by matching the region descriptors $\mathbf{f}$. Each region proposal in the database has an associated ground-truth 2D label field L, appropriately cropped from the ground truth label images. The search therefore produces a set of label fields $\{L_1, L_2, ..., L_m\}$. We warp these retrieved label fields to match the sizes of the query region proposals, and denote the resized label fields by $\{L'_1, L'_2, ..., L'_m\}$.

Let $\{M_1, M_2, ..., M_m\}$ denote label masks constructed from $\{L'_1, L'_2, ..., L'_m\}$. Each mask M_j is the size of the current frame and is zero (null) everywhere except within the corresponding region proposal W_j, where it is populated with the values of L'_j. The image-space unary potential update associated with assigning a category label l to pixel i is then given by

$$\psi^*_{i,l}(x_i) = -\sum_{j=1}^{m} \delta[M_j(i) = l]\phi_{\text{h}}(i,l)\phi_{\text{sz}}(j)\phi_{\text{idf}}(l) \tag{3.4}$$

where $\delta[\cdot]$ is the indicator function and $M_j(i)$ is the value of M_j at location i, which can be a category label or null. The term $\phi_{\text{h}}(i,l)$ is an importance weight based on the 3D height of pixel i and the height statistics of category label l in the database:

$$\phi_{\text{h}}(l) = \begin{cases} \exp\left(-\frac{(\text{h}(i)-\mu_{\text{h}}(l))^2}{\sigma^2_{\text{h}}(l)}\right) & \text{if } \mu_{\text{h}}(l) < 0 \\ 1 & \text{otherwise} \end{cases} \tag{3.5}$$

where $\text{h}(i)$ denotes the 3D height of pixel i, $\mu_{\text{h}}(l)$ is the mean height of label l, and $\sigma_{\text{h}}(l)$ is the variance of the height of label l. The importance weight is defined only for categories with a mean height below the camera ($\mu_{\text{h}}(l) < 0$) because height estimates are unreliable for objects extending beyond the upper bounds of the camera field of view. Intuitively, the term $\phi_{\text{h}}(i,l)$ assigns greater importance to matches with heights that are consistent with the database examples. The height means and variances of a category can be computed in an online manner as a running average, and independently of other categories. Height statistics therefore easily scale to new labelled examples and new categories.

The term $\phi_{\text{sz}}(j)$ assigns higher weight to smaller region proposals, and is defined as the inverse of the smaller dimension of the proposal window. This term increases the influence of compact object matches and suppresses larger image regions whose labels may be less precisely matched.

Finally, $\phi_{\text{idf}}(l)$ is the inverse document frequency term commonly used in retrieval:

$$\phi_{\text{idf}}(l) = \frac{1}{\text{freq}(l)} \tag{3.6}$$

where freq(l) is the number of pixels with label l in the database region proposals. MF3D does not compute co-occurrence statistics explicitly but captures co-

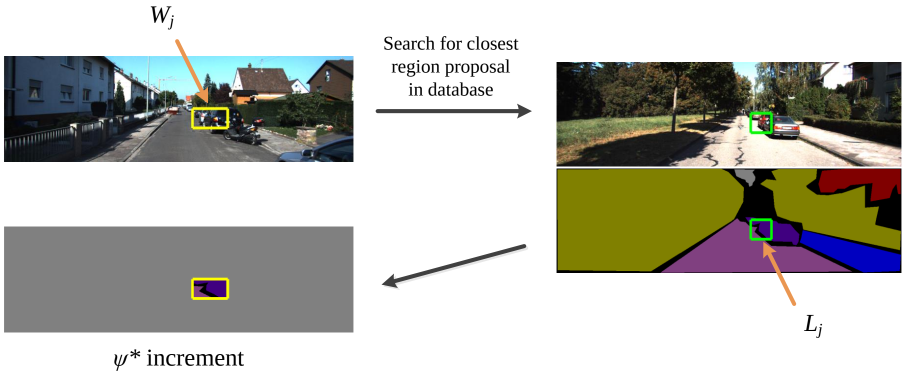


Figure 3.3: Visualization of image-space label transfer for a single region proposal W_j.

occurrence patterns implicitly in the window-based label transfer.

Fig. 3.3 illustrates matching and label transfer for a single region proposal W_j. Instead of allocating memory for individual masks M_j, in implementation we simply accumulate energies directly in $\psi^*_{i,l}$. Fig. 3.4 visualizes the image-space unary potential updates for the same image after all m region proposals have been processed. Saturated colors represent higher values of $\psi^*_{i,l}$.

We have discussed the question of how we can compute a unary potential update from a database of labelled examples, and now arrive at the remaining question: how can we best project the unary potential update into the 3D reconstruction? A straightforward approach would be to simply project the entire $\psi^*_{i,l}$ into the 3D reconstruction. However, we observed that this approach can lead to some bleeding of category energies across object boundaries or discontinuities. This is a consequence of label transfer being coarsely localized and is a limitation of our system when examples are limited. The spatial characteristics of the transferred evidence depend on the positions of the objects in the database examples. If the positions in the database examples do not match the observed frame exactly, there will be some spatial inexactness in the transferred labels. To address this, MF3D performs a segmentation of the frame into superpixels, followed by a winner-take-all pooling operation that selects only the most confident category within each superpixel. Each superpixel S selects a single category label l^s as the mode of the highest-evidenced labels at each pixel:

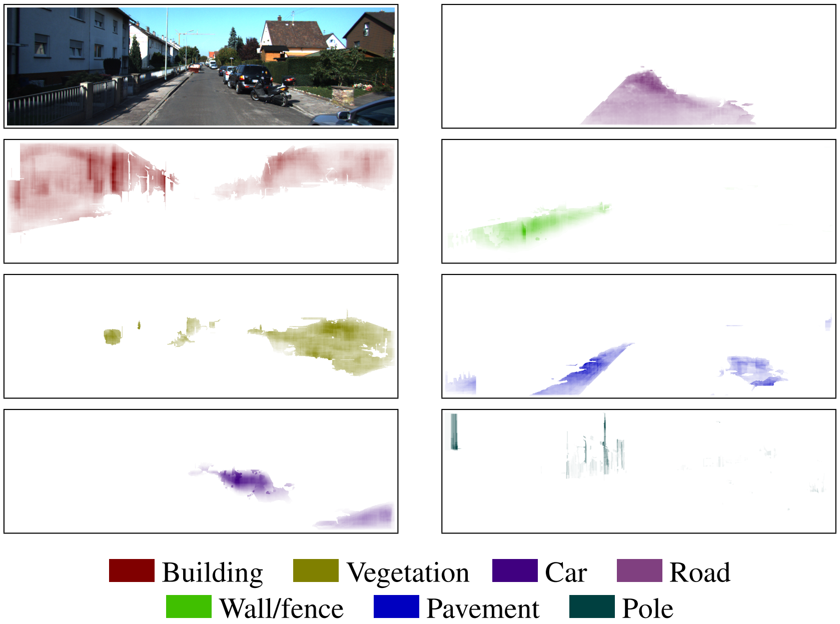


Figure 3.4: Visualization of the image-space unary potential update for a sample image. Higher saturation represents higher values of $\psi^*_{i,l}$.

$$l^s = \operatorname*{mode}_{i \in S} \left\{ \arg\max_{l} \psi^*_{i,l}(x_i) \right\} \tag{3.7}$$

With winner-take-all pooling, the projection of the unary potential update is restricted to a single category label at each point, and this label is the same across all pixels within a superpixel. The unary potential update is zeroed out for all other category labels. This pooling operation improves the alignment of transferred evidence to the observed frame and reduces the bleeding effects caused by differences in the positions of the objects in the database examples. Our experiments indicate that winner-take-all pooling leads to a modest improvement in labelling accuracy, though some bleeding is still noticeable, especially for thin structures such as poles (Fig. 3.7, bottom).

Fig. 3.5 shows a visualization of the 3D voxel unary potentials at the frame shown in Fig. 3.4. The voxel unary potentials are accumulated over multiple frames in an online manner as a rolling average.

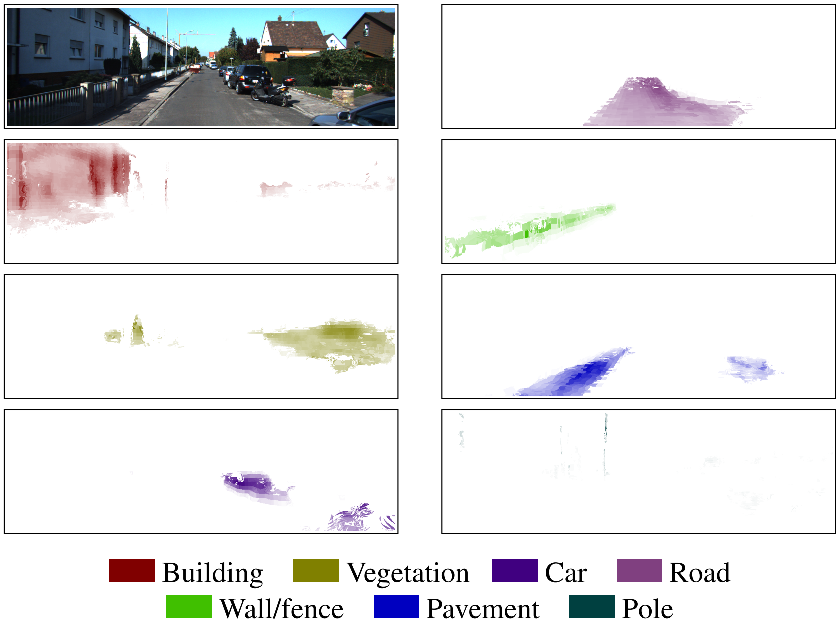


Figure 3.5: Visualization of the 3D CRF unary potentials for a sample image to a maximum depth of 25 meters. Voxel unary potentials are accumulated over multiple frames in an online manner as a rolling average. Higher saturation represents higher values of $\psi_{i,l}$.

3.2.4 Computation time

Most of MF3D's computation time is spent performing label transfer. As described so far, MF3D performs an exact nearest neighbour search during label transfer, which is computationally expensive in terms of both time and memory. It is possible to speed up this expensive operation, while at the same time reducing the memory overhead, by integrating efficient methods for large-scale approximate nearest neighbour search. We address this topic in detail in Chapter 4.

MF3D's 3D reconstruction pipeline has been demonstrated to be runnable in real-time given a modern hardware platform and efficient software implementation [64, 138]. Region proposal generation and description using RPN can performed at 5 fps on a modern GPU [107].

3.3 Experiments

Benchmark details. We benchmarked MF3D on the KITTI labelled dataset [38, 113, 135], a standard benchmark for 3D scene parsing of video. We chose KITTI because it is the common dataset on which the state-of-the-art 3D scene parsing methods [113, 135, 138] have all been evaluated. The KITTI labelled dataset is derived from the broader KITTI benchmark for autonomous driving [38] and consists of four training video sequences (KITTI sequences 05 to 08), totally spanning 5957 stereo pairs, and one independent testing video sequence (KITTI sequence 15), totally spanning 1901 stereo pairs. All sequences are captured at 10 frames per second. The four training sequences are captured at a resolution of 1226×370, while the testing sequence is captured at a resolution of 1241×376. A subset of 45 frames from the training sequences and 25 frames from the testing sequence is manually annotated by the authors of [113, 135] with semantic ground truth labels. Since we build the 3D semantic reconstruction incrementally in an online manner, we process all frames but evaluate only on the 25 in the test sequence with ground truth annotations. Labelling accuracy is assessed for seven semantic categories: building, vegetation, car, road, wall/fence, pavement, and pole. Camera calibration parameters are provided by the authors of KITTI [38].

Implementation details. Since the region proposal network generates windows of limited aspect ratio (1:2, 1:1, 2:1) and some objects of interest are very thin (in particular, poles), we augment the set of network proposals with Edge Boxes [151] narrower than 1:4. Edge Boxes have been independently shown to be good for localization and fast to compute [55]. The feature descriptors for these proposals are generated the same way as the regular network proposals.

Evaluation methodology. Labelling performance is measured by the mean per-class accuracy over the seven classes. Since the ground truth annotations are in 2D, the semantically labelled 3D reconstruction is projected onto the camera plane for evaluation. Evaluation is performed on the online 3D reconstruction (i.e. using only the frames observed so far). Pixels beyond a maximum depth of 25 meters are ignored following Vineet et al. [138].

Results and comparison to state-of-the-art. Fig. 3.6 shows qualitative 3D scene parsing results at frames 3, 36, and 68 in the testing sequence (KITTI sequence

Table 3.1: Comparison to state-of-the-art methods for 3D scene parsing of video on the KITTI labelled benchmark

	3D	Online	Model-free	Mean per-class accuracy
Valentin et al. [135]	✓			67.5
Sengupta et al. [113]	✓			77.2
Vineet et al. [138]	✓	✓		82.2
MF3D	✓	✓	✓	**83.5**

15). These results are generated online, using only the frames observed up to that time. We also render the labelled 3D reconstruction from alternative viewpoints. There are many gaps in the alternative renderings because these are virtual cameras and we have real observations only from the single canonical viewpoint. We can see that MF3D densely reconstructs the 3D scene around the vehicle with mostly accurate semantic labels.

Fig. 3.7 shows typical failure cases. The top row shows a reconstruction failure at frame 639 in the testing sequence, in which the FOVIS tracker fails to accurately estimate the camera pose. This tracking failure results in local corruption in the 3D model. However, the system is able to robustly recover from the failure as it incorporates new sensor observations. The second row shows the labelling result one second later at frame 649. We can see that there only a few erroneous reconstruction artifacts and the labelling of the scene is mostly accurate. The third row shows another typical error case. While the winner-take-all pooling reduces the degree of undesirable 'bleeding' during label transfer, some bleeding still occurs for thin structures such as poles. This is apparent in the bottom row, which shows the reconstruction result at frame 1050.

Table 3.1 shows the quantitative results as well as a qualitative comparison with three strong baselines for 3D scene parsing in video [113, 135, 138]. MF3D achieves labelling accuracy comparable to the state-of-the-art method of Vineet et al. [138] and leads the other recent methods of Sengupta et al. [113] and Valentin et al. [135] by several percent.

Fig. 3.8 shows the sensitivity of MF3D to the depth feature parameters and

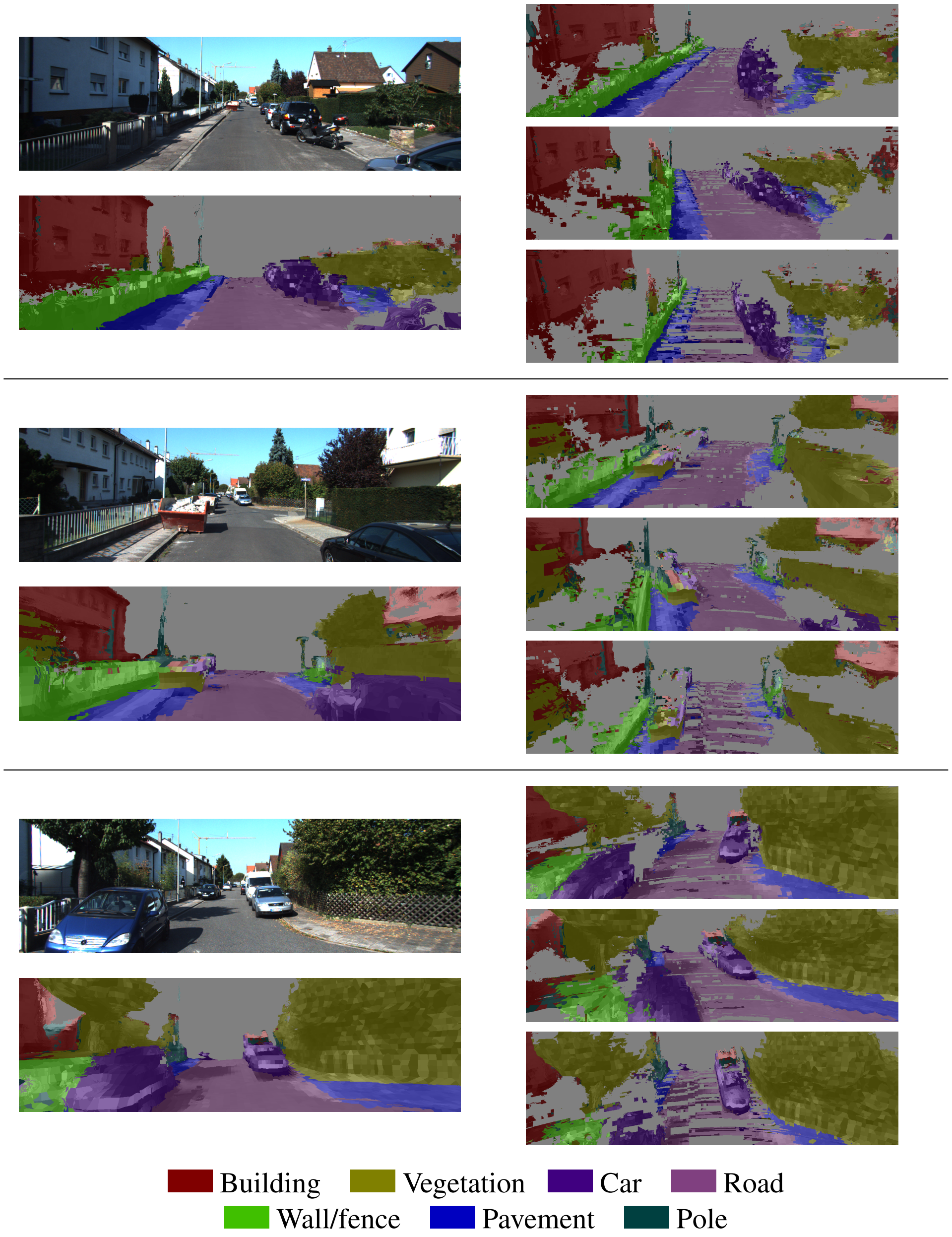


Figure 3.6: Qualitative 3D scene parsing results on the KITTI benchmark. Left column: RGB frame and labelling result from the camera pose. Right column: 3D renderings of the labelling result from different virtual viewpoints. Labelling results are to a maximum depth of 25 meters.

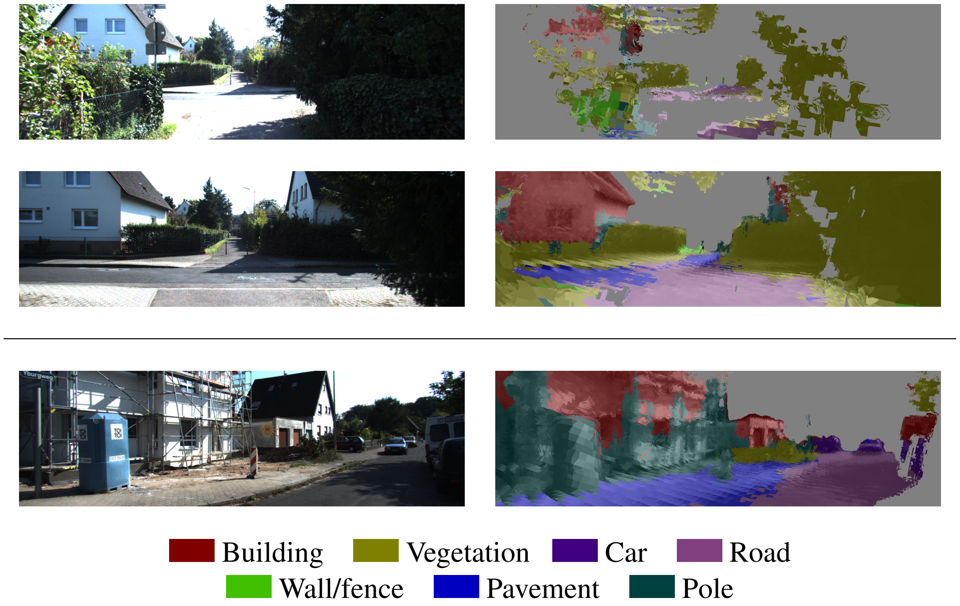


Figure 3.7: Failure cases on the KITTI benchmark (see Section 3.3 for discussion). Labelling results are to a maximum depth of 25 meters.

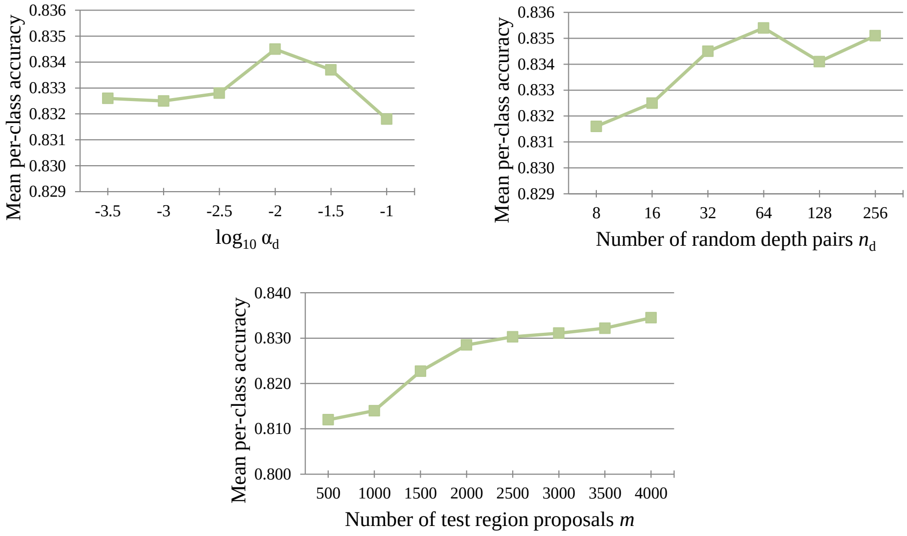


Figure 3.8: Sensitivity experiments on the KITTI labelled benchmark (see text for discussion).

the number of region proposals. MF3D is robust to the depth descriptor scaling factor α_d: we set $\alpha_d = 0.01$ and it suffices to be within an order of magnitude. The depth features are computed using $n_d = 32$ random depth comparisons. Increasing the number of comparisons up to 8-fold does not improve results consistently; at any rate, the depth descriptor has only a small effect on labelling accuracy. We generated 1000 region proposals for each database image, and increased this to $m = 4000$ for the test images. Increasing the number of region proposals at test time does improve the labelling accuracy but with diminishing returns.

Label transfer and discriminative classification. Finally, we completed an additional experiment to characterize the effective difference between using model-free label transfer and discriminative classification. In this experiment, we replaced the label transfer in our pipeline with the recent SegNet [7]. SegNet is a fully convolutional neural network architecture comprising an encoder network, a corresponding decoder network, and a pixelwise classification layer. The network is designed for 2D scene parsing, and targets road scene understanding in particular. The encoder network follows the VGG-16 network architecture [117]. The decoder performs upsampling based on the max-pooling indices computed during encoding, and further convolves with learned filters.

We used the public implementation of SegNet built on the Caffe deep learning framework [61], and fine-tuned it on the KITTI dataset with pre-training on ImageNet [111]. To fine-tune the network we used stochastic gradient descent with a learning rate of 0.01 and momentum of 0.9. Differences in class frequency were handled by median frequency balancing [7]. Holding the rest of the 3D reconstruction pipeline fixed, replacing label transfer with SegNet reduces the mean per-class accuracy on KITTI to 81.8% (original: 83.5%). While SegNet is effective in predicting the single most likely class per pixel, our label transfer better accommodates noisy predictions as it makes use of redundancy via overlapping regions. These independently matched overlapping regions provide multiple sources of evidence for the classes at a given pixel.

Chapter 4

Scaling Nonparametric Scene Parsing: Methods for Large-Scale Search

In the previous chapter, we presented a nonparametric approach to 3D scene parsing that differs from conventional practice in its use of search-based label transfer instead of discriminative classification. MF3D achieves competitive labelling accuracy and is easily extensible to new categories as no model training is required. However, as described so far, MF3D performs an exact nearest neighbor search during label transfer, which is computationally expensive in terms of both time and memory. We address this point next, and show how unsupervised binary encoding (or hashing) can be easily incorporated into our nonparametric scene parsing framework for scalability to larger databases.

In this chapter, we propose two novel binary encoding algorithms for large-scale approximate nearest neighbor search. The first algorithm is data-independent and easily scalable to new examples and categories. The second algorithm is supervised and targets high-dimensional feature vectors such as the proposal descriptors in our scene parsing framework. We evaluate these methods on standard retrieval benchmarks, and then demonstrate their effectiveness in the context of 3D scene parsing on the KITTI labelled dataset.

4.1 Related work

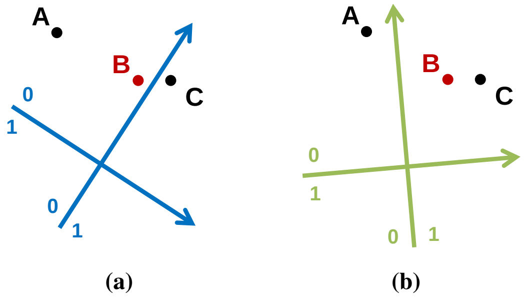
One class of methods for performing large-scale approximate nearest neighbor search is *binary encoding*, also referred to as *hashing* (distinct from voxel hashing in Section 3.2.1). Binary encoding methods learn a similarity-preserving transformation of the feature vectors from their original, real-valued, high-dimensional feature space to a low-dimensional binary-valued space. By transforming feature vectors to compact binary codes, significant gains in memory and time efficiency can be achieved. The Hamming distance between two binary codes can additionally be computed by taking an XOR and counting the ones, which is directly supported in modern hardware.

A broad range of binary encoding algorithms has been developed for large-scale search, with varying tradeoffs in speed, accuracy, and underlying assumptions. Locality-sensitive hashing (LSH) [18, 39] is a traditional, data-independent binary encoding method that uses randomly generated projections. In LSH, the probability of two feature vectors hashing to the same value is proportional to some measure of similarity between the vectors, such as their cosine similarity [18]. LSH offers theoretical asymptotic guarantees and can be kernelized [62, 75, 104], however a large number of bits (i.e. a large random projection matrix) is typically required for good performance in practice.

This performance limitation motivated the development of binary encoding methods that rely on training or optimization to learn the transformation. For example, binary reconstructive embedding [74] uses coordinate descent to minimize the error between the original pairwise distances and the Hamming pairwise distances. Spectral hashing [141] solves a relaxed version of a balanced graph partitioning problem to learn efficient codes. Iterative quantization (ITQ) [43] learns a rotation of PCA (unsupervised) or CCA (supervised) projected data that minimizes the quantization error, or distortion. The rotation is determined iteratively by alternating between updating the assignments given the current rotation, and updating the rotation to minimize the quantization error given the current assignments. Minimal loss hashing [95] applies structured prediction techniques to minimize a hinge-like error. Kernel-based supervised hashing [85] sequentially learns a projection in kernel space. K-means hashing [49] clusters the data such that the Euclidean distance between two data points is closely approximated by the Hamming distance between their cluster indices. Graph cuts coding [36] integrates the binary codes into the optimization as auxiliary variables. Supervised discrete hashing [114] learns a projection that optimizes the linear classification performance of the binary codes. The authors formulate a discrete optimization problem that is solved iteratively by alternating among three sub-problems: optimizing the hash functions, the linear classifier, and the binary code assignments, given the other two. We compare to several of these methods in the experiments (Section 4.2.2 and Section 4.3.2).

Gong et al. [42] showed that binary encoding methods are often infeasible for searching high-dimensional feature vectors such as VLAD [59] and Fisher Vectors [102]. Their bilinear projections method replaces the usual expensive projection matrix with two smaller bilinear projections. Circulant binary embedding [147] learns a projection by a circulant matrix, which can be accelerated using the Fast Fourier Transform. Xia et al. [143] proposed the unsupervised sparse projection (SP) method for searching high-dimensional feature vectors. The sparse projection is obtained by an iterative, ITQ-like optimization with an additional L0 regularization constraint. SP achieves an order of magnitude faster encoding than unsupervised dense methods while offering comparable accuracy. In addition, SP produces much more accurate results than bilinear projections or circulant binary embedding, while having similar encoding speed. Concurrent with SP, Rastegari et al. [105] proposed sparse binary embedding, which finds a sparse projection matrix via a different SVM-based optimization. Kronecker binary embedding [149] computes fast projections using a structured matrix that is the Kronecker product of small orthogonal matrices. Like these methods, our proposed supervised sparse projections method (Section 4.3) targets high-dimensional feature vectors.

Besides binary encoding, approximate nearest neighbor search can also be approached via vector quantization, which encodes data points using learned codebooks that can be combined in product [35, 53, 60, 96] or additive [5, 6] forms. Vector quantization algorithms tend to achieve high retrieval accuracy but are slower than hashing algorithms. Binary codes can also be used directly in downstream clustering or classification modules, which is not possible with vector quantization algorithms [43].



(a) (b)

Figure 4.1: Toy example illustrating neighbor retrieval under two different sets of hashing hyperplanes. The hyperplanes in (a) map A and B to the same binary code, and C to a different binary code. The hyperplanes in (b) map B and C to the same code, and A to a different code.

4.2 Bank of random rotations

Most modern binary encoding and vector quantization algorithms require training, whether in the form of learning optimal transformations or codebooks. We propose a data-independent binary encoding method that, like traditional LSH, requires no training and is therefore well-suited for reducing the time and memory requirements of nonparametric scene parsing.

4.2.1 Algorithm description

Graphically, we can think of the bits in a binary code as the decisions of a set of hash functions or hyperplanes in the feature space. A wide variety of techniques have been developed to find the best set of hyperplanes (Section 4.1), which may depend on the application, or more generally, the loss function to be minimized. For example, many binary encoding methods optimize for a transformation that minimizes the overall quantization error, or distortion [43, 49, 143, 147].

We can observe that, for many individual data points, the globally optimal transformation may not yield good retrieval performance. That is, neighbor relationships in the original feature space may not be well preserved. Figure 4.1 illustrates with a toy example in a two-dimensional feature space. Figure 4.1(a) shows two hyperplanes determining two bits of the binary code. This transformation maps A and B to the code 00 and C to the code 01. From the perspective of B, this transformation does not preserve its neighbor relationship: B's true neighbor, C, is mapped to a different code and appears farther in Hamming space than A, which is mapped to the same code (i.e., the Hamming distance between the binary codes of A and B is zero). Consider the alternative transformation in Figure 4.1(b). From B's perspective, this is a better transformation because it maps B and C to the same binary code and A to a different code.

We ask whether a bank of multiple transformations can be more effective than a single globally optimal transformation. For example, instead of learning a transformation to minimize the overall quantization error, we consider having each data point *select* from a bank of transformations the one that minimizes its own quantization error. This approach removes the need for global optimization or training, and as we will demonstrate later, allows for improved search accuracy at the cost of extra encoding overhead at query time.

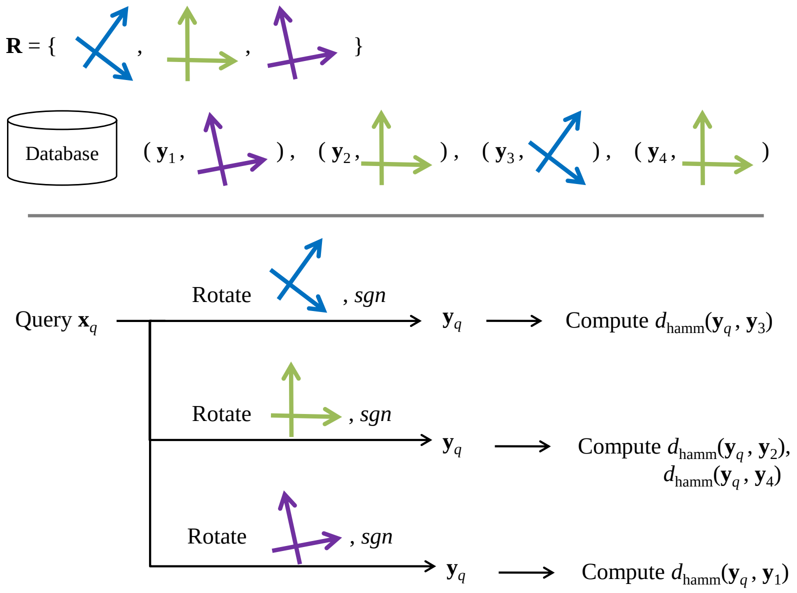
Following binary encoding and vector quantization methods that search for a globally optimal rotation of the data [35, 43, 96], our method generates a bank of *random* rotations $\mathcal{R} = \{\mathbf{R}_1, \mathbf{R}_2, ..., \mathbf{R}_K\}$. The database is encoded as follows. Given a database of n points, we first preprocess the database points by zero-centering and PCA-embedding. Next, for each individual data point $\mathbf{x}$, we select $\mathbf{R}^* \in \mathcal{R}$ that satisfies

$$\mathbf{R}^* = \arg\max_{\mathbf{R} \in \mathcal{R}} \|\mathbf{x}\mathbf{R}\|_1 \tag{4.1}$$

where $||\cdot||$ is the L1 norm. Intuitively, $\mathbf{R}^*$ maps $\mathbf{x}$ far from the hyperplanes to reduce the quantization error (e.g. the right transformation in the toy example of Figure 4.1). We apply the rotation to $\mathbf{x}$ and obtain the binary code $\mathbf{y}$ by the usual thresholding at zero:

$$\mathbf{y} = \text{sgn}(\mathbf{x}\mathbf{R}^*) \tag{4.2}$$

Finally, we store the pair $(\mathbf{y}, j)$, where j is the index of $\mathbf{R}^*$ in the bank of random rotations $\mathcal{R}$. That is, j identifies the rotation used to produce the binary code $\mathbf{y}$. We allocate bits from our budget to index the best rotation (i.e. to store j)



R = { , , }
Database
($\mathbf{y}_1$,), ($\mathbf{y}_2$,), ($\mathbf{y}_3$,), ($\mathbf{y}_4$,)
Query $\mathbf{x}_q$
Rotate , sgn
$\mathbf{y}_q$
Compute $d_{\text{hamm}}(\mathbf{y}_q, \mathbf{y}_3)$
Rotate , sgn
$\mathbf{y}_q$
Compute $d_{\text{hamm}}(\mathbf{y}_q, \mathbf{y}_2)$, $d_{\text{hamm}}(\mathbf{y}_q, \mathbf{y}_4)$
Rotate , sgn
$\mathbf{y}_q$
Compute $d_{\text{hamm}}(\mathbf{y}_q, \mathbf{y}_1)$

Figure 4.2: Toy example illustrating a query with a bank of random rotations

for each individual data point, and therefore incur no additional memory usage for the database. There is, however, a small memory overhead for storing the bank.

Next, we explain how online queries work in the bank of random rotations framework. Given a (zero-centered, PCA-embedded) query point $\mathbf{x}_q$, we compute distances to the database points adaptively, using each database point's selected rotation. More specifically, for each rotation $\mathbf{R}$ in the bank, we quantize $\mathbf{x}_q$ with respect to the rotation $\mathbf{R}$ via $\mathbf{y}_q = \text{sgn}(\mathbf{x}_q\mathbf{R})$, and compute the Hamming distance of $\mathbf{y}_q$ to all database binary codes that selected $\mathbf{R}$ as their rotation. Figure 4.2 illustrates with a toy example. For greater clarity, a single query performs K rotations and then computes only n Hamming distances – not Kn distances. The number of Hamming-space comparisons is unchanged by the bank of random rotations.

Large-scale applications often involve databases that grow over time as users continually add images or video. In the proposed bank of random rotations approach, when a new point is added to the database it selects a rotation from the bank independently of all other database points. In contrast to training-based methods, no global optimization problem over the database needs to be re-solved. The proposed approach therefore offers a particularly scalable solution in large-scale scene parsing applications where the database is expected to grow over time.

4.2.2 Experiments

We evaluated the retrieval performance of the bank of random rotations on three standard unsupervised retrieval benchmarks: SIFT1M, GIST1M [60], and CIFAR-10 [71]. The bank of random rotations is compared with three binary encoding baselines: spectral hashing [141], iterative quantization [43], and k-means hashing [49].

Benchmark details. The SIFT1M dataset [60] consists of 1 million SIFT descriptors for the database and 10,000 independent descriptors for queries. SIFT1M was used by He et al. [49] in the experimental evaluation of k-means hashing. The GIST1M dataset [60] consists of 1 million, 960-dimensional Gist descriptors [97] for the database and 1,000 descriptors for queries. The CIFAR-10 dataset [71] is a 60,000-image, 10-class subset of the Tiny Images dataset [129]. We computed 384-dimensional Gist descriptors for each CIFAR-10 image. Gong et al. [43] used a similar 11-class CIFAR dataset in the experimental evaluation of iterative quantization. We followed their methodology of randomly generating five training-testing splits, each containing 1,000 test queries, and averaging over the five trials.

Implementation details. We used publicly available implementations of spectral hashing, iterative quantization, and k-means hashing. K-means hashing requires a product-space decomposition [60] to be tractable. The split into product subspaces is performed following the split used by He et al. [49] in their experiments: 4 bits per subspace for 64-bit and 128-bit codes from SIFT descriptors, and 8 bits per subspace for Gist descriptors. Recall that, for each database point, we allocate bits from our budget to store the identifier (index) of the rotation selected from the bank. Experiments on all three datasets use a bank of 256 random rotations. For a fair comparison, we adjusted the number of bits in our binary codes accordingly. For example, for 64-bit codes and a bank of 256 random rotations, we computed 56-bit codes since 8 bits are needed to index into the bank.

Evaluation methodology. Retrieval performance is illustrated using recall@N curves [35, 49, 60, 96], which plot the proportion of true neighbors retrieved in

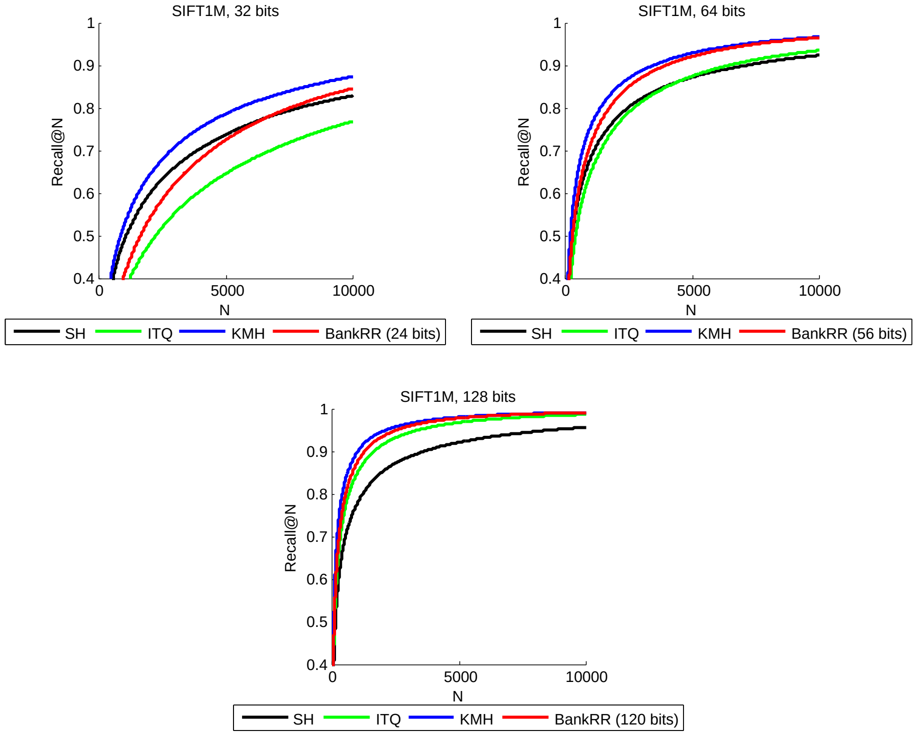


Figure 4.3: Retrieval performance on SIFT1M unsupervised retrieval benchmark with 32-bit, 64-bit, and 128-bit binary codes (SH: Spectral hashing, ITQ: Iterative quantization, KMH: K-means hashing, BankRR: Bank of random rotations).

the first N Hamming neighbors. Following the convention of He et al. [49], the ground truth is considered to be the query's 10 nearest Euclidean neighbors in the original feature space.

Results. Figures 4.3, 4.4, and 4.5 show the retrieval performance of 32-bit, 64-bit, and 128-bit codes on the SIFT1M, GIST1M, and CIFAR-10 benchmarks. Despite not requiring any training or complex sequential optimization, the data-independent bank of random rotations achieves state-of-the-art results on CIFAR-10 and GIST1M. On SIFT1M, results at 64 and 128 bits are on par with k-means hashing, which has been demonstrated to outperform earlier methods including locality sensitive hashing, PCA hashing, spectral hashing, and minimal loss hashing

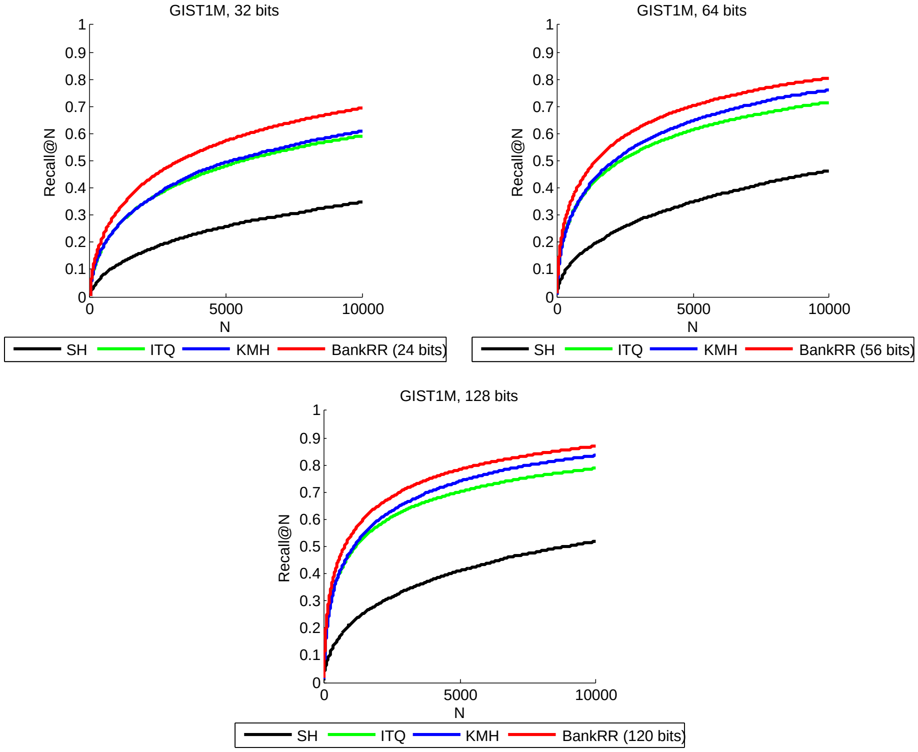


Figure 4.4: Retrieval performance on GIST1M unsupervised retrieval benchmark with 32-bit, 64-bit, and 128-bit binary codes (SH: Spectral hashing, ITQ: Iterative quantization, KMH: K-means hashing, BankRR: Bank of random rotations).

[49]. The bank of random rotations consistently achieves higher retrieval accuracy than the natural baseline, iterative quantization, which uses a single globally optimal rotation.

Figure 4.6 shows qualitative results on CIFAR-10. The bank of random rotations allows for the efficient retrieval of similar neighbors at a fraction of the memory requirements. Irrelevant images are sometimes retrieved due to the approximation error of binary encoding, especially at low code lengths. However, even the full-dimensional Gist descriptors sometimes retrieve incorrect images. This inexactness is expected because image descriptors only coarsely characterize the image appearance, and visually similar instances from different semantic

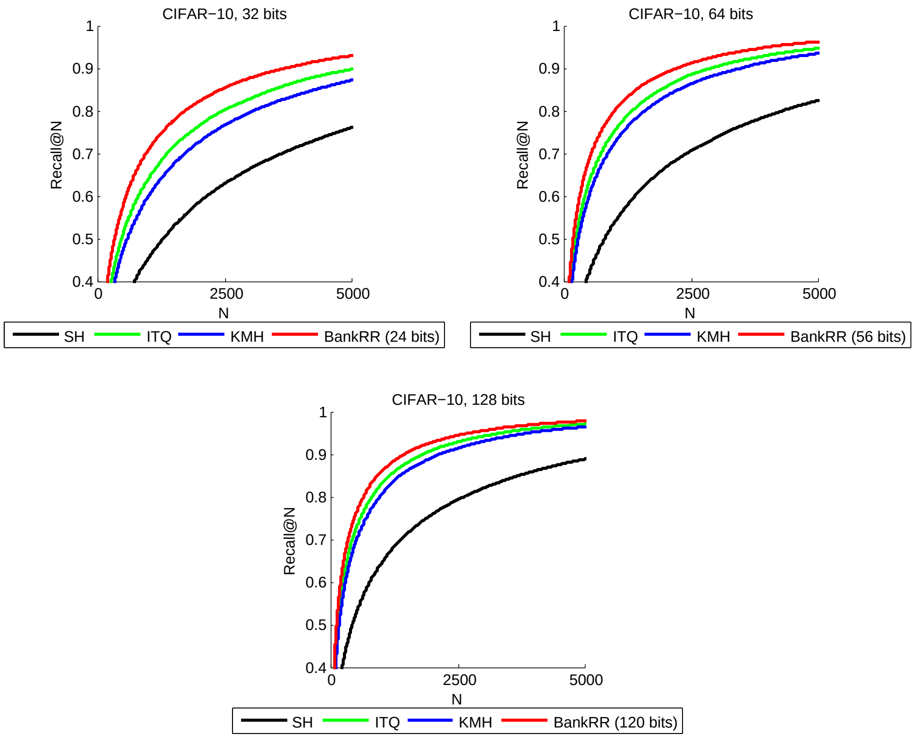


Figure 4.5: Retrieval performance on CIFAR-10 unsupervised retrieval benchmark with 32-bit, 64-bit, and 128-bit binary codes (SH: Spectral hashing, ITQ: Iterative quantization, KMH: K-means hashing, BankRR: Bank of random rotations).

categories may be mapped to similar descriptors. Replacing a 384-dimensional Gist descriptor with a 128-bit binary code represents a reduction in memory by two orders of magnitude (1,536 or 3,072 bytes to 16 bytes).

The bank of random rotations approach provides efficient unsupervised large-scale search without training, making it well suited to our nonparametric scene parsing framework, as both are designed to be easily extensible to an online growth in the database. In scene parsing we are interested in matching region descriptors that have associated semantic labels. A natural question to ask is whether we can achieve even better performance if we leverage this supervisory information. We next describe a novel supervised binary encoding method that is particularly ap-

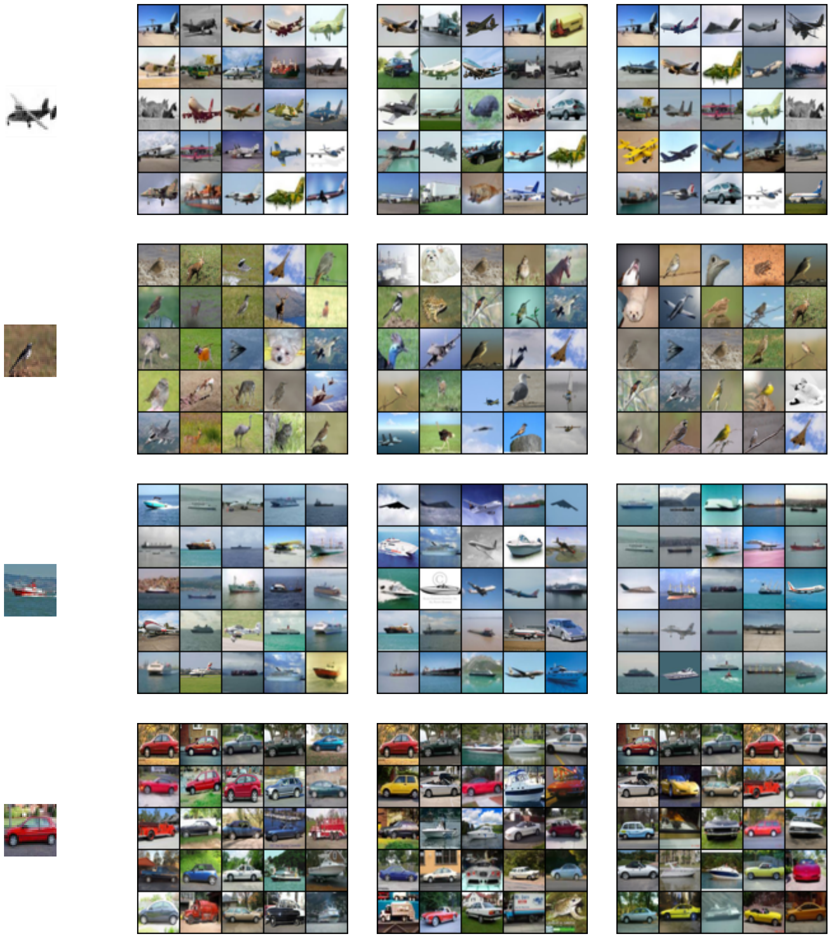


Figure 4.6: Qualitative retrieval results of the bank of random rotations on the CIFAR-10 dataset. From left to right: query image; feature-space (Gist) neighbors; 32-bit code neighbors; 128-bit code neighbors.

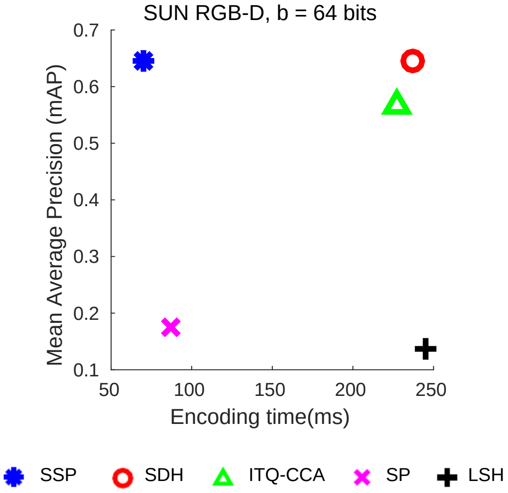


Figure 4.7: Sample results. Retrieval accuracy versus encoding time on the SUN RGB-D benchmark [122] using 4096-dimensional convolutional neural network features. Binary encoding methods: supervised discrete hashing (SDH) [114]; iterative quantization with canonical correlation analysis (ITQ-CCA) [43]; sparse projections (SP) [143]; locality-sensitive hashing (LSH) [18, 39]. See Section 4.3.2 for complete results.

plicable to searching high-dimensional feature vectors, such as today's ubiquitous deep neural network features.

4.3 Supervised sparse projections

With the growth of big data, computer vision algorithms are relying on increasingly complex models and producing increasingly high dimensional features. From second-order pooling of traditional hand-crafted features [17], to Fisher Vectors [102], to the activations of deep neural networks [19], features with thousands or tens of thousands of dimensions have become the norm. However, high-dimensional features pose particular challenges for binary encoding algorithms. For example, traditional algorithms lack an effective regularizer for the learned projections, which sometimes leads to overfitting in high dimensions [143]. Moreover, encoding in high dimensions is computationally expensive due to the large projection matrix [42, 143].

In this section, we present a novel binary encoding method called Supervised Sparse Projections (SSP), in which the conventional dense projection matrix is replaced with a discriminatively-trained sparse projection matrix. This design decision enables SSP to achieve two to three times faster encoding in comparison to conventional dense methods, while obtaining comparable retrieval accuracy. SSP is also more accurate than unsupervised high-dimensional binary encoding methods while providing similar encoding speeds. Fig. 4.7 shows sample experimental results using 4096-dimensional convolutional neural network features.

4.3.1 Algorithm description

Let $\mathbf{X} \in \mathbf{R}^{n \times d}$ denote a database containing n feature vectors, with each feature vector being d-dimensional. The goal of a binary encoding algorithm is to learn a projection matrix $\mathbf{P} \in \mathbf{R}^{d \times b}$ that can be applied to $\mathbf{X}$ to produce similarity-preserving binary codes $\mathbf{B} \in \{-1, 1\}^{n \times b}$, where b is the number of bits in the binary codes. In the case of supervised binary encoding, we also have semantic class labels $\mathbf{Y} \in \mathbf{R}^{n \times C}$, where C is the number of classes, and $y_{ki} = 1$ if $\mathbf{x}_i$ belongs to class k and 0 otherwise.

Given a novel query $\mathbf{x}_q \in \mathbf{R}^d$, the learned projection $\mathbf{P}$ is applied to obtain the corresponding binary code $\mathbf{b} \in \{-1, 1\}^b$:

$$\mathbf{b}_q = \text{sgn}(\mathbf{x}_q \mathbf{P}) \tag{4.3}$$

where sgn$(\cdot)$ denotes the sign function applied to a vector, which returns +1 for positive values and -1 otherwise. It is also possible to learn and apply the projection to a non-linear mapping of $\mathbf{x}_q$: that is, $\mathbf{b}_q = \text{sgn}(\phi(\mathbf{x}_q)\mathbf{P})$ where ϕ is a non-linear mapping such as an RBF kernel mapping. For ease of formulation and experimentation we develop the simple linear case in Eq. 4.3.

What makes for an effective projection matrix $\mathbf{P}$? A good projection matrix should preserve semantic relationships, as revealed in the supervisory signal $\mathbf{Y}$. It should be generalizable to new, previously unseen queries. Finally, we argue that it should be fast to apply at test time – an important criterion as today's increasingly complex models produce increasingly high-dimensional features.

To obtain an effective projection matrix, the Supervised Sparse Projections

(SSP) method finds an approximate solution to the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{B},\mathbf{W},\mathbf{P}} ||\mathbf{Y} - \mathbf{B}\mathbf{W}||_F^2 + \lambda||\mathbf{W}||_F^2 \\ &\text{s.t. } \mathbf{B} = \text{sgn}(\mathbf{X}\mathbf{P}),\ ||\mathbf{P}||_1 \leq \tau \end{aligned} \tag{4.4}$$

where $\mathbf{W} \in \mathbf{R}^{b \times C}$, $||\cdot||_F$ denotes the Frobenius norm, λ is a regularization parameter, and τ is a sparsity parameter. The objective (4.4) can be seen as a standard L2 classification loss fitting the binary codes $\mathbf{B}$ to the supervisory signal $\mathbf{Y}$, with an additional constraint that $\mathbf{B}$ is generated by applying a sparse projection $\mathbf{P}$ to the feature vectors $\mathbf{X}$. We solve this optimization problem approximately following a sequential approach. In the first subproblem, we find $\hat{\mathbf{B}}$ and $\hat{\mathbf{W}}$ satisfying

$$\begin{aligned} &\min_{\mathbf{B},\mathbf{W}} ||\mathbf{Y} - \mathbf{B}\mathbf{W}||_F^2 + \lambda||\mathbf{W}||_F^2 \\ &\text{s.t. } \mathbf{B} \in \{-1, 1\}^{n \times b} \end{aligned} \tag{4.5}$$

In the second subproblem, given $\hat{\mathbf{B}}$ from the first subproblem, we solve for the projection matrix $\mathbf{P}$ satisfying

$$\begin{aligned} &\min_{\mathbf{P}} ||\hat{\mathbf{B}} - \mathbf{X}\mathbf{P}||_F^2 \\ &\text{s.t. } ||\mathbf{P}||_1 \leq \tau \end{aligned} \tag{4.6}$$

An alternative approach would be to incorporate the constraint $\mathbf{B} = \text{sgn}(\mathbf{X}\mathbf{P})$ as a penalty term in a single objective,

$$\begin{aligned} &\min_{\mathbf{B},\mathbf{W},\mathbf{P}} ||\mathbf{Y} - \mathbf{B}\mathbf{W}||_F^2 + \lambda||\mathbf{W}||_F^2 + \nu||\mathbf{B} - \mathbf{X}\mathbf{P}||_F^2 \\ &\text{s.t. } \mathbf{B} \in \{-1, 1\}^{n \times b},\ ||\mathbf{P}||_1 \leq \tau, \end{aligned}$$

and then iterate the following until convergence, similar to [114]: (i) fix $\mathbf{B}$, $\mathbf{P}$, and update $\mathbf{W}$, (ii) fix $\mathbf{W}$, $\mathbf{P}$, and update $\mathbf{B}$, (iii) fix $\mathbf{W}$, $\mathbf{B}$, and update $\mathbf{P}$. However, the sequential approach is much faster to optimize and produces similar results in practice. Concretely, on the SUN RGB-D benchmark with b = 32 bits, the sequential approach is over 800% faster than iterating (i),(ii),(iii), with less than 0.1% difference in mAP (averaged over ten trials). Training time is dominated by the subproblem of solving for $\mathbf{P}$ given the sparsity constraint and fixed $\mathbf{W}$ and $\mathbf{B}$. We therefore optimize the original objective (4.4) by solving subproblems (4.5) and (4.6) sequentially.

Our formulation is similar to that of [114], with two main differences: our method computes a sparse projection, which allows for two to three times faster encoding with comparable accuracy; and our optimization procedure is different.

Solving for $\hat{\mathbf{B}}$ and $\hat{\mathbf{W}}$

We solve for $\hat{\mathbf{B}}$ and $\hat{\mathbf{W}}$ in subproblem (4.5) via an iterative alternating optimization. First, we keep $\mathbf{B}$ fixed and update $\mathbf{W}$; then, we keep $\mathbf{W}$ fixed and update $\mathbf{B}$. The update of $\mathbf{W}$ is optimal given fixed $\mathbf{B}$, while the update of $\mathbf{B}$ given $\mathbf{W}$ is an approximation. These steps are repeated until convergence or a maximum number of iterations is reached. The alternating minimization method finds a local optimum, and so will be sensitive to initialization.

(i) Fix $\mathbf{B}$*, update* $\mathbf{W}$.

With $\mathbf{B}$ fixed in subproblem (4.5), the problem of solving for $\mathbf{W}$ becomes standard L2-regularized least squares regression (ridge regression), which admits a closed form solution. In particular, we update $\mathbf{W}$ as

$$\mathbf{W} = (\mathbf{B}^\top \mathbf{B} + \lambda \mathbf{I})^{-1} \mathbf{B}^\top \mathbf{Y} \tag{4.7}$$

(ii) Fix $\mathbf{W}$*, update* $\mathbf{B}$.

With $\mathbf{W}$ fixed in subproblem (4.5), the resulting problem for $\mathbf{B}$ admits an iterative solution via the discrete cyclic coordinate descent (DCC) method [114]. DCC allows us to solve for $\mathbf{B}$ bit by bit – in other words, one column of $\mathbf{B}$ at a time. Each column is solved in closed form holding the other columns fixed. In particular, with the other columns fixed, the kth bit (column) of $\mathbf{B}$ has the optimal

Algorithm 1 Learning Supervised Sparse Projections (SSP)

```
Input: Database feature vectors $\mathbf{X} \in \mathbf{R}^{n \times d}$; Supervisory labels $\mathbf{Y} \in \mathbf{R}^{n \times C}$; $\lambda$, $\tau$
Output: Projection matrix $\mathbf{P} \in \mathbf{R}^{d \times b}$; Binary codes $\mathbf{B} \in \{-1, 1\}^{n \times b}$
  Initialize $\mathbf{P}$ as a random projection and $\mathbf{B}$ as sgn($\mathbf{XP}$).
  repeat
    Update $\mathbf{W}$ by Eq. 4.7.
    Update $\mathbf{B}$ by discrete cyclic coordinate descent (Eq. 4.8).
  until converged or maximum number of iterations reached.
  Solve L1-regularized least squares (lasso) problem (Eq. 4.6) to obtain $\mathbf{P}$.
```

solution

$$\mathbf{b}_{:k} = \text{sgn}(\mathbf{q}_{:k} - \mathbf{B}'\mathbf{W}'\mathbf{w}_k^\top) \tag{4.8}$$

where $\mathbf{q}_{:k}$ is kth column of $\mathbf{Q} = \mathbf{YW}^\top$, $\mathbf{B}'$ is $\mathbf{B}$ excluding column k, $\mathbf{w}_k$ is the kth row of $\mathbf{W}$, and $\mathbf{W}'$ is $\mathbf{W}$ excluding row k. We invite the interested reader to refer to [114] for the complete derivation.

Solving for P

Given $\hat{\mathbf{B}}$ from the previous section, we next solve for $\mathbf{P}$ in subproblem (4.6). With $\hat{\mathbf{B}}$ fixed, we can recognize subproblem (4.6) as an L1-regularized least squares (lasso) problem. The L1 regularizer $||P||_1 \leq \tau$ induces sparsity in the solution $\mathbf{P}$, where the degree of sparsity is controlled by the user parameter τ.

L1-regularized least squares is a well studied problem and we apply one of the standard algorithms in the optimization community, SPGL1 [137], to solve for $\mathbf{P}$ iteratively. We use the publicly available implementation of SPGL1 with the default parameter settings.

The entire learning algorithm is summarized in Algorithm 1.

4.3.2 Experiments

We compared SSP with several representative binary encoding methods:

1. Locality-sensitive hashing (LSH) [18, 39] is an unsupervised binary encoding method that has been applied in a variety of computer vision tasks, in-

cluding image retrieval, feature matching, and object classification [24].

2. Iterative quantization [43] with canonical correlation analysis (ITQ-CCA) and the recently introduced supervised discrete hashing (SDH) method [114] are supervised binary encoding methods.

3. Sparse projections (SP) [143] is a state-of-the-art method for fast unsupervised high-dimensional binary encoding.

Our method can be extended to a non-linear embedding by RBF kernel in the same way as SDH [114]. For a fair comparison with ITQ-CCA, SP, and LSH, we evaluated all five methods without RBF embedding.

We performed experiments on three large vision datasets from different task domains:

1. The SUN RGB-D dataset [122] is a state-of-the-art benchmark for 3D scene understanding. It consists of 10,335 RGB-D images of indoor scenes and supports a wide range of scene understanding tasks, including scene categorization, semantic segmentation, object detection, and room layout estimation. We followed the recommended protocol for scene categorization, which uses the scene categories having more than 80 images (a total of 9,862 images over 21 categories). The SUN RGB-D benchmark provides convolutional neural network features as a baseline. These convolutional neural network features have $d = 4096$ dimensions and are generated using Places-CNN [150], a leading architecture for scene categorization on the SUN database [145]. We conducted ten trials, randomly setting aside 1,000 images as queries and using the remainder as the database in each trial.

2. The Animals with Attributes (AwA) dataset [79] was originally introduced for evaluating zero-shot learning. The task in zero-shot learning is to perform image classification for categories that are described in terms of their semantic attributes [29, 79], but which have no training examples. Here we use AwA simply for multi-class retrieval. The dataset consists of 30,475 images covering 50 animal categories, with a minimum of 92 images in each

category. AwA includes as a baseline 4096-dimensional convolutional neural network features generated using the VGG-19 “very deep” network [117]. We conducted ten trials, randomly setting aside 1,000 images as queries and using the remainder as the database in each trial.

3. The ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 2012 dataset [111] is a standard benchmark for image classification algorithms. It consists of 1.2 million images spanning 1,000 semantic categories. We extracted 4096-dimensional convolutional neural network features from each image using the VGG CNN-M network [19]. To hold the database in memory for training, we conducted ten trials in which 100 classes were randomly selected each time. Within the subset of 100 classes, we randomly set aside 1,000 images as queries and used the remainder as the database. We sampled 10,000 images from the database to train the binary encoding methods.

We measure retrieval accuracy and encoding speed for all methods. Retrieval accuracy is measured by the mean average precision (mAP). With labelled data, we are interested in preserving semantic similarity. The retrieval ground truth for computing the mean average precision consists of database examples sharing the same semantic category label as the query. We compute the encoding time as the processor time required for the matrix multiplication $\mathbf{b}_q = \text{sgn}(\mathbf{x}_q\mathbf{P})$ (Eq. 4.3), where $\mathbf{x}_q$ is the query vector and the projection matrix $\mathbf{P}$ may be sparse or dense, depending on the binary encoding method. In particular, SP and the proposed SSP produce sparse projections $\mathbf{P}$; LSH, ITQ-CCA, and SDH produce dense projections $\mathbf{P}$. Both sparse and dense matrix-vector multiplications are executed using the Intel Math Kernel Library (MKL). All timings are performed on a desktop computer with a 3.60 GHz CPU.

Fig. 4.8 summarizes the experimental results on SUN RGB-D for $b = 32$, 64, 96, and 128 bits per feature vector. The vertical axis shows the mean average precision; the horizontal axis shows the encoding time in milliseconds for 1,000 queries. Compared to the unsupervised high-dimensional binary encoding method, SP, the proposed SSP obtains substantially higher retrieval accuracy at similar encoding speed. For example, at $b = 64$ bits SSP achieves 64.6% mAP compared to 17.4% for SP. These results demonstrate the advantage of taking into consideration the

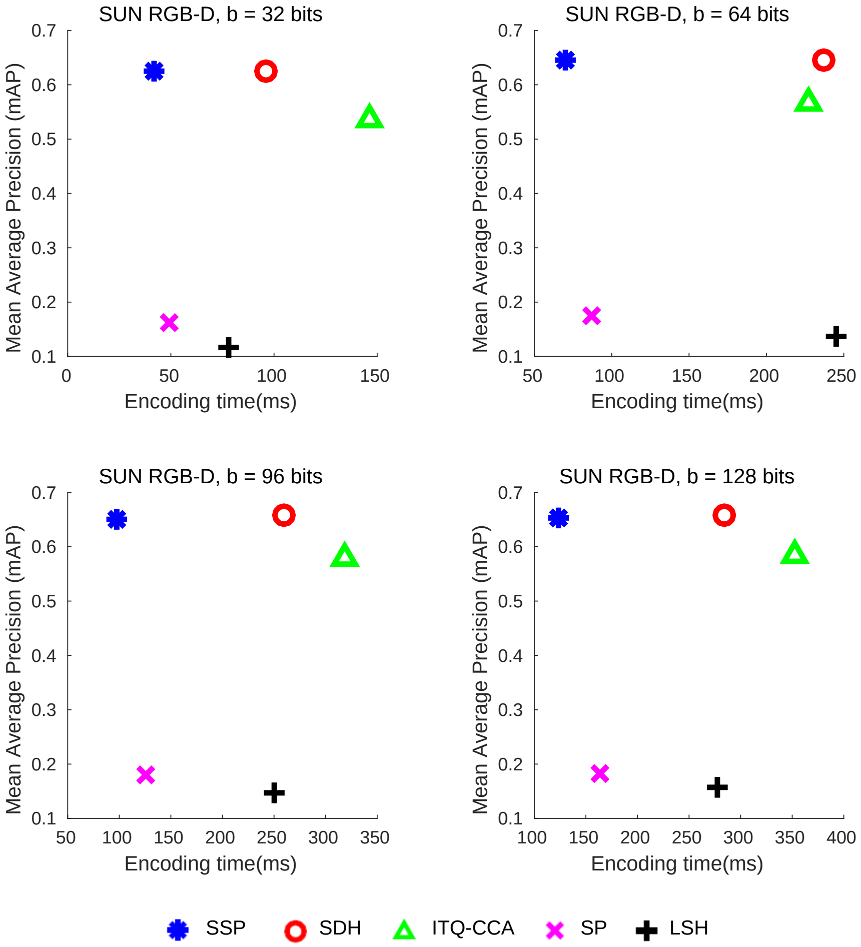


Figure 4.8: Experimental results on the SUN RGB-D dataset [122] using 4096-dimensional convolutional neural network features (Places-CNN [150]) and binary code lengths $b = 32$, 64, 96, and 128. Results are averaged over ten trials. Binary encoding methods: supervised sparse projections (SSP); supervised discrete hashing (SDH) [114]; iterative quantization with canonical correlation analysis (ITQ-CCA) [43]; sparse projections (SP) [143]; locality-sensitive hashing (LSH) [18, 39].

supervisory information available in many large-scale databases.

Because of the sparsity of the projection matrix, SSP provides significantly faster encoding speed than modern dense methods such as LSH, ITQ-CCA, and SDH. At the same time, retrieval accuracy is competitive with the dense methods. SSP achieves higher accuracy than ITQ-CCA and is comparable with SDH across all bit lengths. In particular, at b = 32, 64, 96, and 128 bits, SSP is 2.3 to 3.4 times faster than SDH to encode, while matching SDH's accuracy within 1% mAP. SSP is similarly faster than ITQ-CCA to encode, while obtaining 6 to 9% higher mAP.

In fact, these experimental results likely underestimate the practical improvement in encoding speed. The recent work on SP [143] reports encoding speeds that are linearly related to the proportion of nonzeros. For example, encoding with a projection matrix with 20% nonzeros is reportedly 5 times faster than encoding with a dense projection matrix. On the other hand, in Fig. 4.8, the sparse projection matrix for SSP has 20% nonzeros but we observed the encoding to be only 2.3 to 3.4 times faster than SDH, for instance. Since the authors of [143] also use MKL for timing evaluation[1], this difference may be due to optimizations not fully supported in our implementation. In summary, comparisons with LSH, ITQ-CCA, and SDH on SUN RGB-D demonstrate that L1 regularization can be used to sparsify the projection matrix and improve the encoding efficiency of high-dimensional feature vectors, with only a minor cost in retrieval accuracy.

We observed similar results on the AwA dataset. Fig. 4.9 shows the mean average precision versus encoding time for 1,000 queries, for b = 32, 64, 96, and 128 bits per feature vector. SSP achieves significantly faster encoding speed than the dense methods while providing comparable retrieval accuracy. For example, in comparison with the best performing dense method SDH, SSP is 2.2 to 3.2 times faster to encode and within 2% mAP across all bit lengths. SSP is significantly more accurate than the fast high-dimensional encoding method SP, with a relative improvement of 49 to 57% mAP across all bit lengths.

Fig. 4.10 summarizes the experimental results on ImageNet (ILSVRC 2012) for b = 32, 64, 96, and 128 bits per feature vector. Experimental results on ImageNet follow the general trends observed on the SUN RGB-D and AwA datasets.

[1] Personal communication.

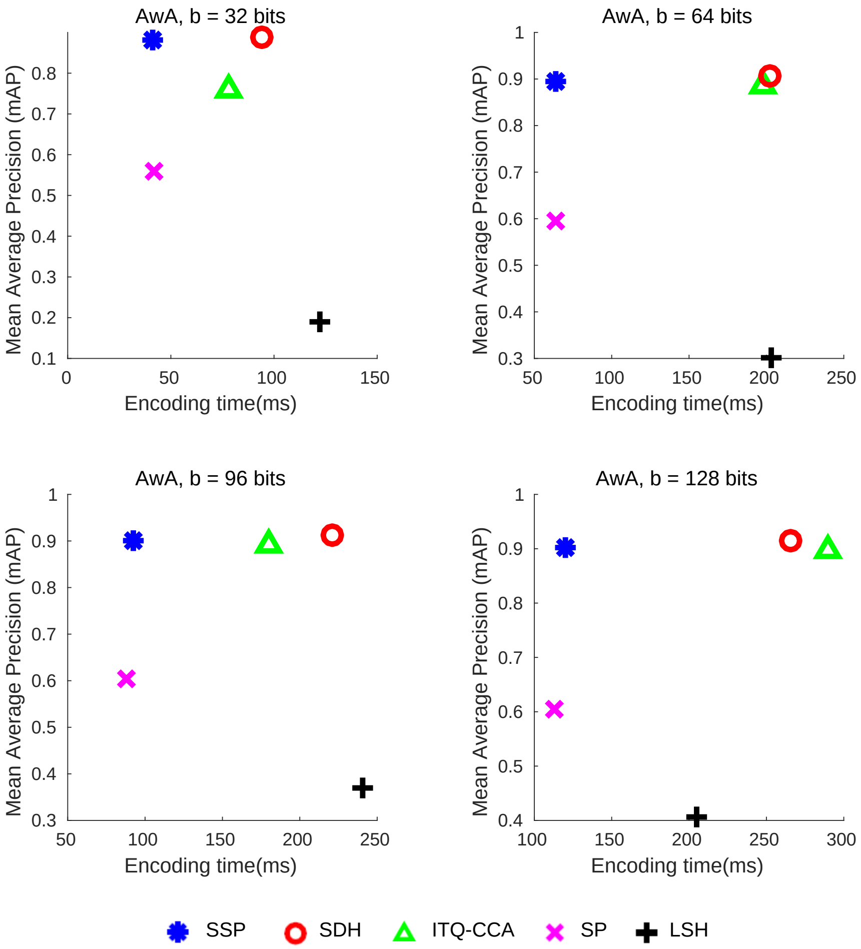


Figure 4.9: Experimental results on the AwA dataset [79] using 4096-dimensional convolutional neural network features (VGG-19 [117]) and binary code lengths $b = 32$, 64, 96, and 128. Results are averaged over ten trials. Binary encoding methods: supervised sparse projections (SSP); supervised discrete hashing (SDH) [114]; iterative quantization with canonical correlation analysis (ITQ-CCA) [43]; sparse projections (SP) [143]; locality-sensitive hashing (LSH) [18, 39].

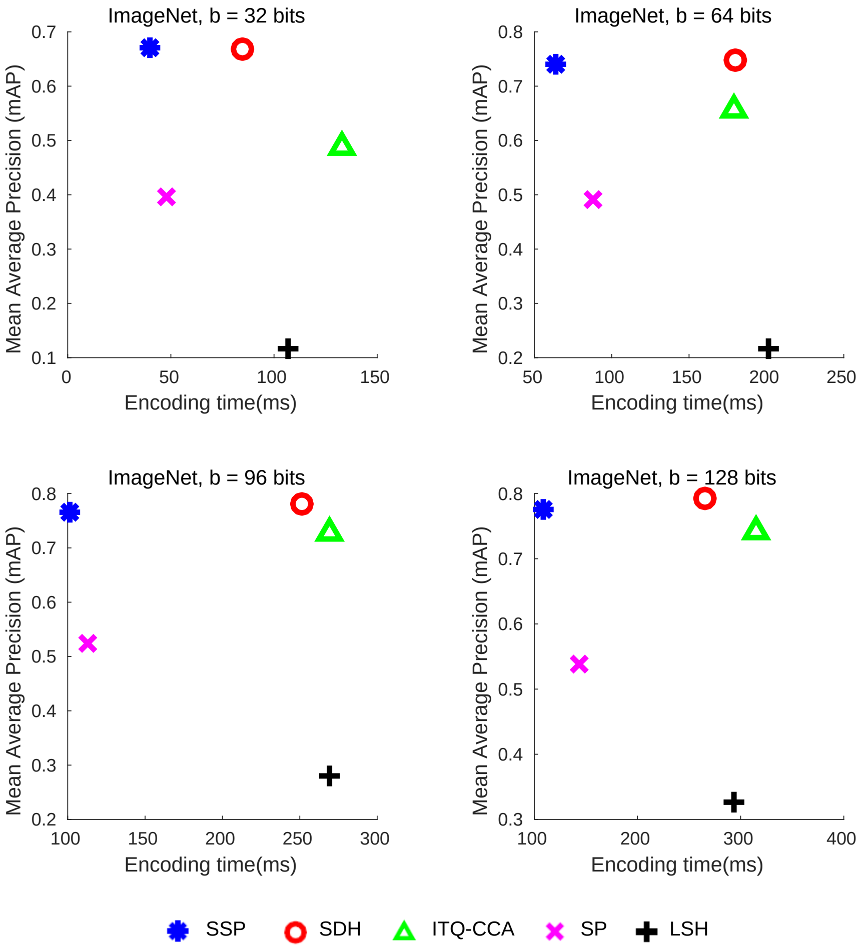


Figure 4.10: Experimental results on the ImageNet dataset [111] using 4096-dimensional convolutional neural network features (VGG CNN-M [19]) and binary code lengths b = 32, 64, 96, and 128. Results are averaged over ten trials. Binary encoding methods: supervised sparse projections (SSP); supervised discrete hashing (SDH) [114]; iterative quantization with canonical correlation analysis (ITQ-CCA) [43]; sparse projections (SP) [143]; locality-sensitive hashing (LSH) [18, 39].

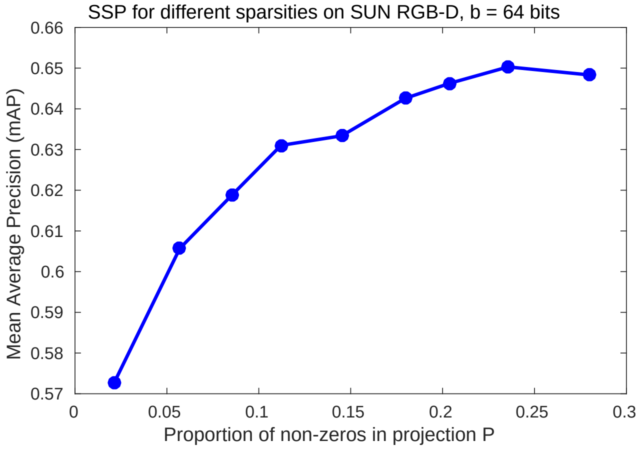


Figure 4.11: Retrieval accuracy versus percentage of non-zeros in the SSP projection matrix $\mathbf{P}$, for SUN RGB-D at $b = 64$ bits. The sparsity of $\mathbf{P}$ is determined by the parameter τ in Eq. 4.4. We set τ to obtain approximately 20% non-zeros in $\mathbf{P}$ in all other experiments.

Compared to fast high-dimensional binary encoding with SP, SSP obtains more accurate results at similar encoding speeds, with relative improvements of 43 to 69% mAP across all bit lengths. Compared to the best performing dense method SDH, SSP is 2.1 to 2.8 times faster to encode and is within 2% mAP across all bit lengths.

The sparsity of the learned projection matrix in SSP can be controlled by setting the τ parameter. In the preceding experiments we set τ to obtain approximately 20% non-zeros in the projection matrix $\mathbf{P}$. Fig. 4.11 shows how retrieval accuracy changes as we vary τ to obtain sparser or denser projection matrices. Experiments were conducted on the SUN RGB-D dataset with 64-bit codes, averaging over ten random trials. A tradeoff between projection sparsity and retrieval accuracy can be seen in the figure. In general, making the projection matrix more sparse enables faster encoding but at a cost in retrieval accuracy.

On the SUN RGB-D dataset, training requires 5.9 min for b = 32 bits; 10.3 min for b = 64 bits; 15.9 min for b = 96 bits; and 21.0 min for b = 128 bits, averaging over ten random trials.

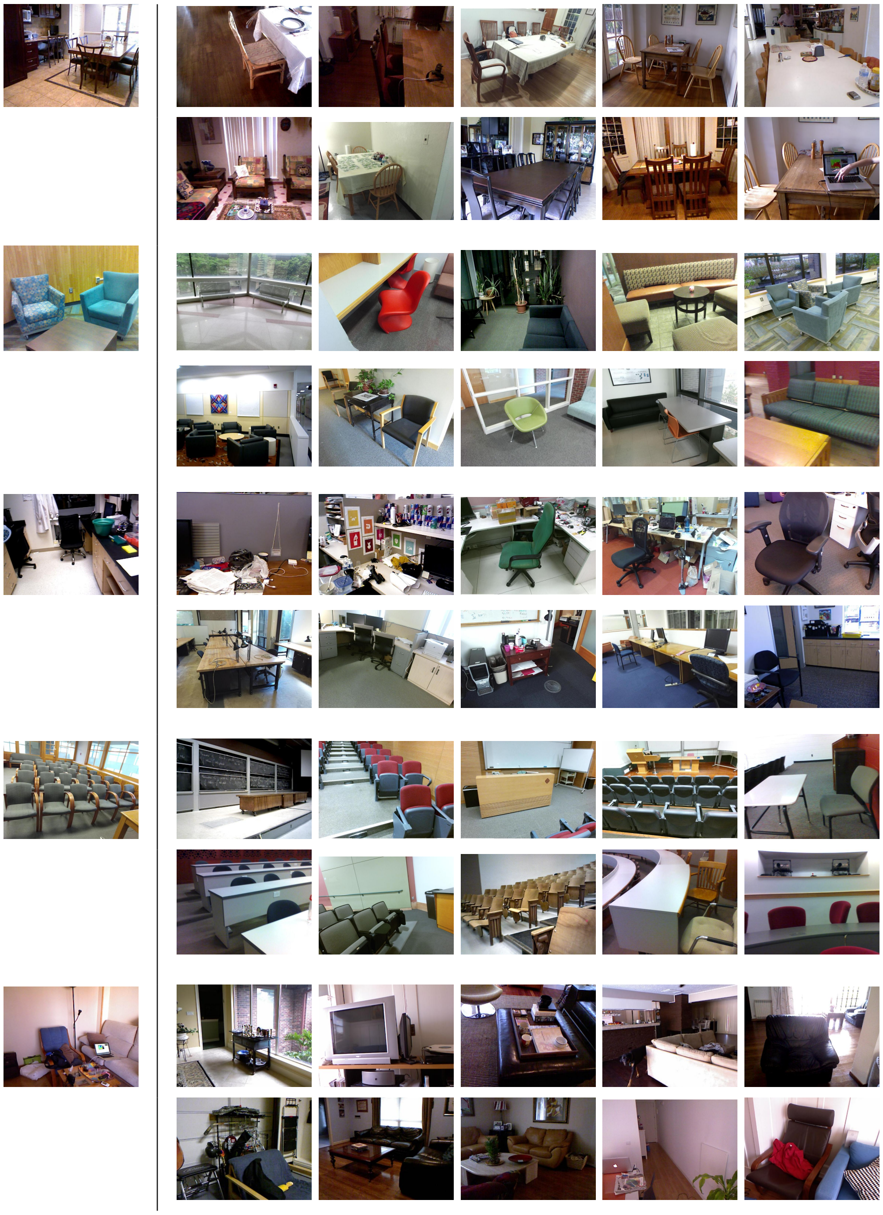


Figure 4.12: Qualitative results on the SUN RGB-D dataset at $b = 64$ bits. The left column shows query images. The right column shows ten of the top retrieved neighbors (sampled from multi-way ties) using SSP. The results indicate that SSP reliably preserves semantic neighbors in the learned binary codes.

Fig. 4.12 shows typical qualitative results on the SUN RGB-D dataset using SSP with $b = 64$ bits. SSP reliably preserves the supervisory information in the learned binary codes. We observed that even in failure cases, SSP returns semantically related neighbors. For example, the ground truth class label of the third query image from the top is 'lab', but the top retrieved nearest neighbors are 'office' images. In summary, SSP allows for scalable, fast, and semantics-preserving retrieval in large labelled databases. Replacing the original $d = 4096$ dimensional convolutional neural network features with 64-bit SSP binary codes results a reduction in storage costs by three orders of magnitude (131,072 or 262,144 bits reduced to 64 bits).

4.4 Integration with 3D scene parsing

Having presented two novel binary encoding methods with experimental validation on standard retrieval datasets, we next study the integration of these methods into our MF3D framework for nonparametric 3D scene parsing. We follow the same experimental protocol as described in Section 3.3.

Implementation details. The bank of random rotations method is unsupervised and can be applied directly on the region proposal descriptors without training. We generated four random rotations for the bank; an additional overhead of two bits is required to index the selected rotation per feature vector, and this is reflected in the slight difference in memory reduction reported later between the two binary encoding methods.

The supervised sparse projections method takes as input both the region proposal descriptors and a representation of their associated labels. Recall that each region proposal in the database has an associated ground-truth 2D label field L, appropriately cropped from the ground truth label images. We divide each proposal window into thirds horizontally and vertically to form a spatial grid. Within each grid cell, we compute an indicator vector showing the presence or absence of each class (i.e. the ith entry is 1 if class i is present, or 0 if class i is absent). The indicator vectors of the nine grid cells are concatenated to form a vector of length $9C$, where C is the number of classes, and the concatenated vectors are stacked to form SSP's supervisory matrix $\mathbf{Y} \in \mathbf{R}^{n \times 9C}$ (Section 4.3.1).

Table 4.1: MF3D with binary encoding on KITTI labelled benchmark using 256-bit binary codes

	Search time (s)	Mean per-class accuracy	Memory reduction
Baseline MF3D	24.4	83.5	–
MF3D + BankRR	1.34	82.6 ± 0.4	1,020×
MF3D + SSP	1.22	82.2 ± 0.7	1,028×

Quantitative results. Table 4.1 summarizes the time and memory efficiencies gained via binary encoding, using both the bank of random rotations and supervised sparse projections to generate 256-bit binary codes. Results are averaged over five random trials. Timing is performed in Matlab, single threaded execution on a 3.60 GHz CPU with hardware accelerated popcount for evaluating Hamming distances. The search time for supervised sparse projections is in fact a slight overestimate, as the sparse-dense matrix multiplication executes in Matlab no faster than the equivalent dense-dense matrix multiplication; however, the search time in both sparse and dense cases is dominated by the Hamming distance computation ($\sim$ 1.1s on average). The baseline MF3D searches the full-dimensional descriptors using a fast exact nearest neighbor search as implemented in the Yael library[2].

At 256 bits, binary encoding reduces the search time of the $m = 4000$ region proposals by an order of magnitude, while obtaining a mean per-class accuracy within 1% of the full dimensional descriptors. Replacing the full dimensional descriptors with 256-bit binary codes enables a reduction in memory costs of three orders of magnitude.

Fig. 4.13 shows the progression of labelling accuracy with respect to the number of bits in the binary code for the bank of random rotations, supervised sparse projections, and locality-sensitive hashing (LSH). The bank of random rotations and supervised sparse projections methods obtain comparable labelling results, with the bank of random rotations performing slightly better at longer binary code lengths (128 and 256 bits) and supervised sparse projections performing better at the shortest code length (32 bits). These results suggest that, for the specific

[2] http://yael.gforge.inria.fr/

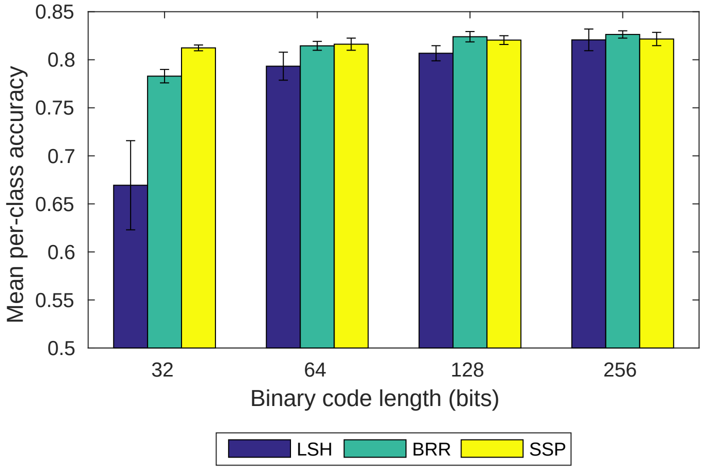


Figure 4.13: MF3D with binary encoding on KITTI labelled benchmark: mean per-class accuracy versus binary code length for locality-sensitive hashing (LSH), bank of random rotations (BRR), and supervised sparse projections (SSP).

purpose of label transfer in MF3D, the additional supervisory signal $\mathbf{Y}$ provides added benefit only at high levels of compression. As the number of bits in the binary code increases, preserving the similarity relationships in the region proposals feature space becomes sufficient for accurate label transfer; preserving $\mathbf{Y}$ becomes less important. At 64 bits and up, the bank of random rotations is preferred since its labelling outcomes are as good as the more complex supervised sparse projections approach, while not requiring training. Increasing the number of bits eventually produces diminishing returns, with all three binary encoding methods obtaining mean per-class accuracies within 0.5% by 256 bits.

Label transfer is highly parallelizable because each region proposal is matched and transferred independently. MF3D can therefore take advantage of multiple parallel GPU threads, each processing a separate region proposal. We leave this implementation for future work.

Qualitative results. Fig. 4.14 presents qualitative 3D scene parsing results at frames 3, 36, and 68 in the testing sequence (KITTI sequence 15), following Fig. 3.6. Results are generated online, using only the frames observed up to that time. In each block, the left column shows the RGB frame and the labelling result using full dimensional region descriptors. The top right visualizes the labelling result using the bank of random rotations, and the bottom right visualizes the labelling result using supervised sparse projections. 256-bit binary codes are generated for both methods. We observed that label transfer with the binary encoding methods produces labellings that are largely consistent with those produced by label transfer using full dimensional descriptors. Since binary encoding is an approximation, there are some minor but discernable discrepancies in the labelling results. For example, in frame 3, MF3D with the bank of random rotations mislabels the tree on the left side as building and pole; MF3D with supervised sparse projections does not capture the full extent of the pavement in frame 3; and neither binary encoding algorithm preserves the partial detection of the vegetation in the left foreground of frame 36. However, our qualitative results overall indicate that at 256 bits, binary encoding introduces a small degree of approximation error while obtaining a significant improvement in both time and space efficiency.

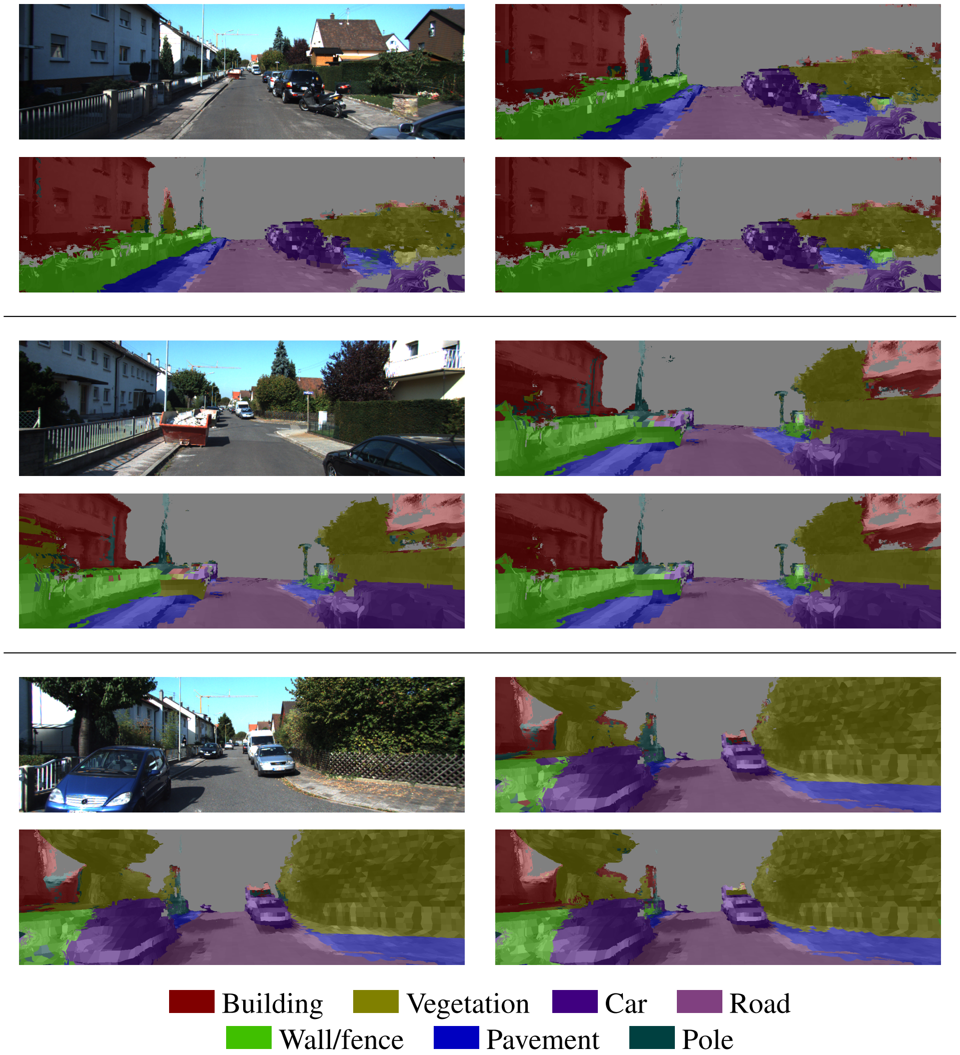


Figure 4.14: Qualitative 3D scene parsing results on the KITTI benchmark with binary encoding. Left column: RGB frame and labelling result of MF3D with full dimensional descriptors. Top right: Labelling result of MF3D with bank of random rotations. Bottom right: Labelling result of MF3D with supervised sparse projections. 256-bit binary codes are used and labelling results are to a maximum depth of 25 meters.

Chapter 5

Conclusion

Computer vision is about developing systems that can “see” and make sense of a complex visual world. Vision capabilities are being embedded in an increasingly broad range of computer systems, from smartphone apps, to self-driving vehicles, to bionics. As our capacity to bridge the semantic gap grows, we will demand more from our algorithms. It is no longer enough to detect that an image contains a car instead of a cat. In scene parsing, we aim to explain or reconstruct the entire scene, identifying all of the objects and background elements of interest.

“Big data” has also enabled us to build more sophisticated data-driven systems, with exciting advances in particular in deep neural networks for classification. These learning algorithms allow us to distinguish, with increasing accuracy, among pixels of a pre-defined set of semantic categories. In this thesis we have leveraged big data in a much less commonly explored way: for scene parsing by large-scale nonparametric search. Leveraging big data for search allows for powerful exemplar-based scene parsing that does not require training and can be easily extended to novel categories as the database grows. In Chapter 2, we presented a nonparametric algorithm for 2D scene parsing of images. In contrast to traditional nonparametric scene parsing methods that perform label transfer on the level of pixels or superpixels, CollageParsing integrates information from overlapping object proposals. In Chapter 3, we developed a nonparametric algorithm for 3D scene parsing of video. MF3D constructs a semantically labelled 3D reconstruction of the environment in an online manner, using only the frames observed so far. Our experimental results, both in 2D using images and in 3D using video, show that large-scale nonparametric search can be an effective strategy for scene parsing, producing accurate dense labellings while retaining extensibility to new categories and examples.

We argued that nonparametric search and parametric classification strategies are complementary in scene parsing, and we believe their effective combination is a promising direction for future work. Indeed, since the original publication of CollageParsing in [130], other authors have explored hybrid nonparametric-parametric approaches to 2D scene parsing. For example, Liu et al. [84] developed a hybrid approach for parsing humans in images, in which a convolutional neural network is trained to predict the confidence and displacements for matching labelled regions from neighbor database images to the query image. Shuai et al. [116] constructed a graphical model for 2D scene parsing in which unary potentials are computed locally using a trained convolutional neural network, and global potentials provide high-level context and are derived non-parametrically from similar exemplars.

Another promising direction for future work in scene parsing is the exploration of novel sources of relevant side information and unconventional ways to augment the labelled database of examples. Since obtaining fully labelled databases can be expensive, it is worth investigating how other sources of information can provide a useful prior over labellings. For instance, Wang et al. [140] recently leveraged a crowd-sourced mapping database called OpenStreetMaps to form a geographic prior for outdoor scene understanding. Given a geo-tagged image and camera parameters, the map is rendered in 3D to provide priors for geometry (depth) and semantic labels. Besides novel sources for priors, synthetic data augmentation may be helpful when real-world labelled data is scarce. Along this line of investigation, Gaidon et al. [34] recently introduced the Virtual KITTI dataset, consisting of synthetic clones of KITTI sequences photorealistically rendered using the Unity graphics engine. Their experiments indicate that virtual pre-training improves the performance of multi-object tracking in real video sequences.

Nonparametric scene parsing requires new techniques for searching large labelled collections of data that are time and memory efficient. In Chapter 4, we proposed two novel binary encoding algorithms for approximate nearest neighbor search and demonstrated how these algorithms can make nonparametric scene parsing more scalable to larger databases. The bank of random rotations algorithm is data-independent and therefore easily scalable to new categories and examples. The supervised sparse projections algorithm specifically targets search in high dimensions and enables faster encoding at test time. These contributions to approximate nearest neighbor search are general: they are not specific to scene parsing and can be applied to any unlabelled (bank of random rotations) or labelled (supervised sparse projections) database.

Section 1.2 briefly touched on the point of synergy between large-scale search and deep neural network classification frameworks. We believe this will be an important and fruitful avenue for future work. Though the techniques of binary encoding and deep learning may seem different at first glance, we are just now starting to see the exciting potential of their convergence. For example, the recently proposed XNOR-Net [106] is a fast and compact approximation to standard convolutional neural networks for classification in which filters and/or inputs are constrained to be binary. Binarizing the weights produces a network that requires 32 times less memory than a conventional network with single-precision weights, while maintaining comparable classification accuracy. Binarizing the inputs enables a speed-up of 58 times relative to standard 3x3 convolution, with some degradation in classification accuracy. XNOR-Net aims to enable real-time inference on devices with limited memory and no GPUs.

In closing, this is an exciting time for large-scale, data-driven computer vision. Tasks which had previously seemed far in the future, such as free-form visual question answering, semantic 3D reconstruction at an urban scale, and self-driving vehicles, are becoming more plausible with each year. The success of these projects will have widespread and lasting impact: on how we work, how we design our cities, and how we interact with machines.

Bibliography

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012. → pages 6, 10, 14, 15, 19, 25

[2] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. → pages 1

[3] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. → pages 14, 15

[4] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3D semantic parsing of large-scale indoor spaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. → pages 35

[5] A. Babenko and V. Lempitsky. Additive quantization for extreme vector compression. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. → pages 54

[6] A. Babenko and V. Lempitsky. Tree quantization for large-scale similarity search and classification. In *IEEE Conference Computer Vision and Pattern Recognition*, 2015. → pages 54

[7] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv:1511.00561, 2015. → pages 51

[8] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: a randomized correspondence algorithm for structural image editing. In *ACM SIGGRAPH*, 2009. → pages 12

[9] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *European Conference on Computer Vision*, 2010. → pages 2

[10] M. Bláha, C. Vogel, A. Richard, J. D. Wegner, T. Pock, and K. Schindler. Large-scale semantic 3D reconstruction: an adaptive multi-resolution model for multi-class volumetric labeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. → pages 35

[11] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. → pages 2, 20

[12] X. Boix, J. M. Gonfaus, J. van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzàlez. Harmony potentials: fusing global and local scale for semantic image segmentation. *International Journal of Computer Vision*, 96:83–102, 2012. → pages 11

[13] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9): 1124–1137, 2004. → pages 23

[14] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1222–1239, 2001. → pages 23

[15] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision*, 2008. → pages 3, 33

[16] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, 2012. → pages 14, 15

[17] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *European Conference on Computer Vision*, 2012. → pages 63

[18] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *ACM Symposium on Theory of Computing*, 2002. → pages xi, xii, 53, 63, 67, 70, 72, 73

[19] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: delving deep into convolutional nets. In *British Machine Vision Conference*, 2014. → pages xii, 20, 63, 69, 73

[20] M. Cheng, Z. Zhang, W. Lin, and P. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. → pages 14, 16

[21] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1996. → pages 37

[22] D. Dai, H. Riemenschneider, G. Schmitt, and L. V. Gool. Example-based facade texture synthesis. In *IEEE International Conference on Computer Vision*, 2013. → pages 2, 7

[23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005. → pages 14

[24] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 68

[25] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *IEEE International Conference on Computer Vision*, 2013. → pages 1, 16, 19

[26] D. Eigen and R. Fergus. Nonparametric image parsing using adaptive neighbor sets. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. → pages 6, 9, 12, 25

[27] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):222–234, 2014. → pages 14, 16

[28] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *International Conference on Machine Learning*, 2012. → pages 3, 12, 25

[29] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. → pages 1, 68

[30] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998. → pages 5

[31] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9): 1627–1645, 2010. → pages 2, 14

[32] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. → pages 24

[33] G. Floros and B. Leibe. Joint 2D-3D temporally consistent semantic segmentation of street scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. → pages 33

[34] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. → pages 82

[35] T. Ge, K. He, Q. Ke, and J. Sun. Optimized product quantization for approximate nearest neighbor search. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 54, 56, 58

[36] T. Ge, K. He, and J. Sun. Graph cuts for supervised binary coding. In *European Conference on Computer Vision*, 2014. → pages 54

[37] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision*, 2010. → pages 39

[38] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. → pages viii, 6, 47

[39] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *International Conference on Very Large Data Bases*, 1999. → pages xi, xii, 53, 63, 67, 70, 72, 73

[40] R. Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision*, 2015. → pages 15, 41

[41] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. → pages 2, 15

[42] Y. Gong, S. Kumar, H. A. Rowley, and S. Lazebnik. Learning binary codes for high-dimensional data using bilinear projections. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 54, 63

[43] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: a Procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2916–2929, 2013. → pages xi, xii, 7, 53, 54, 55, 56, 58, 63, 68, 70, 72, 73

[44] S. Gould and Y. Zhang. PatchMatchGraph: Building a graph of dense patch correspondences for label transfer. In *European Conference on Computer Vision*, 2012. → pages 12, 25

[45] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *IEEE International Conference on Computer Vision*, 2009. → pages 25

[46] C. Häne, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3D scene reconstruction and class segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 2, 33

[47] J. Hays and A. A. Efros. Scene completion using millions of photographs. In *ACM SIGGRAPH*, 2007. → pages 5

[48] K. He, J. Sun, and X. Tang. Guided image filtering. In *European Conference on Computer Vision*, 2010. → pages 1

[49] K. He, F. Wen, and J. Sun. K-means hashing: an affinity-preserving quantization method for learning binary compact codes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 53, 55, 58, 59, 60

[50] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015. → pages 41

[51] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. → pages 3

[52] G. Heitz and D. Koller. Learning spatial context: using stuff to find things. In *European Conference on Computer Vision*, 2008. → pages 6, 9

[53] J.-P. Heo, Z. Lin, and S.-E. Yoon. Distance encoded product quantization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. → pages 54

[54] L. Horne, J. Alvarez, C. McCarthy, M. Salzmann, and N. Barnes. Semantic labeling for prosthetic vision. *Computer Vision and Image Understanding*, in press. → pages 2

[55] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *British Machine Vision Conference*, 2014. → pages 16, 47

[56] X. Hou and L. Zhang. Saliency detection: a spectral residual approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. → pages 15

[57] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *International Symposium on Robotics Research*, 2011. → pages 38

[58] P. Isola and C. Liu. Scene collaging: analysis and synthesis of natural images with semantic layers. In *IEEE International Conference on Computer Vision*, 2013. → pages 14

[59] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. → pages 54

[60] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011. → pages 54, 58

[61] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093, 2014. → pages 51

[62] K. Jiang, Q. Que, and B. Kulis. Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. → pages 53

[63] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: distinctive parts for scene classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 1

[64] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray. Very high frame rate volumetric integration of depth images on mobile devices. In *International Symposium on Mixed and Augmented Reality*, 2015. → pages 38, 46

[65] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality*, 2007. → pages 37

[66] P. Kohli, L. Ladický, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82 (3):302–324, 2009. → pages 11, 33

[67] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004. → pages 23

[68] A. Kovashka, D. Parikh, and K. Grauman. WhittleSearch: image search with relative attribute feedback. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. → pages 1

[69] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Advances in Neural Information Processing Systems*, 2011. → pages 33, 40

[70] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *European Conference on Computer Vision*, 2014. → pages 14, 16

[71] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. → pages 58

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. → pages 3

[73] D. Kuettel and V. Ferrari. Figure-ground segmentation by transferring window masks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. → pages 13, 14

[74] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in Neural Information Processing Systems*, 2009. → pages 7, 53

[75] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(6): 1092–1104, 2012. → pages 53

[76] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3D reconstruction from monocular video. In *European Conference on Computer Vision*, 2014. → pages 2, 21, 34

[77] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical CRFs for object class image segmentation. In *IEEE International Conference on Computer Vision*, 2009. → pages 3, 11, 33

[78] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Graph cut based inference with co-occurrence statistics. In *European Conference on Computer Vision*, 2010. → pages 11

[79] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. → pages xii, 1, 68, 72

[80] F. Li, J. Carreira, G. Lebanon, and C. Sminchisescu. Composite statistical inference for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 13

[81] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. → pages 2

[82] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2368–2382, 2011. → pages vii, viii, 1, 2, 5, 9, 10, 11, 19, 22, 24, 25

[83] C. Liu, J. Yuen, and A. Torralba. SIFT Flow: dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011. → pages 5, 12

[84] S. Liu, X. Liang, L. Liu, X. Shen, J. Yang, C. Xu, L. Lin, X. Cao, and S. Yan. Matching-CNN meets KNN: quasi-parametric human parsing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. → pages 82

[85] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. → pages 53

[86] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of Exemplar-SVMs for object detection and beyond. In *IEEE International Conference on Computer Vision*, pages 89–96, 2011. → pages 13

[87] S. Manen, M. Guillaumin, and L. V. Gool. Prime object proposals with randomized Prim's algorithm. In *IEEE International Conference on Computer Vision*, 2013. → pages 14, 15

[88] A. Martinović, J. Knopp, H. Riemenschneider, and L. V. Gool. 3D all the way: semantic segmentation of urban scenes from start to end in 3D. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. → pages 2, 7, 34

[89] K. Matzen and N. Snavely. Scene chronology. In *European Conference on Computer Vision*, 2014. → pages 1

[90] S. McCann and D. G. Lowe. Spatially local coding for object recognition. In *Asian Conference on Computer Vision*, 2012. → pages 20

[91] H. Myeong, J. Y. Chang, and K. M. Lee. Learning object relationships via graph-based context model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. → pages 6, 9, 12, 25

[92] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *International Symposium on Mixed and Augmented Reality*, 2011. → pages 35, 37, 38

[93] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision*, 2011. → pages 37

[94] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics*, 32(6):169:1–169:11, 2013. → pages 38

[95] M. Norouzi and D. J. Fleet. Minimal loss hashing for compact binary codes. In *International Conference in Machine Learning*, 2011. → pages 7, 53

[96] M. Norouzi and D. J. Fleet. Cartesian k-means. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 54, 56, 58

[97] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. → pages 19, 58

[98] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2Text: describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems*, 2011. → pages 5

[99] V. Ordonez, J. Deng, Y. Choi, A. C. Berg, and T. L. Berg. From large scale image categorization to entry-level categories. In *IEEE International Conference on Computer Vision*, 2013. → pages 2

[100] C. Pantofaru, C. Schmid, and M. Hebert. Object recognition by integrating multiple image segmentations. In *European Conference on Computer Vision*, 2008. → pages 13

[101] D. Parikh and K. Grauman. Relative attributes. In *IEEE International Conference on Computer Vision*, 2011. → pages 1

[102] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *European Conference on Computer Vision*, 2010. → pages 2, 54, 63

[103] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *IEEE International Conference on Computer Vision*, 2007. → pages 11

[104] M. Raginsky and S. Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in Neural Information Processing Systems*, 2009. → pages 53

[105] M. Rastegari, C. Keskin, P. Kohli, and S. Izadi. Computationally bounded retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. → pages 54

[106] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-Net: ImageNet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, 2016. → pages 83

[107] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. → pages 2, 3, 15, 41, 46

[108] H. Riemenschneider, A. Bódis-Szomorú, J. Weissenberg, and L. V. Gool. Learning where to classify in multi-view semantic segmentation. In *European Conference on Computer Vision*, 2014. → pages 7, 34

[109] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, 2006. → pages 38

[110] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005. → pages 23

[111] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575, 2014. → pages xii, 3, 51, 69, 73

[112] B. C. Russell, A. Torralba, K. Murphy, and W. T. Freeman. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008. → pages 4, 10, 28

[113] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. S. Torr. Urban 3D semantic modelling using stereo vision. In *IEEE International Conference on Robotics and Automation*, 2013. → pages 2, 3, 7, 32, 33, 47, 48

[114] F. Shen, C. Shen, W. Liu, and H. T. Shen. Supervised discrete hashing. In *IEEE Conference Computer Vision and Pattern Recognition*, 2015. → pages xi, xii, 7, 54, 63, 66, 67, 68, 70, 72, 73

[115] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, 2006. → pages 11

[116] B. Shuai, G. Wang, Z. Zuo, B. Wang, and L. Zhao. Integrating parametric and non-parametric models for scene labeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. → pages 82

[117] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. → pages xii, 41, 51, 69, 72

[118] G. Singh and J. Košecká. Nonparametric scene parsing with adaptive feature relevance and semantic context. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 6, 9, 12, 25

[119] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision*, 2012. → pages 1

[120] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12): 1349–1380, 2000. → pages 1

[121] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *ACM SIGGRAPH*, 2006. → pages 1

[122] S. Song, S. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. → pages xi, 3, 63, 68, 70

[123] C. J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–620, 1977. → pages 4

[124] P. Sturgess, K. Alahari, L. Ladický, and P. H. S. Torr. Combining appearance and structure from motion features for road scene understanding. In *British Machine Vision Conference*, 2009. → pages 33

[125] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. → pages 4

[126] K. Tateno, F. Tombari, and N. Navab. When 2.5d is not enough: simultaneous reconstruction, segmentation and recognition on dense SLAM. In *IEEE International Conference on Robotics and Automation*, 2016. → pages 35

[127] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. *International Journal of Computer Vision*, 101 (2):329–349, 2013. → pages vii, ix, 1, 6, 9, 12, 13, 22, 23, 25, 26, 28, 29, 31

[128] J. Tighe and S. Lazebnik. Finding things: image parsing with regions and per-exemplar detectors. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 13, 25, 28

[129] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large data set for nonparametric object and scene recognition. *IEEE Transactions*

on Pattern Analysis and Machine Intelligence, 30(11):1958–1970, 2008. → pages 5, 58

[130] F. Tung and J. J. Little. CollageParsing: Nonparametric scene parsing by adaptive overlapping windows. In *European Conference on Computer Vision*, 2014. → pages iii, 82

[131] F. Tung and J. J. Little. SSP: Supervised Sparse Projections for large-scale retrieval in high dimensions. In *Asian Conference on Computer Vision*, 2016. → pages iv

[132] F. Tung and J. J. Little. Scene parsing by nonparametric label transfer of content-adaptive windows. *Computer Vision and Image Understanding*, 143:191–200, 2016. → pages iii, 23

[133] F. Tung, J. Martinez, H. H. Hoos, and J. J. Little. Bank of quantization models: A data-specific approach to learning binary codes for large-scale retrieval applications. In *IEEE Winter Conference on Applications of Computer Vision*, 2015. → pages iii

[134] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell. The NBNN kernel. In *IEEE International Conference on Computer Vision*, pages 1824–1831, 2011. → pages 20

[135] J. P. C. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. H. S. Torr. Mesh based semantic modelling for indoor and outdoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. → pages 2, 3, 7, 32, 33, 47, 48

[136] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *IEEE International Conference on Computer Vision*, 2011. → pages 14, 15

[137] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008. → pages 67

[138] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Pérez, and P. H. S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *IEEE International Conference on Robotics and Automation*, 2015. → pages 2, 3, 7, 32, 34, 46, 47, 48

[139] C. Wang, N. Komodakis, and N. Paragios. Markov Random Field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11): 1610–1627, 2013. → pages 11

[140] S. Wang, S. Fidler, and R. Urtasun. Holistic 3D scene understanding from a single geo-tagged image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. → pages 82

[141] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, 2008. → pages 53, 58

[142] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. → pages 5

[143] Y. Xia, K. He, P. Kohli, and J. Sun. Sparse projections for high-dimensional binary codes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. → pages xi, xii, 7, 54, 55, 63, 68, 70, 71, 72, 73

[144] J. Xiao and L. Quan. Multiple view semantic segmentation for street view images. In *IEEE International Conference on Computer Vision*, 2009. → pages 33

[145] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. → pages 19, 28, 68

[146] J. Yang, B. Price, S. Cohen, and M. Yang. Context driven scene parsing with attention to rare classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. → pages 13, 25, 26, 28, 30

[147] F. X. Yu, S. Kumar, Y. Gong, and S.-F. Chang. Circulant binary embedding. In *International Conference in Machine Learning*, 2014. → pages 54, 55

[148] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In *European Conference on Computer Vision*, 2014. → pages 1

[149] X. Zhang, F. X. Yu, R. Guo, S. Kumar, S. Wang, and S.-F. Chang. Fast orthogonal projection based on Kronecker product. In *IEEE International Conference on Computer Vision*, 2015. → pages 54

[150] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using Places database. In *Advances in Neural Information Processing Systems*, 2014. → pages xi, 68, 70

[151] C. L. Zitnick and P. Dollár. Edge Boxes: locating object proposals from edges. In *European Conference on Computer Vision*, 2014. → pages viii, 6, 10, 14, 16, 19, 20, 47