Brown University

Doctoral Thesis Proposal

Bayesian Nonparametric Discovery of Layers and Parts from Scenes and Objects

Author:
Soumya Ghosh

Supervisor:
Dr. Erik B. Sudderth
Readers:
Dr. Michael J. Black
Dr. James Hays

A thesis proposal submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Department of Computer Science
Brown University

November 2013

Contents

List of Figures

BROWN UNIVERSITY

Abstract

Thesis Proposal for Doctor of Philosophy in Computer Science

Bayesian Nonparametric Discovery of Layers and Parts from Scenes and Objects

by Soumya Ghosh

We develop statistical methods for analyzing natural images, videos and three–dimensional (3D) representations of articulated objects. The goal of our analysis is to discover and characterize regions, objects, and the parts constituting them. Images and videos typically exhibit wide variability in complexity, with some containing only a few objects while others depict complex 3D structure. The video regions corresponding to real-world objects exhibit strong spatio-temporal correlations, and are often spatially contiguous. Effective models for such data must automatically learn the number of constituent objects and parts, while simultaneously modeling strong spatio-temporal dependencies.

We study two flexible classes of priors that satisfy the above desiderata. We first study a distribution over partitions called the distance dependent Chinese restaurant process (ddCRP) that allows for dependencies between data instances. We show that a mixture model endowed with a ddCRP prior can produce state-of-the-art segmentations of 3D objects. We then develop hierarchical versions of the ddCRP prior better suited for image and video segmentation tasks. We also derive efficient MCMC algorithms for inference in the hierarchical ddCRP.

Next, focusing on images we explore a powerful class of models that generalize the Pitman-Yor (PY) process to produce depth-ordered decompositions of images into layers. Here, an ordered set of smooth Gaussian processes (GP) is used to encourage piecewise smooth allocation of pixels to layers. We develop methods for effective learning and robust inference in such models, and demonstrate competitive performance on standard image segmentation benchmarks.

Building on our layered segmentation model, we further propose significant extensions that allow shared segmentation of a corpus of partially labeled, color-and-depth images while simultaneously estimating coarse 3D representations of the semantic categories contained in the corpus. Using the beta-geometric process, a Bayesian nonparametric prior over matrices containing object occurrence counts, we model uncertainty in the

number of categories and instances occurring in each image. The final goal of this proposal is to develop efficient, reliable, and effective inference algorithms for the proposed model, allowing analysis of large-scale image collections.

Chapter 1

Introduction

Computer vision systems aim to make inferences about the the world by analyzing snapshots captured in images and videos. A wide variety of such systems have been developed to automatically analyze the semantic content of image corpuses, detect, characterize and track objects through image sequences and recover 3D scene geometry from image and video collections. They find use in augmented reality, multimedia retrieval, robot navigation and perception, remote sensing, cell counting in biology, crater detection and landform classification in planetary science.With the acquisition of images and related modalities (such as depth) getting progressively cheaper and image datasets growing to petabytes the impact of systems and techniques developed for computer vision problems will likely grow in the coming years.

The fundamental difficulty computer vision systems have to deal with is that the summary of the world provided by images and videos arises from an inherently lossy projection of the 3D world onto a 2D space. Recovery of the scene responsible for generating an image is under constrained and can only be solved under simplifying assumptions that constrain the problem. Additionally, realistic images and videos exhibit large appearance variability both within and between object classes, foreshortening, occlusion and illumination effects further complicating their analysis.

Statistical methods are widely used for building models robust to uncertainty exhibited by image and video data. Further, instead of reasoning about isolated pixels our models constrain the inference problem by considering pixels in context, combining local evidence from pixels with globally consistent interpretations.

This thesis focuses on two complimentary vision problems – understanding scenes described by images and videos by decomposing them into constituent regions and objects

and understanding objects through the discovery and analysis of object parts. We develop novel statistical models that build on recent advances in Bayesian nonparametrics and are robust to both uncertainty in the number and appearance of scene and object constituents while modeling dependencies between pixels.

1.1 Thesis Proposal Organization

The remainder of this proposal discusses:

Articulated Object Segmentation. In Chapter 2, we consider the problem of articulated 3D object segmentation. We develop a statistical model that combines a distance dependent Chinese restaurant process (ddCRP) prior over object partitions with likelihood distributions over affine transformations. Our model learns both the number and extent of independently deforming parts of objects from unlabeled data. We demonstrate the effectiveness of our model on a collection of human 3D scans of widely varying shape and in a variety of poses.

Image and Video segmentations with hierarchical ddCRPs. Chapter 3 develops hierarchical extensions to the ddCRP prior necessary for modeling partitions of natural images and videos. Approximate inference in these hierarchical models is more involved and we develop MCMC based Metropolis Hastings (MH) algorithms for exploring the intractable posteriors. Finally, we demonstrate competitive performance on standard image and video segmentation benchmarks.

Layered Image Segmentation. Chapter 4.1 focuses on segmenting monocular natural images into a set of depth ordered layers. The cardinality of the set is inferred automatically from the observed image. Building on the work of [4] we model image partitions through a collection of thresholded smooth functions sampled from Gaussian processes. We then develop novel learning and inference algorithms which allow efficient, robust and reliable recovery of layers and corresponding partition from natural images. The recovered image partitions are competitive with state-of-the-art image segmentation techniques on standard benchmarks.

Proposed work. Chapter 5 proposes extensions to the layered segmentation model that allow joint segmentation and estimation of coarse 3D structure from a corpus of RGB-D images. The proposed model uses sophisticated Bayesian nonparametric priors

for counting object categories and instances in the image corpus and models coarse object shape through thresholded Gaussian processes.

Chapter 2

Distance Dependent Chinese Restaurant Processes for 3D Object Segmentation

The *distance dependent Chinese restaurant process* (ddCRP) [5] is a generalization of the Chinese restaurant process (CRP). The CRP induces an exchangeable distribution on all possible partitions of a set of objects [6]. While exchangeability provides a computational advantage, from the perspective of approximate inference, it is an unrealistic assumption in our problems of interest where neighboring data instances are far more likely to belong to the same partition component.

The ddCRP relaxes the CRP exchangeability assumption and accommodates random partitions of non-exchangeable data [7]. It alters the CRP by modeling customer links not to tables, but to other customers. The link c_m for customer m is sampled according to the distribution

$$p\left(c_m = n \mid D, f, \alpha\right) \propto \begin{cases} f(d_{mn}) & m \neq n, \\ \alpha & m = n. \end{cases} \tag{2.1}$$

Here, d_{mn} is an externally specified distance between data points m and n, and α determines the probability that a customer links to themselves rather than another customer. D is a matrix of pairwise distances with $D[m, n] = d_{mn}$. The monotonically decreasing decay function $f(d)$ mediates how the distance between two data points affects their probability of connecting to each other. The overall link structure specifies a partition: two customers are clustered together if and only if one can reach the other by traversing the link edges.

The ddCRP can capture a wide variety of correlations among the data through the specification of appropriate choices of distance and decay functions. They have been used for language modeling, clustering time stamped documents and networked data [5]. In this chapter, we use the ddCRP to segment articulated 3D objects. In Chapter 3 we present hierarchical extensions to the ddCRP better suited for image and video segmentation problems.

2.1 Distance dependent CRP mixtures

Like the CRP the ddCRP is a valid distribution over partitions [5] and can be used as an allocation prior in mixture models. In this setting, the ddCRP clusters data in a biased way: each data point is more likely to be clustered with other data that are near it according to the externally specified distance d. This provides us with a flexible class of models that both learn the cardinality of the partition and capture apriori notions of correlations in the data.

The ddCRP mixture generates data as follows:

- For each data instance $i \in [1, N]$ sample a customer link c_i according to equation 2.1. The connected components of the links $C = \{c_i \mid i = 1, \ldots, \mathrm{N}\}$ determine a partition of the dataset $Z(C) = \{z_i \mid i = 1, \ldots, \mathrm{N}\}$.
- For each component $k \in \{1, \ldots, \}$ sample a data generating parameter from a base distribution $\phi_k \sim G_0$.
- Finally, generate data $X = \{x_i \mid i = 1, \ldots, \mathrm{N}\}$ by sampling the data generating distribution $x_i \sim p(x_i \mid \phi_{z_i})$.

Notice that the prior term uses the customer representation to take into account distances between data points while the likelihood term uses the cluster representation to generate observations.

2.1.1 Inference with Gibbs Sampling

The ddCRP mixture has two sets of latent variables the customer assignments C and the cluster specific data generating distribution parameters ϕ_k. The distribution of C conditioned on the observed data X and the model parameters ,with ϕ_k marginalized out is:

$$p(C \mid X, \alpha, d, f, G_0) = \frac{\left(\prod_{n=1}^{N} p(c_i \mid D, f, \alpha)\right) p(X \mid Z(C), G_0)}{\sum_C \left(\prod_{n=1}^{N} p(c_i \mid D, f, \alpha)\right) p(X \mid Z(C), G_0)} \tag{2.2}$$

where $Z(C)$ is the cluster representation derived from the customer representation C.

The posterior in Equation (2.2) is not tractable to compute and we approximate it using Gibbs sampling by iteratively sampling each latent variable c_i conditioned on the others and the observations,

$$p(c_i \mid c_{-i}, X, D, \alpha, G_0) \propto p(c_i \mid D, \alpha) p(X \mid Z(C), G_0). \tag{2.3}$$

The prior term is given in Equation (2.1). We can decompose the likelihood term as follows:

$$\begin{aligned} p(X \mid Z(C), G_0) &= \prod_{k=1}^{K(C)} \int p(\phi_k \mid G_0) \prod_{\{i \mid Z(C)_i = k\}} p(x_i \mid \phi_k) d\phi_k \\ &= \prod_{k=1}^{K(C)} p(X_{Z(C)=k} \mid Z(C), G_0). \end{aligned} \tag{2.4}$$

We have introduced notation to more easily move from the customer representation—the primary latent variables of our model—and the cluster representation. We let $K(C)$ be the number of unique clusters in the customer assignments, $z(C)$ be the cluster assignments derived from the customer assignments, and $X_{Z(C)=k}$ be the collection of observations assigned to the kth cluster. When the base distribution G_0 is conjugate to the data generating distribution the integral in Equation (2.4) is easily computed. In nonconjugate settings, ϕ_k can no longer be analytically marginalized out and an additional layer of sampling is needed to deal with them.

Sampling from Equation (2.3) happens in two stages. First, we remove the customer link c_i from the current configuration. Then, we consider the prior probability of each possible value of c_i and how it changes the likelihood term, by moving from $p(X \mid Z(C_{-i}), G_0)$ to $p(X \mid Z(C), G_0)$.

In the first stage, removing c_i either leaves the cluster structure intact, i.e., $Z(C^{\text{old}}) = Z(C_{-i})$, or splits the cluster assigned to data point i into two. In the second stage, randomly reassigning c_i either leaves the cluster structure intact, i.e., $Z(C_{-i}) = Z(C)$, or joins the cluster assigned to data point i to another. Via these moves, the sampler explores the space of possible segmentations.

Let ℓ and m be the indices of the tables that are joined to index k. We first remove c_i, possibly splitting a cluster. Then we sample from

$$p(c_i \mid c_{-i}, X, D, \alpha, G_0) \propto \begin{cases} p(c_i \mid D, \alpha) \Gamma(X, Z, G_0) & \text{if } c_i \text{ joins } \ell \text{ and } m; \\ p(c_i \mid D, \alpha) & \text{otherwise,} \end{cases} \tag{2.5}$$

where

$$\Gamma(X, Z, G_0) = \frac{p(X_{Z(C)=k} \mid G_0)}{p(X_{Z(C)=\ell} \mid G_0) p(X_{Z(C)=m} \mid G_0)}. \tag{2.6}$$

This defines a Markov chain whose stationary distribution is the posterior of the ddCRP mixture.

2.2 Motion based 3D object segmentation

In this section, we leverage the ddCRP mixture machinery for performing mesh segmentation. Mesh segmentation methods decompose a three-dimensional (3D) mesh, or a collection of aligned meshes, into their constituent parts. This well-studied problem has numerous applications in computational graphics and vision, including texture mapping, skeleton extraction, morphing, and mesh registration and simplification. We focus in particular on the problem of segmenting an articulated object, given aligned 3D meshes capturing various object poses. The meshes we consider are complete surfaces described by a set of triangular faces, and we seek a segmentation into spatially coherent parts whose spatial transformations capture object articulations. Applied to various poses of human bodies as in Figure 2.1, our approach identifies regions of the mesh that deform together, and thus provides information which could inform applications such as the design of protective clothing.

Mesh segmentation has been most widely studied as a static clustering problem, where a single mesh is segmented into "semantic" parts using low-level geometric cues such as distance and curvature [8, 9]. While supervised training data can sometimes lead to improved results [10], there are many applications where such data is unavailable, and the proper way to partition a single mesh is inherently ambiguous. By searching for parts which deform consistently across many meshes, we create a better-posed problem whose solution is directly useful for modeling objects in motion.

Several issues must be addressed to effectively segment collections of articulated meshes. First, the number of parts comprising an articulated object is unknown *a priori*, and must be inferred from the observed deformations. Second, mesh faces exhibit strong spatial correlations, and the inferred parts must be contiguous. This spatial connectivity is needed to discover parts which correspond with physical object structure, and required by target applications such as skeleton extraction. Finally, our primary goal is to understand the structure of human bodies, and humans vary widely in size and shape. People move and deform in different ways depending on age, fitness, body fat, etc. A segmentation of the human body should take into account this range of variability in the population. To our knowledge, no previous methods for segmenting meshes combine

FIGURE 2.1: Human body segmentation. *Left:* Reference poses for two female bodies, and those bodies captured in five other poses. *Right:* A manual segmentation used to align these meshes [1], and the segmentation inferred by our ddCRP model from 56 poses. The ddCRP segmentation discovers parts whose motion is nearly rigid, and includes small parts such as elbows and knees absent from the manual segmentation.

information about deformation from multiple bodies to address this *corpus segmentation* problem.

In the rest of this section, we develop a statistical model which addresses all of these issues. We adapt the ddCRP to model spatial dependencies among mesh triangles, and enforce spatial contiguity of the inferred parts [11]. Unlike most previous mesh segmentation methods, our approach allows data-driven inference of an appropriate number of parts, and uses an affine transformation-based likelihood to accommodate object instances of varying shape.

2.2.1 A Part-Based Model for Mesh Deformation

Consider a collection of J meshes, each with N triangles. For some input mesh j, we let $y_{jn} \in \mathbb{R}^3$ denote the 3D location of the center of triangular face n, and $Y_j = [y_{j1}, \ldots, y_{jN}] \in \mathbb{R}^{3\times N}$ the full mesh configuration. Each mesh j has an associated N-triangle reference mesh, indexed by b_j. We let $x_{bn} \in \mathbb{R}^4$ denote the location of triangle n in reference mesh b, expressed in homogeneous coordinates ($x_{bn}(4) = 1$). A full reference mesh $X_b = [x_{b1}, \ldots, x_{bN}]$. In our later experiments, Y_j encodes the 3D mesh for a person in pose j, and X_{b_j} is the reference pose for the same individual.

We estimate aligned correspondences between the triangular faces of the input pose meshes Y_j, and the reference meshes X_b, using a recently developed method [1]. This approach robustly handles 3D data capturing varying shapes and poses, and outputs meshes which have equal numbers of faces in one-to-one alignment. Our segmentation model does not depend on the details of this alignment method, and could be applied to data produced by other correspondence algorithms.

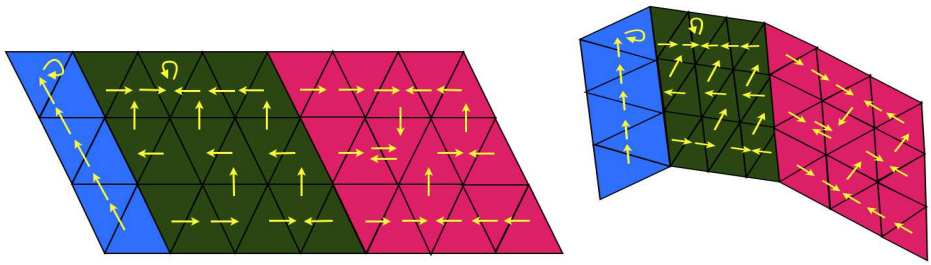

FIGURE 2.2: *Left:* A reference mesh in which links (yellow arrows) currently define three parts (connected components). *Right:* Each part undergoes a distinct affine transformation, generated as in Equation (2.7).

2.2.2 Nonparametric Spatial Priors for Mesh Partitions

The ddCRP, endowed with an appropriate distance function, is particularly well suited for modeling segmentations of articulated objects. In addition to allowing data-driven inference of the true number of mostly-rigid parts underlying the observed data and encouraging spatially adjacent triangles to lie in the same part, it *guarantees* that all inferred parts are spatially contiguous.

We define the distance between two triangles as the minimal number of hops, between adjacent faces, required to reach one triangle from the other. A "window" decay function of width 1, $f(d) = \mathbf{1}_{d \leq 1}$, then restricts triangles to link only to immediately adjacent faces. Note that this doesn't limit the size of parts, since all pairs of faces are potentially reachable via a sequence of adjacent links. However, it does guarantee that only spatially contiguous parts have non-zero probability under the prior. This constraint is preserved by our MCMC inference algorithm.

2.2.3 Modeling Part Deformation via Affine Transformations

Articulated object deformation is naturally described via the spatial transformations of its constituent parts. We expect the triangular faces within a part to deform according to a coherent part-specific transformation, up to independent face-specific noise. The near-rigid motions of interest are reasonably modeled as affine transformations, a family of co-linearity preserving linear transformations. We concisely denote the transformation from a reference triangle to an observed triangle via a matrix $A \in \mathbb{R}^{3\times 4}$. The fourth column of A encodes translation of the corresponding reference triangle via homogeneous coordinates x_{bn}, and the other entries encode rotation, scaling, and shearing.

Previous approaches have treated such transformations as parameters to be estimated during inference [12, 13]. Here, we instead define a prior distribution over affine transformations. Our construction allows transformations to be analytically marginalized when

learning our part-based segmentation, but retains the flexibility to later estimate transformations if desired. Explicitly modeling transformation uncertainty makes our MCMC inference more robust and rapidly mixing [14], and also allows data-driven determination of an appropriate number of parts.

The matrix of numbers encoding an affine transformation is naturally modeled via multivariate Gaussian distributions. We place a conjugate, matrix normal-inverse-Wishart [15, 16] prior on the affine transformation A and residual noise covariance matrix Σ:

$$\begin{aligned} \Sigma &\sim \mathcal{IW}(n_0, S_0) \\ A \mid \Sigma &\sim \mathcal{MN}(M, \Sigma, K) \end{aligned} \tag{2.7}$$

Here, $n_0 \in \mathbb{R}$ and $S_0 \in \mathbb{R}^{3\times 3}$ control the variance and mean of the Wishart prior on Σ^{-1}. The mean affine transformation is $M \in \mathbb{R}^{3\times 4}$, and $K \in \mathbb{R}^{4\times 4}$ and Σ determine the variance of the prior on A. Applied to mesh data, these parameters have physical interpretations and can be estimated from the data collection process. While such priors are common in Bayesian regression models, our application to the modeling of geometric affine transformations appears novel.

Allocating a different affine transformation for the motion of each part in each pose (Figure 2.2), the overall generative model can be summarized as follows:

1. For each triangle i, sample an associated link $c_i \sim \text{ddCRP}(\alpha, f, D)$. The part assignments z are a deterministic function of the sampled links $C = [c_1, \ldots, c_N]$.

2. For each pose j of each part k, sample an affine transformation A_{jk} and residual noise covariance Σ_{jk} from the matrix normal-inverse-Wishart prior of Equation (2.7).

3. Given these pose-specific affine transformations and assignments of mesh faces to parts, independently sample the observed location of each pose triangle relative to its corresponding reference triangle, $y_{ji} \sim \mathcal{N}(A_{jz_i} x_{bji}, \Sigma_{jz_i})$.

Note that Σ_{jk} governs the degree of non-rigid deformation of part k in pose j. It also indirectly influences the number of inferred parts: a large S_0 makes large Σ_{jk} more probable, which allows more non-rigid deformation and permits models which utilize

fewer parts. The overall model is

$$p(\mathbf{Y}, c, A, \Sigma \mid \mathbf{X}, b, D, \alpha, f, \eta) = p(c \mid D, f, \alpha) \prod_{j=1}^{J} \left[\prod_{k=1}^{K(c)} p(A_{jk}, \Sigma_{jk} \mid \eta) \right] \left[\prod_{i=1}^{N} \mathcal{N}(y_{ji} \mid A_{jz_i} x_{b_j i}, \Sigma_{jz_i}) \right] \quad (2.8)$$

where $\mathbf{Y} = \{Y_1, \ldots, Y_J\}$, $\mathbf{X} = \{X_1, \ldots, X_B\}$, $b = [b_1, \ldots, b_J]$, the ddCRP links C define assignments z to $K(C)$ parts, and $\eta = \{n_0, S_0, M, K\}$ are likelihood hyperparameters. There is a single reference mesh X_b for each object instance b, and Y_j captures a single deformed pose of X_{b_j}.

2.2.4 Related Work

Previous work has also sought to segment a mesh into parts based on observed articulations [12, 17–19]. The two-stage procedure of Rosman et al. [18] first minimizes a variational functional regularized to favor piecewise constant transformations, and then clusters the transformations into parts. Several other segmentation procedures [17, 19] lack coherent probabilistic models, and thus have difficulty quantifying uncertainty and determining appropriate segmentation resolutions.

Anguelov et al. [12] define a global probabilistic model, and use the EM algorithm to jointly estimate parts and their transformations. They explicitly model spatial dependencies among mesh faces, but their Markov random field cannot ensure that parts are spatially connected; a separate connected components process is required. Heuristics are used to determine an appropriate number of parts.

Ambitious recent work has considered a model for joint mesh alignment and segmentation [13]. However, this approach suffers from many of the issues noted above: the number of parts must be specified *a priori*, parts may not be contiguous, and their EM inference appears prone to local optima.

Mixture of Regression models. Although, not presented as such both the mesh-crp and mesh-ddcrp models are a special case of the mixture of regressions [20] framework. In particular, when $J = 1$, both models extend traditional mixtures of linear regressions. The crp-mesh model can be seen as a Bayesian nonparametric mixture of linear regressions [21] and the ddcrp-mesh model is a further extension that accounts for dependencies among data when inferring the regression model responsible for explaining a group of feature response pairs. Finally when $J > 1$ our models provide a further generalization – the ability to model multiple outputs.

2.2.5 Inference

We seek the constituent parts of an articulated model, given observed data ($\mathbf{X}$, $\mathbf{Y}$, and b). These parts are characterized by the posterior distribution of the customer links c_i. Following, Section 2.1.1 we develop a collapsed Gibbs sampler, which iteratively draws c_i from the conditional distribution:

$$p(c_i \mid c_{-i}, \mathbf{X}, \mathbf{Y}, b, D, f, \alpha, \eta) \propto p(c_i \mid D, f, \alpha) p(\mathbf{Y} \mid Z(C), \mathbf{X}, b, \eta). \tag{2.9}$$

Here, $Z(C)$ is the clustering into parts defined by the customer links C. The likelihood term in the above equation factorizes as:

$$p(\mathbf{Y} \mid Z(C), \mathbf{X}, b, \eta) = \prod_{k=1}^{K(C)} \prod_{j=1}^{J} p(Y_{jk} \mid X_{b_j k}, \eta) \tag{2.10}$$

where $Y_{jk} \in \mathbb{R}^{3 \times N_k}$ is the set of triangular faces in part k of pose j, and $X_{b_j k}$ are the corresponding reference faces. Exploiting the conjugacy of the normal likelihood to the prior over affine transformations in Equation (2.7), we marginalize the part-specific latent variables A_{jk} and Σ_{jk} to compute the marginal likelihood in closed form (Section A.1):

$$p(Y_{jk} \mid X_{b_j k}, \eta) = \frac{|K|^{3/2} |S_0|^{(n_0/2)} \Gamma_3\left(\frac{N_k + n_0}{2}\right)}{\pi^{(3N_k/2)} |S_{xx}|^{(3/2)} |S_0 + S_{y|x}|^{((N_k+n_0)/2)} \Gamma_3(\frac{n_0}{2})}, \tag{2.11}$$

$$S_{xx} = X_{b_j k} {X_{b_j k}}^T + K, \quad S_{yx} = Y_{jk} {X_{b_j k}}^T + MK, \tag{2.12}$$

$$S_{y|x} = Y_{jk} {Y_{jk}}^T + MKM^T - S_{yx} (S_{xx})^{-1} S_{yx}^T. \tag{2.13}$$

Putting Equation (2.5) and Equation (2.13) we have the required posterior distribution:

$$p(c_i \mid c_{-i}, \mathbf{X}, \mathbf{Y}, b, D, f, \alpha, \eta) \propto \begin{cases} p(c_i \mid D, f, \alpha) \Delta(\mathbf{Y}, \mathbf{X}, b, Z(C), \eta) & \text{if } c_i \text{ links } k_1 \text{ and } k_2; \\ p(c_i \mid D, \alpha) & \text{otherwise,} \end{cases}$$

$$\Delta(\mathbf{Y}, \mathbf{X}, b, Z(C), \eta) = \frac{\prod_{j=1}^{J} p(Y_{jk_1 \cup k_2} \mid X_{b_j k_1 \cup k_2}, \eta)}{\prod_{j=1}^{J} p(Y_{jk_1} \mid X_{b_j k_1}, \eta) \prod_{j=1}^{J} p(Y_{jk_2} \mid X_{b_j k_2}, \eta)}. \tag{2.14}$$

Here, k_1 and k_2 are parts in $z(c_{-i})$. Note that if the mesh segmentation C is the only quantity of interest, the analytically marginalized affine transformations A_{jk} need not be directly estimated. However, for some applications the transformations are of direct interest. Given a sampled segmentation, the part-specific parameters for pose j have

the following posterior [15]:

$$p(A_{jk}, \Sigma_{jk} \,|\, Y_{jk}, X_{b_jk}, \eta) \propto \mathcal{MN}(A_{jk} \,|\, S_{yx}S_{xx}^{-1}, \Sigma_{jk}, S_{xx})\mathcal{IW}(\Sigma_{jk} \,|\, N_k + n_0, S_{y|x} + S_0) \tag{2.15}$$

Marginalizing the noise covariance matrix, the distribution over transformations is then

$$\begin{aligned} p(A_{jk} \,|\, Y_{jk}, X_{b_jk}, \eta) &= \int \mathcal{MN}(A_{jk} \,|\, S_{yx}S_{xx}^{-1}, \Sigma_{jk}, S_{xx}) IW(\Sigma_{jk} \,|\, N_k + n_0, S_{y|x} + S_0)\, d\Sigma_{jk} \\ &= \mathcal{MT}(A_{jk} \,|\, N_k + n_0, S_{yx}S_{xx}^{-1}, S_{xx}, S_{y|x} + S_0) \end{aligned} \tag{2.16}$$

where $\mathcal{MT}(\cdot)$ is a matrix-t distribution [16] with mean $S_{yx}S_{xx}^{-1}$, and $N_k + n_0 - 2$ degrees of freedom.

2.2.6 Experimental Results

We now experimentally validate, the *mesh-ddcrp* model developed in the previous sections. Both qualitative and quantitative comaprisons are provided. Because "ground truth" parts are unavailable for the real body pose datasets of primary interest, we propose an alternative evaluation metric based on the prediction of held-out object poses, and show that the mesh-ddcrp performs favorably against competing approaches.

We primarily focus on a collection of 56 training meshes, acquired and aligned [1] from 3D scans of two female subjects in 27 and 29 poses. For quantitative tests, we employ 12 meshes of each of six different female subjects [22] (Figure 2.4). For each subject, a mesh in a canonical pose is chosen as the reference mesh (Figure 2.1). These meshes contain about 20,000 faces.

2.2.6.1 Hyperparameter Specification and MCMC Learning

The hyperparameters that regularize our mesh-ddcrp prior have intuitive interpretations, and can be specified based on properties of the mesh data under consideration. As described in Section 2.2.2, the ddCRP distances D and f are set to guarantee spatially connected parts. The self-connection parameter is set to a small value, $\alpha = 10^{-8}$, to encourage creation of larger parts.

The matrix normal-inverse-Wishart prior on affine transformations A_{jk}, and residual noise covariances Σ_{jk}, has hyperparameters $\eta = \{n_0, S_0, M, K\}$. The mean affine transformation M is set to the identity transformation, because on average we expect mesh faces to undergo small deformations. For the noise covariance prior, we set the degrees of freedom $n_0 = 5$, a value which makes the prior variance nearly as large as possible

while ensuring that the mean remains finite. The expected part variance S_0 captures the degree of non-rigidity which we expect parts to demonstrate, as well as noise from the mesh alignment process. The correspondence error in our human meshes is approximately 0.01m; allowing for some part non-rigidity, we set $\sigma = 0.015$m and $S_0 = \sigma^2 \times \mathbf{I}_{3\times 3}$. K is a precision matrix set to $K = \sigma^2 \times \text{diag}(1, 1, 1, 0.1)$.The Kronecker product of K^{-1} and S_0 governs the covariance of the distribution on A. Our settings make this nearly identity for most components, but the translation components of A have variance which is an order of magnitude larger, so that the expected scale of the translation parameters matches that of the mesh coordinates.

In our experiments, we ran the mesh-ddcrp sampler for 200 iterations from each of five random initializations, and selected the most probable posterior sample. The computational cost of a Gibbs iteration scales linearly with the number of meshes; our unoptimized Matlab[1] implementation required around 10 hours to analyze 56 human meshes.

2.2.6.2 Baseline Segmentation Methods

We compare the mesh-ddcrp model to three competing methods. The first is a modified agglomerative clustering technique [23] which enforces spatial contiguity of the faces within each part. At initialization, each face is deemed to be its own part. Adjacent parts on the mesh are then merged based on the squared error in describing their motion by affine transformations. Only adjacent parts are considered in these merge steps, so that parts remain spatially connected.

Our second baseline is based on a publicly available implementation of spectral clustering methods [24], a popular approach which has been previously used for mesh segmentation [25]. We compare to an affinity matrix specifically designed to cluster faces with similar motions [26]. The affinity between two mesh faces u, v is defined as $C_{uv} = \exp\{-\frac{\sigma_{uv}+\sqrt{m_{uv}}}{S^2}\}$, where $m_{uv} = \frac{1}{J^2}\sum_j \delta_{uvj}$, δ_{uvj} is the Euclidean distance between u and v in pose j, $\sigma_{uv} = \sqrt{\frac{1}{J}\sum_j (\delta_{uvj} - \bar{\delta}_{uv})^2}$ is the corresponding standard deviation, and $S = \frac{1}{M}\sum_{u,v} \sigma_{uv} + \sqrt{m_{uv}}$ for all M pairs of faces u, v.

For the agglomerative and spectral clustering approaches, the number of parts must be externally specified; we experimented with $K = 5, 10, 15, 20, 25, 30$ parts. We also consider a Bayesian nonparametric baseline which replaces the ddCRP prior over mesh partitions with a standard CRP prior. The resulting *mesh-crp* model may estimate the number of parts, but doesn't model mesh structure or enforce part contiguity. The expected number of parts under the CRP prior is roughly $\alpha \log N$; we set $\alpha = 2$ so that

[1] Available at www.cs.brown.edu/ sghosh

the expected number of mesh-crp parts is similar to the number of parts discovered by the mesh-ddcrp. To exploit bilateral symmetry, for all methods we only segment the right half of each mesh. The resulting segmentation is then reflected onto the left half.

2.2.6.3 Part Discovery and Motion Prediction

We first consider the synthetic Tosca dataset [2], and separately analyze the Centaur (six poses) and Horse (eight poses) meshes. These meshes contain about 31,000 and 38,000 triangular faces, respectively. Figure 2.3 displays the segmentations of the Tosca meshes inferred by mesh-ddcrp. The inferred parts largely correspond to groups of mesh faces which undergo similar transformations.

Figure 2.4 displays the results produced by the ddCRP, as well as our baseline methods, on the human mesh data. Qualitatively, the segmentations produced by mesh-ddcrp correspond to our intuitions about the body. Note that in addition to capturing the head and limbs, the segmentation successfully segregates distinctly moving small regions such as knees, elbows, shoulders, biceps, and triceps. In all, the mesh-ddcrp detects 20 distinctly moving parts for one half of the body.

We now introduce a quantitative measure of segmentation quality: segmentations are evaluated by their ability to explain the articulations of test meshes with novel shapes and poses. Given a collection of T test meshes Y_t with corresponding reference meshes X_{b_t}, and a candidate segmentation into K parts, we compute

$$\mathcal{E} = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K} ||Y_{tk} - A^*_{tk}X_{b_tk}||_2. \tag{2.17}$$

Here, A^*_{tk} is the least squares estimate of the single affine transformation responsible for mapping X_{b_tk} to Y_{tk}. Note that Equation (2.17) is trivially zero for a degenerate solution wherein each mesh face is assigned to its own part. However, segmentations of similar resolution may safely be compared using Equation (2.17), with lower errors corresponding to better segmentations.

On our test set of human meshes, the mesh-ddcrp model produces an error of $\mathcal{E} = 1.39$ meters, which corresponds to sub-millimeter accuracy when normalized by the number of faces. Figure 2.4 displays a plot comparing the errors achieved by the different methods. Mesh-ddcrp is significantly better than all other methods, including for settings of K which allocate 50% more parts to competing approaches, according to a Wilcoxon's signed rank test (5% significance level).

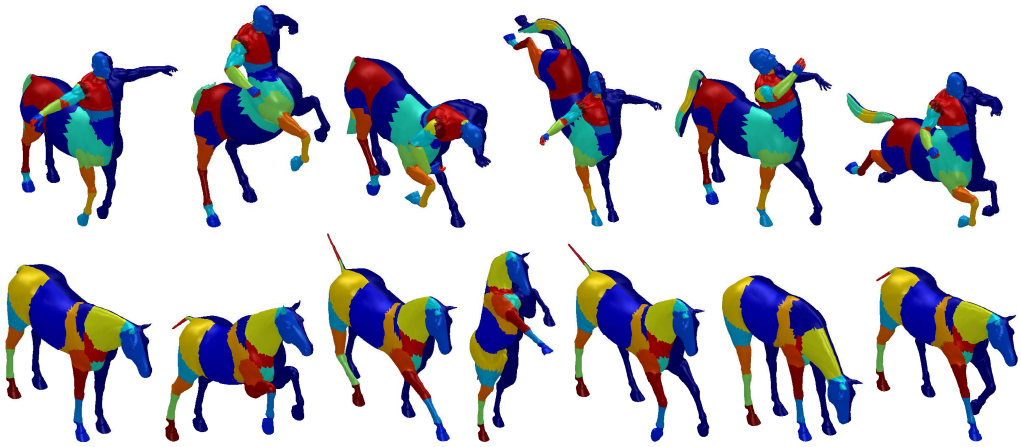

FIGURE 2.3: Segmentations produced by mesh-ddcrp on synthetic Tosca meshes [2]. The first mesh in each row displays the chosen reference mesh. For illustration, we have only segmented the right half of each mesh.

Next, we demonstrate the benefits of sharing information among differently shaped bodies. We selected an illustrative articulated pose for each of the two training subjects in addition to their respective reference poses (Figure 2.4). The chosen poses either exhibit upper or lower body deformations, but not both. The meshes were then segmented both independently for the two subjects and jointly sharing information across subjects. Figure 2.5 demonstrates that the independent segmentations exhibit both undersegmented (legs in the first set) and oversegmented (head in the second) parts. However, sharing information among subjects results in parts which correspond well with physical human bodies. Note that with only two articulated poses, we are able to generate meaningful segmentations in about an hour of computation. This data-limited scenario also demonstrates the benefits of the ddCRP prior: as shown in Figure 2.5, the parts extracted by mesh-crp are "patchy", spatially disconnected, and physically implausible.

2.3 Discussion

Adapting the ddCRP to collections of 3D meshes, we have developed an effective approach for the discovery an unknown number of parts underlying articulated object motion. Unlike previous methods, our model guarantees that parts are spatially connected, and uses transformations to model instances with potentially varying body shapes. Via a novel application of matrix normal-inverse-Wishart priors, our sampler analytically marginalizes transformations for improved efficiency. While we have modeled part motion via affine transformations, future work should explore more accurate Lie algebra characterizations of deformation manifolds [27].

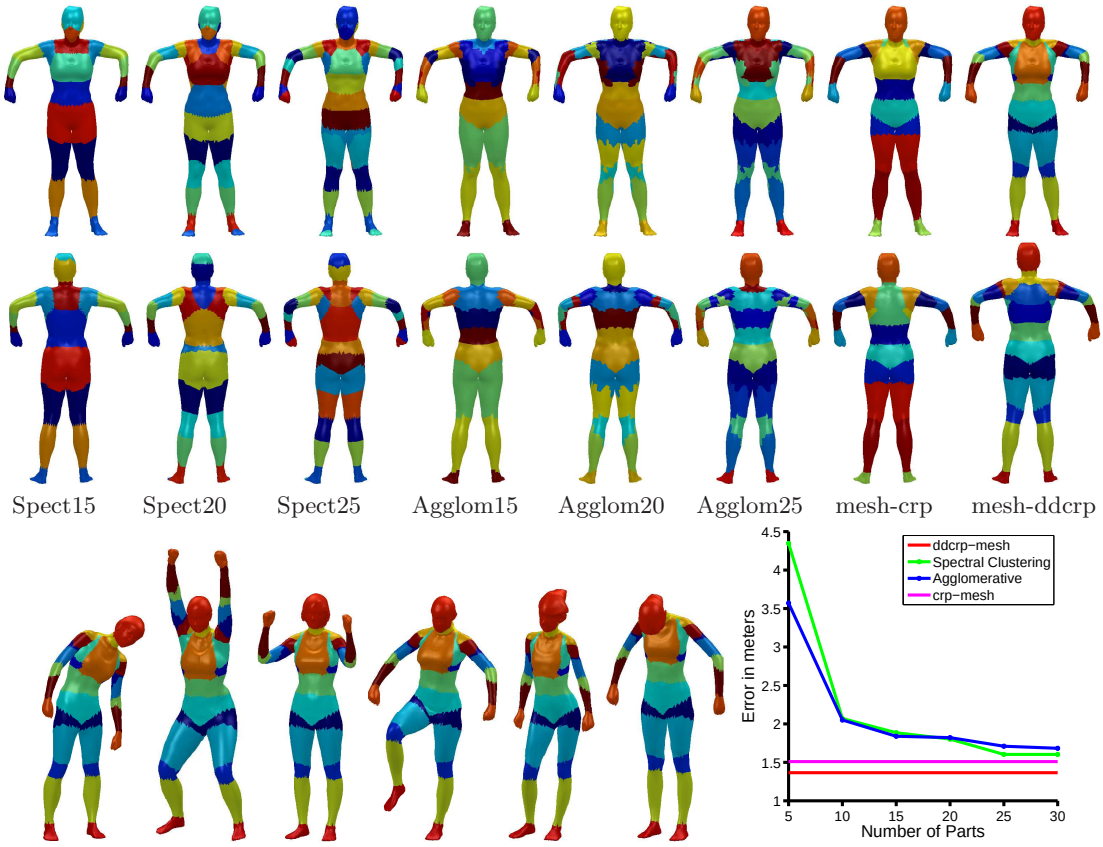

FIGURE 2.4: *Top two rows (left to right):* Segmentations produced by spectral and agglomerative clustering with 15, 20, and 25 clusters respectively, followed by the mesh-crp and mesh-ddcrp segmentations. *Bottom row:* Test set results. We display mesh-ddcrp segmentations for several test meshes, and quantitatively compare methods.

Experiments with dozens of real human body poses provide strong quantitative evidence that our approach produces state-of-the-art segmentations with many potential applications.

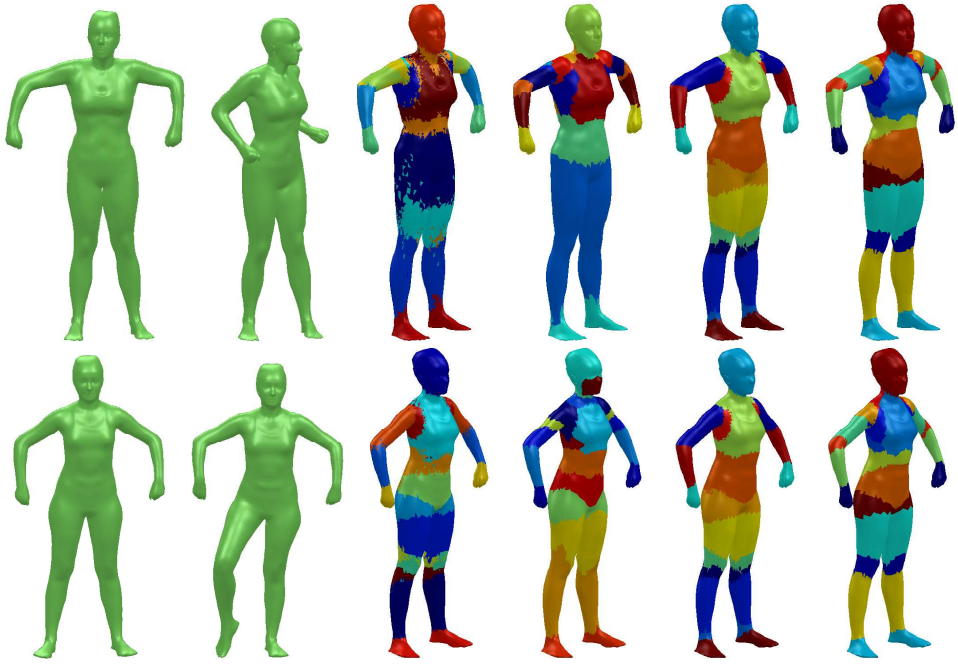

FIGURE 2.5: Impact of sharing information across bodies with varying shapes. The two rows correspond to the training subjects. Each row displays the reference pose, an illustrative articulated pose, mesh-crp and mesh-ddcrp segmentations produced by independently segmenting the pair of poses of each individual, and mesh-crp and mesh-ddcrp segmentations produced by jointly segmenting the chosen poses from both subjects.

Chapter 3

Image and Video Segmentations with Hierarchical ddCRPs

In this chapter we develop hierarchical generalizations of the ddCRP mixture model, motivated by the short comings of the ddCRP mixture models in modeling visual phenomena.

In Chapter 2 we saw that the ddCRP mixture can group similarly deforming mesh faces together, producing meaningful decompositions of 3D objects. We could similarly attempt to use a ddCRP mixture model for grouping pixels of similar appearance. When applied to images, ddCRP mixtures produce an over-segmentation consisting of a large number of small contiguous patches homogeneous in color and texture features (Figure 3.1). While such segmentations are useful for various applications [28], they do not reflect the statistics of human produced segmentations which consist of regions whose sizes follow power law distributions [4]. Introducing a hierarchy wherein the produced patches are grouped into a small number of regions, produces segments which match the statistics of human annotated segments better (Figure 3.1). In Chapter 4 we will explore models which explicitly model the power law behavior exhibited in human segmentations.

Another motivation for developing hierarchical models stems from the need to share segments between images. Consider a video sequence with objects spanning several frames. Segments representing these objects must then be shared among the frames. Additionally, the segments need to exhibit similar size, shape and exhibit coherent motion across frames. The hierarchical ddCRP, discussed in this chapter, captures these desiderata

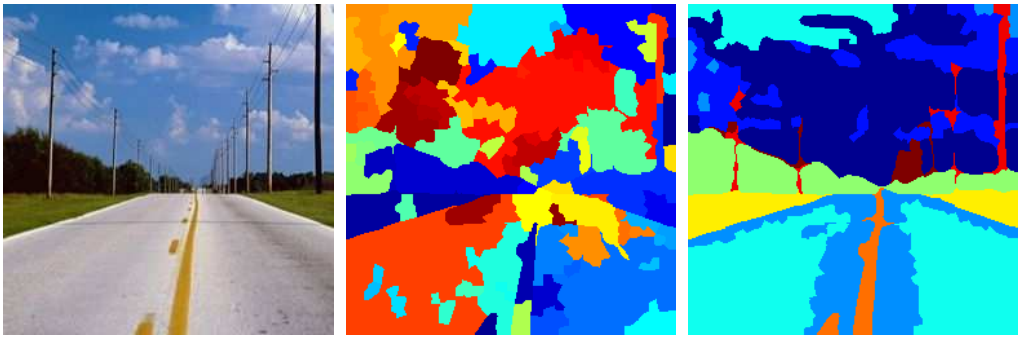

FIGURE 3.1: Comparison of distance-dependent segmentation priors. From left to right, we show segmentations produced by the ddCRP with a = 1,the ddCRP with a = 2, the ddCRP with a = 5, and the rddCRP with a = 1.

3.1 Hierarchical Distance Dependent Chinese Restaurant Process

Informally, the hierarchical ddCRP (hddCRP) applies the the ddCRP formalism twice. Customers link to other customers with probabilities proportional to externally specified distances and form tables. The tables in turn link to other tables with probability proportional to inter-table distances.

Concretely, consider a collection of J images (or restaurants) each containing N_j super-pixels (customers). The i^{th} customer of the j^{th} restaurant is denoted x_{ji} and $X = \{x_{ji} \mid j = 1, \ldots, J, i = 1, \ldots, \mathrm{N}_j\}$. The hierarchical ddCRP is an admixture which clubs customers in a restaurant into tables using a ddCRP, and then groups tables across restaurants using a table level ddCRP.

As in the ddCRP each customer is associated with a link variable c_{ji}. These are however sampled from restaurant specific ddCRPs:

$$p(c_{ji} = j\ell \mid \alpha_j, f_j, D_j) \propto \begin{cases} f_j(d_j(i,\ell)) & i \neq \ell, \\ \alpha_j & i = \ell. \end{cases} \tag{3.1}$$

The decay functions f_j and pairwise distance matrices D_j are restaurant specific. Customers are only allowed to link within restaurants, i.e., $p(c_{ji} = h\ell \mid f_{1:J}, \alpha_{1:J}, D_{1:J}) = 0; \forall j \neq h$. The customer links $C = \{c_{ji} \mid j = 1, \ldots, J, i = 1, \ldots, \mathrm{N}_j\}$ partition X into a set of tables $\mathcal{T}(C)$. For each table $t \in \mathcal{T}(C)$, a table link k_t is sampled from a ddCRP defined over tables:

$$p(k_t = t' \mid \alpha_0, D_0(C)) \propto \begin{cases} f_0(d_0(t,t',C)) & t \neq t', \\ \alpha_0 & t = t'. \end{cases} \tag{3.2}$$

Here, D_0 is the set of pairwise distances between the elements of $\mathcal{T}(C)$ and α_0 is a constant proportional to the self connection probability for table t. The inter table distances $d_0(.)$ can be functions of arbitrary properties of *tables*. Such inter table distances provide a powerful mechanism for capturing *aggregate* properties of groups of data (tables) and allows the modeler to capture high level intuitions about the data such as "segments corresponding to the same object in different video frames should have similar shape and size".

Connected components of the table links $\mathcal{K} = \{k_t \mid t \in \mathcal{T}(C)\}$ then group tables into clusters serving the same dish and determines customer-dish allocations $Z(\mathcal{K}, C) = \{z_{ji} \mid j = 1, \ldots, J, i = 1, \ldots, \mathrm{N}_j\}$ for the entire dataset. Note that $z_{ji} = z_{j'i'}$ iff ji is reachable from $j'i'$ by traversing customer or table links. Next, we endow each dish m with data generating parameters $\phi_m \sim G_0$ from a base distribution G_0 and generate observed data $x_{ji} \sim p(x_{ji} \mid \phi_{z_{ji}})$. We can now state the overall model as follows:

$$p(X, C, \mathcal{K} \mid \alpha_{1:J}, \alpha_0, D_{1:J}, D_0(C), \lambda) = \prod_{j=1}^{J} \prod_{i=1}^{N_j} p(c_{ji} \mid \alpha_j, D_j) \prod_{t \in \mathcal{T}(C)} p(k_t \mid C, \alpha_0, D_0(C)) \prod_{m \in \mathcal{M}(\mathcal{T}(C), \mathcal{K})} p(X_{Z=m} \mid \lambda) \quad (3.3)$$

where $\mathcal{M}(T(C), \mathcal{K})$ represents the set of unique dishes induced by tables $T(C)$ or table links $\mathcal{K}$. The set of all customers sharing a dish m is denoted $X_{Z=m}$ and $p(X_{Z=m} \mid \lambda) = \int p(X_{Z=m} \mid \phi_m) p(\phi_m \mid G_0(\lambda)) d\phi_m$ where, λ represents hyper-parameters governing the base distribution G_0.

Figure 3.2 illustrates the relationship between customer links, tables, table links and dishes. A particular configuration of customer and table links induces a partition of the dataset. The hierarchical ddCRP (hddCRP) is a distribution over the space of these partitions. It places higher probability mass on those partitions which cluster nearby customers into tables *and* nearby tables into dishes.

3.1.1 Related Models

The hddCRP specified above subsumes several related hierarchical models, which are recovered from the ddCRP by appropriately arranging distances between customers and tables. First we note that the Chinese restaurant process (CRP) can be recovered from the ddCRP [5] by arranging customers sequentially and then setting up distances such

that

$$f(d(s,r)) = \begin{cases} 1 & \text{if } s < r \\ 0 & \text{if } s > r \\ \alpha & \text{if } s = r. \end{cases} \tag{3.4}$$

The result follows from observing that the probability of a customer connecting to a customer preceding it in the sequence is proportional to one, thus the probability of a customer sitting at a table with n customers is proportional to n. The probability of sitting at an empty table is α. Also, note that the probability of the partition (seating arrangement) is invariant to the arbitrary sequential arrangement of the data, implying an implicit exchangeability assumption.

This observation leads to the following specializations of the hddCRP:

Chinese Restaurant Franchise. The Chinese restaurant franchise representation of the HDP is recovered from the hddCRP by setting up distances between customers in a restaurant to recover a standard CRP, and by arranging the tables sequentially with inter table distances $f_0(d(t,t',C)) = 1$ for $t < t'$ and $f_0(d_0(t,t',C)) = 0$ for $t > t'$. The CRF assumes both customers and tables are exchangeable.

Other Variants. Along these directions, there are two further specializations of the hddCRP. First, we could use the sequential distance of Equation (3.4) between tables and informed distances between customers, i.e., we could group customers into tables using restaurant specific ddCRPs and then group tables using a traditional CRP [11]. We will explore this model in more detail in Section 3.2 where it is used for image segmentation. Second, we could model customers using a CRP and couple tables using a ddCRP. This was proposed in [29] for time biased topic modeling.

3.2 Image Segmentation with "region" ddCRP

Given an image, represented as a collection of "superpixels" [30] (small blocks of spatially adjacent pixels), our aim is to find segments made up of superpixels homogeneous in appearance *and* whose size statistics loosely match with human annotated segments. In this section, we restrict ourselves to the problem of single image segmentation with J = 1 and drop the explicit dependence on j in our notation.

We consider the hddCRP variant which uses a spatial distance dependent CRP to cluster superpixels (customers) into tables but uses a traditional CRP to group tables into segments. The model allows segments to comprise of several non spatially contiguous tables thus capturing both occlusion and patch redundancy [31] effects commonly observed in

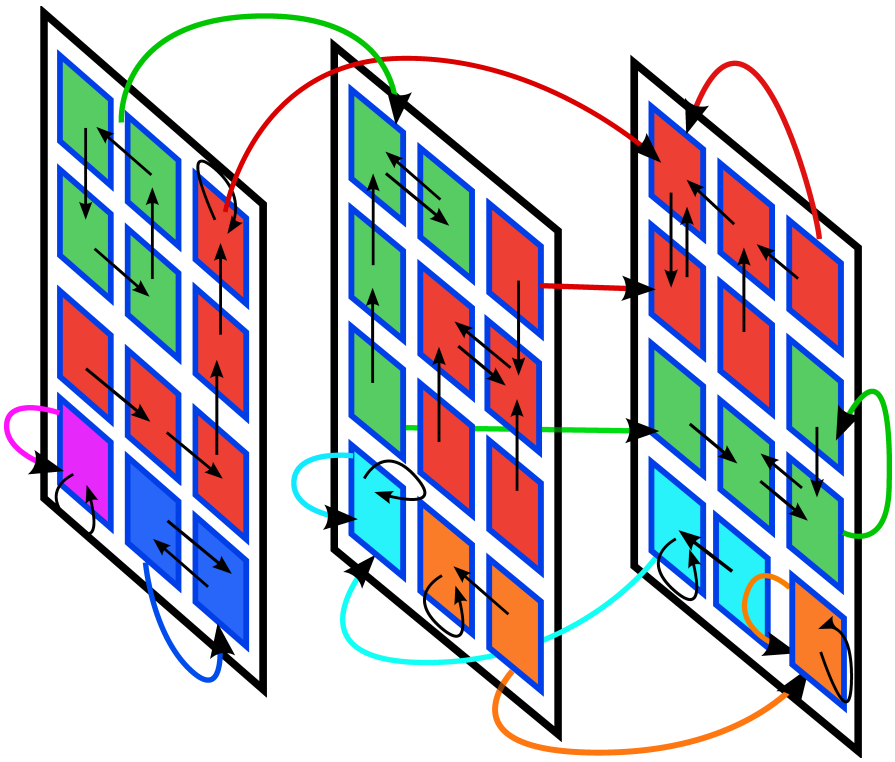

FIGURE 3.2: An illustration of the relationship between the customer links, table links and assignment of data points to dishes in the hddCRP. Three groups containing several squares are shown. Each square is a data point and black arrows between squares are customer links (c_{ji}). Different colors represent different dishes and colored arrows across groups represent table links (k_t).

natural images. Although, we are sacrificing some modeling power by not considering informative inter-table distances, we benefit from the availability of a simpler inference scheme. The model's posterior can be explored through a straightforward extension to the Gibbs sampler presented in Section 2.1.1.

Model and Inference. To capture spatial correlations among image superpixels, we model distances between them as the number of hops required to reach one superpixel from another, with hops being allowed only amongst spatially neighboring superpixels. A "window" decay function of width a, $f(d) = \mathbf{1}_{d \leq a}$ is used. Again, by setting $a = 1$ we can enforce spatial contiguity of tables.

The simple inference scheme alluded to above is only available when the traditional representation of the top level CRP is used as opposed to the customer link representation presented in Equation (3.4). Concretely, this hierarchical ddCRP variant (rddCRP) can be summarized as follows:

- For each superpixel, sample a link c_i according to Equation (2.1), with d and f being hop distances and window decay functions respectively. All superpixel links together partition the image into a set of tables.

- For each table t, sample segment assignments $k_t \sim CRP(\gamma)$.
- For each segment, sample data generating distributions (dishes): $\phi_m \sim G_0(\lambda)$.
- Finally, sample observed data $x_i \sim P(\cdot \mid \phi_{z_i})$, where $z_i = k_{t_i}$

Note that here the top level CRP does not use the link representation. We have slightly abused notation and used k_t to represent links between tables and segments rather than links between two tables as specified in Equation (3.2).

The rddCRP sampler proceeds by cycling through two steps – a) sampling assignments of tables to segments (k_t) from

$$p(k_t = l \mid k_{-t}, X, \mathcal{T}(C), \gamma, \lambda) \propto \begin{cases} m_l^{-t} p(x_t \mid x_{-t}, \lambda) & \text{if } l \text{ is used;} \\ \gamma p(x_t \mid \lambda) & \text{if } l \text{ is new,} \end{cases} \tag{3.5}$$

where x_t is the set of customers sitting at table t, x_{-t} is the set of all customers associated with region l barring x_t, m_l^{-t} is the number of tables associated with region l barring x_t. b) Sampling the customer links c_i. The algorithm for sampling the customer indicators differs from the corresponding ddCRP algorithm in two ways. First, when c_i is removed, it may spawn a new table. In that case, a segment assignment for the new table must be sampled from the region level CRP. Second, the likelihood term in Equation (2.4) now factorizes over tables sharing a common dish. Thus, when computing the ratio in Equation (2.6) all superpixels assigned to a particular dish must be considered, irrespective of their table membership.

3.2.1 Empirical Comparisons.

In this subsection, we perform a controlled comparison of the presented model against standard image segmentation techniques as well as alternative Bayesian nonparametric models proposed for segmentation. The comparison is performed on a collection of images drawn from eight natural scene categories [32]. The collection is available as a subset of the LabelMe [33] dataset.[1] The images come annotated with human segmentations, performed by non-expert users. Following [34], we create a 800 image dataset by randomly selecting 100 images from each category. We compare the segmentations produced by the competing methods against human segmentations via the rand index [**?**]. Qualitative evaluations are provided in figures 3.3 and 3.4

[1] *labelme.csail.mit.edu/browseLabelMe/*

Image Representation. Each image is first divided into approximately 1000 superpixels [30, 35] [2] using the normalized cut algorithm [36].[3] We describe the texture of each superpixel via a local texton histogram [37], using band-pass filter responses quantized to 128 bins. A 120-bin HSV color histogram is used to describe the color of the superpixel. Each superpixel i is summarized via these histograms x_i.

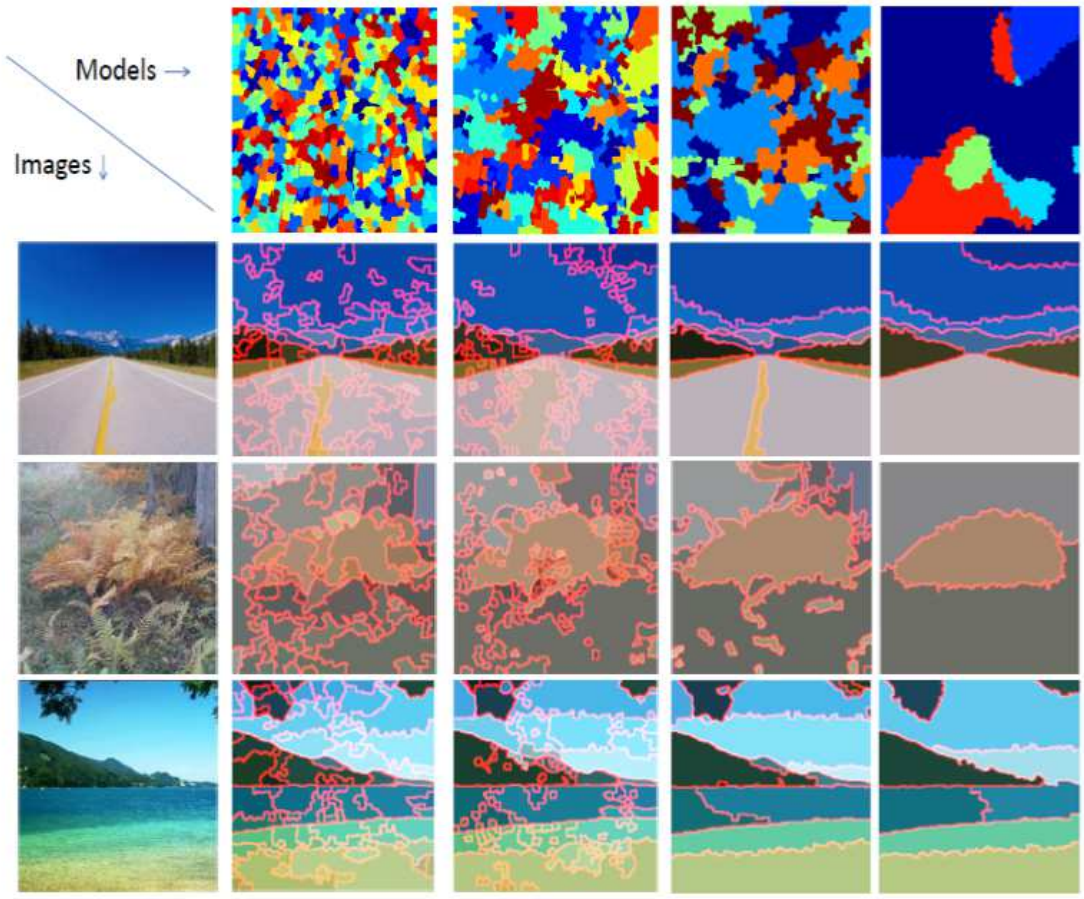

FIGURE 3.3: Segmentations produced by various Bayesian nonparametric methods. From left to right, the columns display natural images, segmentations for the ddCRP with a = 1, the ddCRP with a = 2, the rddCRP with a = 1, and thresholded Gaussian processes (pydist20). The top row displays partitions sampled from the corresponding priors, which have 130, 54, 5, and 6 clusters, respectively.

Sensitivity to Hyperparameters

Our models are governed by the CRP concentration parameters γ and α, the appearance hyperparameter $\lambda = (\lambda_0, ...\lambda_0)$ and the window size a. γ has little impact on the segmentation results, due to the high-dimensional and informative image features. For all our experiments we set γ to 1. α and λ_0 induce opposing biases, a small α encourages larger segments while a large λ_0 encourages larger segments. We found $\alpha = 1e - 8$ and $\lambda_0 = 20$ to work well. The most influential prior parameter is a, the effect of which is

[2] *www.cs.sfu.ca/~mori/*
[3] *www.eecs.berkeley.edu/Research/Projects/CS/vision/*

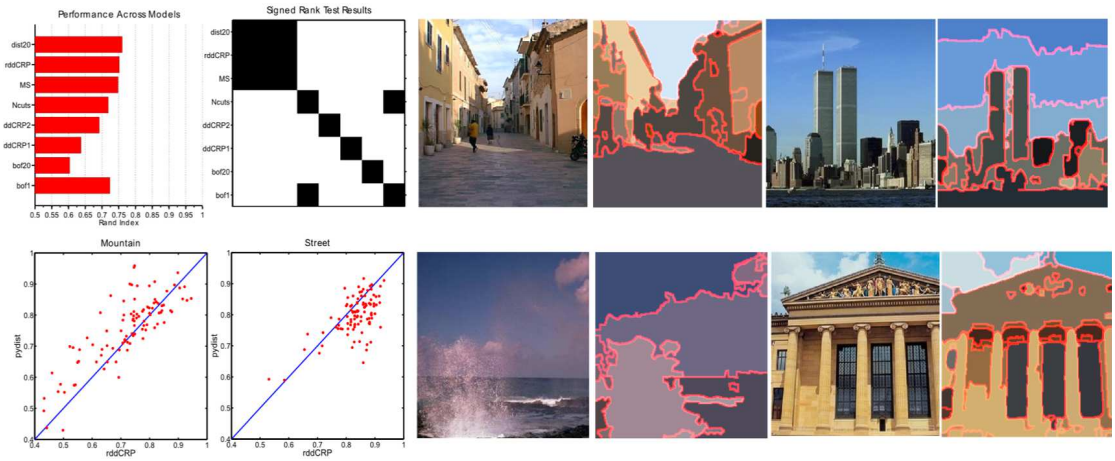

FIGURE 3.4: *Left*: Segmentations produced by rddCRP. *Right (top)*: Average performance across the dataset, as measured by Rand index. rddCRP, pydist20 and Mean Shift are statisticaly indistinguishable and siginficantly better than the rest as determined by a Wilcoxon's signed rank test at 95% confidence. *Right (bottom)*: pydist20 and rddCRP are compared via scatter plots of Rand indexes for the *Mountain* and *Street* categories.

visualized in figure 3.3. For the ddCRP model, setting $a = 1$ (ddCRP1) produces a set of contiguous segments. Increasing to $a = 2$ (ddCRP2), results in fewer segments, but the produced segments are spatially fragmented. The phenomenon is further exacerbated with larger values of a. The rddCRP model groups together segments produced by ddCRP. If the segments produced by ddCRP are poor, rddCRP has a hard time recovering meaningful partitions. Not surprisingly then, rddCRP performs best with $a = 1$.

3.2.2 Image Segmentation Performance

We now quantitatively measure the performance of our models. The ddCRP and the rddCRP samplers were run for 100 and 500 iterations respectively. Both samplers displayed rapid mixing and often stabilized within the first 50 iterations. Note that similar rapid mixing has been observed in other applications of the ddCRP [7].

We also compare to two previous models [34]: a Pitman-Yor mixture model with no spatial dependence (*pybof20*), and the Pitman-Yor mixture with spatial coupling induced via thresholded Gaussian processes (*pydist20*) introduced in Section **??**. To control the comparison as much as possible, the Pitman-Yor models are tested with identical features and base measure, and other hyperparameters as in [34]. We also compare to the non-spatial Pitman-Yor with $\lambda_0 = 1$, the best bag-of-feature model in our experiments (*pybof*). We employ non-hierarchical versions of the Pitman-Yor models, so that each image is analyzed independently, and perform inference via the previously developed

mean field variational method. Finally, we also compare agianst Normalized cuts [38] and Mean shift segmentation [39] techniques[4].

The performance summary is presented in Fig. 3.4. Not surprisingly, rddCRP outscores both versions of the ddCRP model, in terms of Rand index. Nevertheless, the patchy ddCRP1 segmentations are interesting for applications where segmentation is an intermediate step rather than the final goal. The bag of features model with $\lambda_0 = 20$ performs poorly, *pybof* with optimized λ performs reasonably but still falls short of the region level spatial models. The spatial Pitman-Yor model and rddCRP perform similarly, with the Pitman-Yor outperforming rddCRP on some categories and vice versa. The scatter plots in Fig. 3.4 provide insights into when one model outperforms the other. Here, we have plotted the Rand indexes of images from the mountain and street categories. For the street images rddCRP is better, while for images containing mountains spatial PY is superior. In general, street scenes contain more objects, many of which are small, and thus disfavored by the smooth Gaussian processes underlying the PY model. For a fair comparison with hddCRP, we tested a version of the spatial PY model employing a covariance functions dependent only on spatial distance. Such covariances are not flexible enough to allow rapid changes in the GP functions necessary to explain scenes with lots of small objects. Further performance improvements were demonstrated in [34] via a conditionally specified covariance, which depends on detected image boundaries. In Chapter **??** we will further explore this direction and demonstrate that such conditional covariances can be learned from data and improve performance further. Similar conditional specification of the ddCRP distance function is also possible, although based on preliminary experiments have found a smaller benefit from such distances.

3.3 Inference in General Hierarchical ddCRP models

This section develops MCMC based posterior sampling algorithms for the general model presented in Section 3.1 with informative distances both at the table and customer levels.

Here, we will work directly in the link representations, and our latent variables of interest will be the customer links C and table links $\mathcal{K}$. The posterior over them is given by:

$$p(C, \mathcal{K} \mid X, \alpha_{1:J}, \alpha_0, D_{1:J}, D_0(C), \lambda) = \frac{p(C \mid \alpha_{1:J}, D_{1:J}) p(\mathcal{K} \mid C, \alpha_0, D_0(C)) p(X \mid \mathcal{K}, C, \lambda)}{\sum_{C,\mathcal{K}} p(X, C, \mathcal{K} \mid \alpha_{1:J}, \alpha_0, D_{1:J}, D_0(C), \lambda)} \tag{3.6}$$

[4]We used the EDISON implementation of Mean shift. The parameters of Mean shift and Normalized cuts were tuned by performing a grid search over a training set containing 25 images from each of the 8 categories. For normalized cuts the optimal number of segments was determined to be 5, for mean shift we held the spatial bandwidth constant at 7 and found optimal values of feature bandwidth and minimum region size to be 25 and 4000 (pixels) respectively.

Similar to the inference for the image segmentation model, we will explore the posterior distribution by iteratively sampling customer links given table links and vice versa. However, in the general model coupling between customer and table links causes the inference to be more involved. As before, changing a customer link may cause tables to split and/or merge necessitating a change to the table links of the affected tables as well. Since, in the link representation tables point to each other rather than to dishes, in addition to modifying the links of tables being merged or split we must also change links of all tables pointing into these tables. We use a Metropolis Hastings (MH) proposal to make such coordinated changes to the customer and affected table links.

In the remainder of the section we restrict our description of the algorithm to a particular group j and drop the explicit dependence on j from the notation. We proceed by proposing a customer link from a distribution $q(c_i)$. We consider two candidate distributions one that proposes links from the prior $q(c_i^*) = p(c_i^* \mid \alpha, D)$ and another that proposes links from a discrete distribution that conditions on observations $\mathbf{X}$.

$$q(c_i^*) \propto p(c_i^* \mid \alpha, D)\Gamma(\mathbf{X}, \mathbf{z}, \lambda),$$
$$\Gamma(\mathbf{X}, \mathbf{z}, \lambda) = \begin{cases} \dfrac{p(\mathbf{X}_{\mathbf{z}(\Delta)=m_a} \cup \mathbf{X}_{\mathbf{z}(\Delta)=m_b} \mid \lambda)}{p(\mathbf{X}_{\mathbf{z}(\Delta)=m_a} \mid \lambda)p(\mathbf{X}_{\mathbf{z}(\Delta)=m_b} \mid \lambda)} & \text{if } c_i^* \text{ merges dishes } m_a \text{ and } m_b \\ 1 & \text{otherwise.} \end{cases} \quad (3.7)$$

of all table links excluding k_{t_i} and t_i denotes the table assignment of customer i. We refer to the first proposal as the prior proposal. Such proposals, although naive, perform reasonably well when D is sparse. The data dependent proposal is termed the pseudo Gibbs proposal. Since it conditions on data ($\mathbf{X}$) and the current state of the sampler, we expect it to be more effective than the prior proposal.

Once a customer link is proposed, depending on the change to the table structure, we may need to reconfigure table links. If the new customer link c_i^* points to a customer sharing a table with i, the partition remains unchanged and no table links need to be changed. If on the other hand, c_i^* points to a customer i^* assigned to another table, a merge occurs. We then delete the table link k_{t_i} and retain $k_{t_{i^*}}$ as the link of the merged table $k_{t_{i,i^*}}$. Finally, all tables pointing to t_i are reassigned to the merged table t_{i,i^*}. Alternatively, c_i^* may cause an existing table ($t_{i,i'}$, where i' is the old assignment of c_i) to split in two. In this case, tables pointing to $t_{i,i'}$ are reassigned to the split tables. The link $k_{t_{i,i'}}$ is retained by $t_{i'}$ and a new link k_{t_i} is sampled. Figure 3.5 illustrates these moves, note that the splits and merges are reverses of each other. A customer link change might also cause a table to split, then merge with another table, causing customers to *shift* between tables. In such cases, the affected table links are reassigned by sequentially combining the updates required for the split and merge moves. Finally, the proposed

c_i^* and set of changed table links ($\mathcal{K}^*_{c_i^*}$) are accepted with probability proportional to $\min(1, \rho)$ with :

$$\rho = \frac{p(X, C^*, \mathcal{K}^*)}{p(X, C, \mathcal{K})} \frac{q_{rev}(C, \mathcal{K} \mid C^*, \mathcal{K}^*, X)}{q_{fwd}(C^*, \mathcal{K}^* \mid C, \mathcal{K}, X)} \tag{3.8}$$

where $C^* = \{c_1 \dots c_{i-1}, c_i^*, c_{i+1} \dots c_N\}$ and $\mathcal{K}^* = \{\mathcal{K}^*_{c_i^*}, \mathcal{K}_{-\mathcal{K}^*_{c_i^*}}\}$. After iteratively sampling all customer links and affected table links, we use a Gibbs step to resample *all* table links $\mathcal{K}$. Conditioned on C^*, sampling of $\mathcal{K}$ proceeds in an analogous fashion to customer link sampling in the standard ddCRP inference. Details of the algorithm can be found in Appendix B.

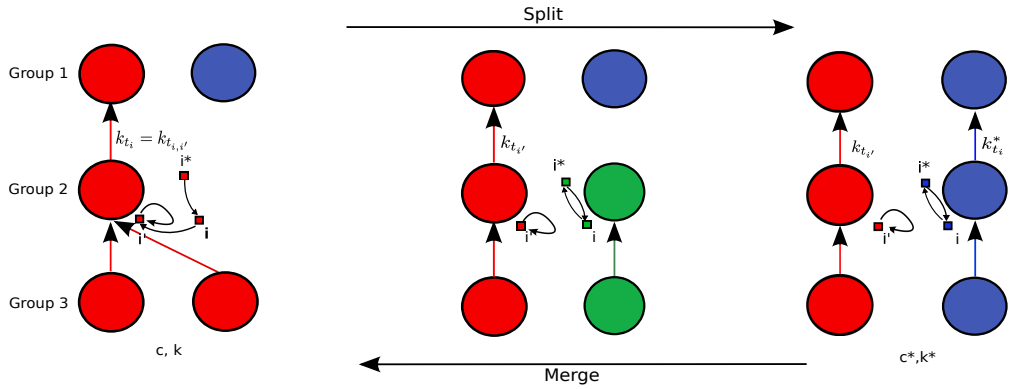

FIGURE 3.5: Illustration of changes induced by a customer link change. Tables are displayed as circles and customer as squares. Colors represent dishes being served at tables. A split involves reassigning the incoming table links and resampling a table link for the newly created table. The dashed ellipse represents a table formed by merging two existing tables. Observe that splitting a table may cause several tables to change dish memberships.

Like split-merge MCMC algorithms [40], our algorithm explores the posterior over the partition by making large moves in the partition space. However, instead of explicitly constructing such moves we propose local changes to the link structure which potentially induce large changes to the partition. Reasoning solely about links, our algorithm is able to make complicated changes to the partition at *different resolutions*, perturbing both the clustering of customers into tables and tables into dishes.

3.4 Experiments

This section has two primary goals – to demonstrate the hddCRP as an effective model for grouped non exchangeable data and to measure the efficacy of the proposed inference algorithm. On the modeling front, we explore the effect of using different types of inter-table distances on toy data and with reasonable distance choices demonstrate competitive performance on the task of video segmentation. On the inference front, we perform a controlled comparison of the two customer link proposals. We further validate

the model and inference via agreement with held out human annotations. For the video segmentation task, agreement is measured through the Rand index [41].

Toy Data. We consider two toy datasets each containing four groups (Figure 3.6). Each group is a 30×30 image containing several objects. The first contains a blue object and two green objects of different sizes. The second dataset emulates a four frame video containing objects moving from top to bottom at different rates. Furthermore, each object exhibits a color gradient rather than a constant color. We compare the hCRP against two versions of the hddCRP model. Both versions use a customer distance which allows pixels to connect to one of their eight neighbors with equal probability. The table distances vary between the two versions. For modeling the data observed in the first dataset we use the difference in table *sizes* as the inter table distances. As seen in Figure 3.6, conditioned on table sizes the hddCRP distinguishes the two green clusters, while the hCRP groups them together. The distinguishing characteristic of the objects in the second dataset is motion. We use optical flow[5] between frames to define inter table distances. Let $w(t)$ denote the position of table t after being warped by optical flow. The decay function modulated distance between t and t' is then defined to be: $f_0(d_0(t,t',C)) = \dfrac{w(t) \cap t'}{w(t) \cup t'}$ $;t \neq t'$, This distance encourages t to link to t' if $w(t)$ has a high overlap with t'. Such distances are popular in the object recognition and detection communities [42]. Using these inter table distances, hddCRP distinguishes four uniquely moving objects in the video. hCRP, which doesn't account for spatial and motion smoothness, produces a noisy segmentation and confuses differently moving objects of similar appearance.

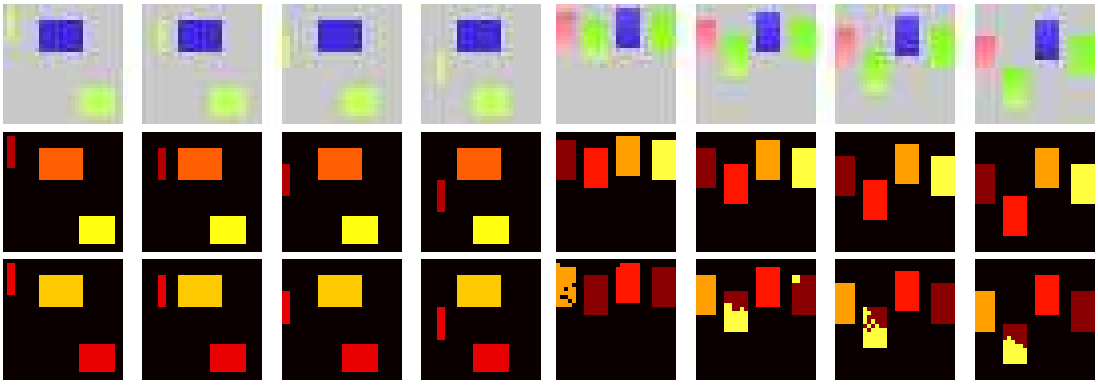

FIGURE 3.6: Illustration of table distances. **Top Row**. Ground truth partitions of two toy datasets each containing four groups. *Left.* Toy data exhibits objects of similar appearance but widely varying sizes. *Right.* Objects exhibit motion and color gradients. **Middle Row**. MAP partitions inferred by hddCRP using size and optical flow based inter table distances. **Bottom row**. MAP partitions discovered by hCRP.

[5]Note since this is a toy dataset, we have access to the true optical flow.

Comparison of customer link proposals. Here, we compare the prior and pseudo Gibbs customer link proposals: $q(c_i^* \mid \alpha, D)$ and $q(c_i^* \mid \mathbf{X}, \mathcal{K}_{-t_i}, C_{-i})$. Figure 3.7 displays 20 pairs of MCMC chains run for several hundred iterations on 10 video frames extracted from the classic "garden" sequence. The pseudo Gibbs proposals clearly outperform the prior proposal. We also display partitions with the highest and lowest log-likelihoods (after discarding the first 75 burn in samples) discovered by the two proposals for a qualitative comparison. The pseudo Gibbs proposals produce qualitatively superior partitions and exhibit lower variation between the best and the worst partitions.

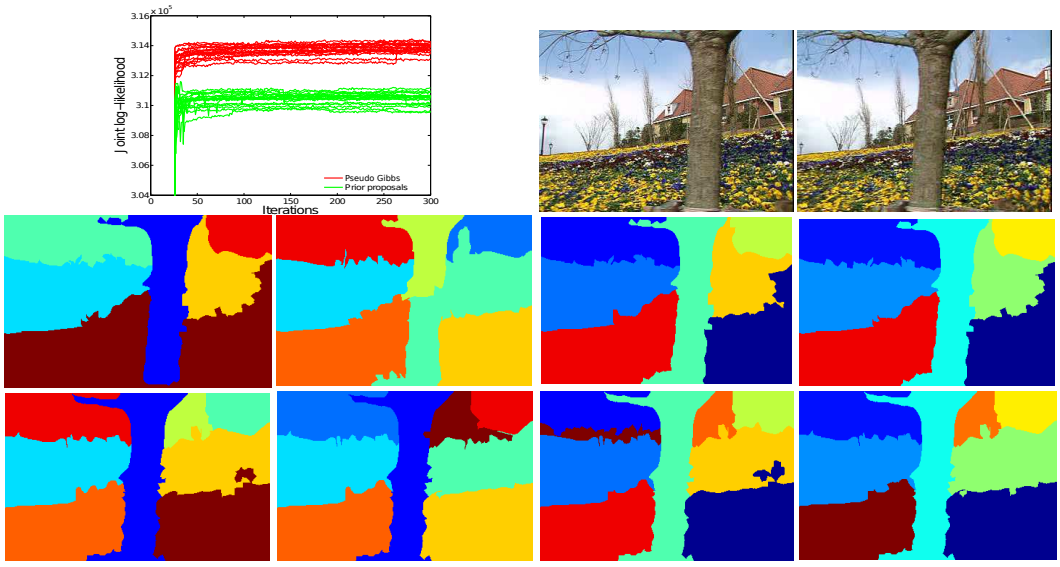

FIGURE 3.7: Customer link proposal comparison. **Top Row**. Joint log-likelihoods across iterations, followed by fifth and tenth frames of the sequence. **Middle Row** (*Left to Right*). Best (MAP) and worst 5^{th} frame partitions discovered by the prior and pseudo Gibbs proposals respectively. **Bottom Row**. Partitions of the 10^{th} frame.

3.4.1 Video Segmentation

We use the MIT human annotated video dataset [43] for assessing performance on the video segmentation task. The dataset consists of 9 human annotated videos and we benchmark performance on the first 10 frames of each video. Following Section 3.2 each frame is divided into approximately 1100 superpixels [44, 45] [6] using the normalized cut algorithm [46].

Likelihoods. Each super-pixel is described using L_2 *unit normalized* 120-bin HSV color and 128-bin local texton histograms [47]. The unit-normalization projects the raw histogram counts to the surface of a hyper-sphere, which are then modeled using von-Mises

[6] *www.cs.sfu.ca/~mori/*

Fisher distributions shared across all tables of a dish. Through preliminary experiments, we found that von-Mises Fisher distributions [48] produced better results than modeling the raw histograms using Multinomial distributions. Similar L_2 normalizations have previously been found to be useful in image retrieval literature [49]. Additionally, we also extract optical flow using "Classic+NL" [50], and associate a two dimensional flow vector to each super-pixel, the median flow of its constituent pixels. The flow vectors are then modeled using Normal inverse-Wishart distributions. A super-pixel i in video frame j is then given by $x_{ji} = \{x^c_{ji}, x^t_{ji}, x^f_{ji}\}$ with

$$x^c_{ji} \sim \text{vMF}(\mu^c_{z_{ji}}, \kappa^c); \mu^c_{z_{ji}} \sim \text{vMF}(\mu^c_0, \kappa^c_0) \tag{3.9}$$

where κ^c, μ^c_0 and κ^c_0 are hyper-parameters controlling the concentration of color features around the direction $\mu^c_{z_{ji}}$, the mean color direction (μ^c_0) and the concentration of $\mu^c_{z_{ji}}$ around μ^c_0 respectively. Texture features are generated analogously. The flow features are modeled using Normal inverse Wishart distributions:

$$x^f_{ji} \sim \mathcal{N}(\mu^j_{z_{ji}}, \Sigma^j_{z_{ji}});\ \Sigma^j_{z_{ji}} \sim \mathcal{IW}(n_0, S_0);\ \mu^j_{z_{ji}} \mid \Sigma^j_{z_{ji}} \sim \mathcal{N}(\mu_0, \tau_0 \Sigma^j_{z_{ji}}) \tag{3.10}$$

Requiring all tables in a dish to share the same flow model is too restrictive. Instead, we model the flow for each frame independently and require that all tables in frame j assigned to dish z_{ji} share the same flow model. If a dish spans F frames, then we endow it with F independent flow models, requiring motion within a frame to be coherent but not restricting motion between frames. See the supplement, for specific hyper-parameter settings.

Prior. The general hddCRP prior requires the specification of distances between customers as well as tables. Following our image segmentation work (Section 3.2), we again use hop distances with window decay functions as the between superpixel distances. The flow based distances defined in equation 3.4 are used as the inter table distances. All $\alpha_{1:J}$ and α_0 were set to 10^{-8}.

Controlled comparisons. We benchmark our video segmentation performance by comparing against a popular non probabilistic hierarchical graph based video segmentation (HGVS) algorithm [51][7], a hierarchical ddCRP (limited-ddCRP) variant, recently proposed for performing video co-segmentation [52], which ignores distances among tables and against the hCRP [53]. For a fair comparison, we augmented the limited-ddCRP model presented in [52] with the appearance and motion likelihoods described above.

[7] *http://neumann.cc.gt.atl.ga.us/segmentation/*

Figure 3.8 provides a comparison of the results produced by the different methods. For HGVS the displayed segmentations were produced at 90 percent of highest hierarchy level, which appears to produce the best visual and quantitative results. For the hddCRP variants, the produced segmentation corresponds to the MAP sample of five MCMC chains. The results indicate that hddCRP clearly produces better results than HGVS both qualitatively and in terms of rand index. Given the strong local likelihoods, the gains over hCRP and limited-ddCRP are more modest. Nonetheless, modeling dependencies between pixels and tables refines the segmentations and provides a small quantitative performance boost and a larger qualitative improvement.

3.5 Discussion

In this chapter, we have developed hierarchical extensions to the ddCRP as well as MCMC inference algorithms for effective inference on these models. We have found these models to be better suited for image and video segmentations. For both these tasks the hierarchical models significantly outperform the vanilla ddCRP mixtures and achieve performance competitive with the state-of-the-art.

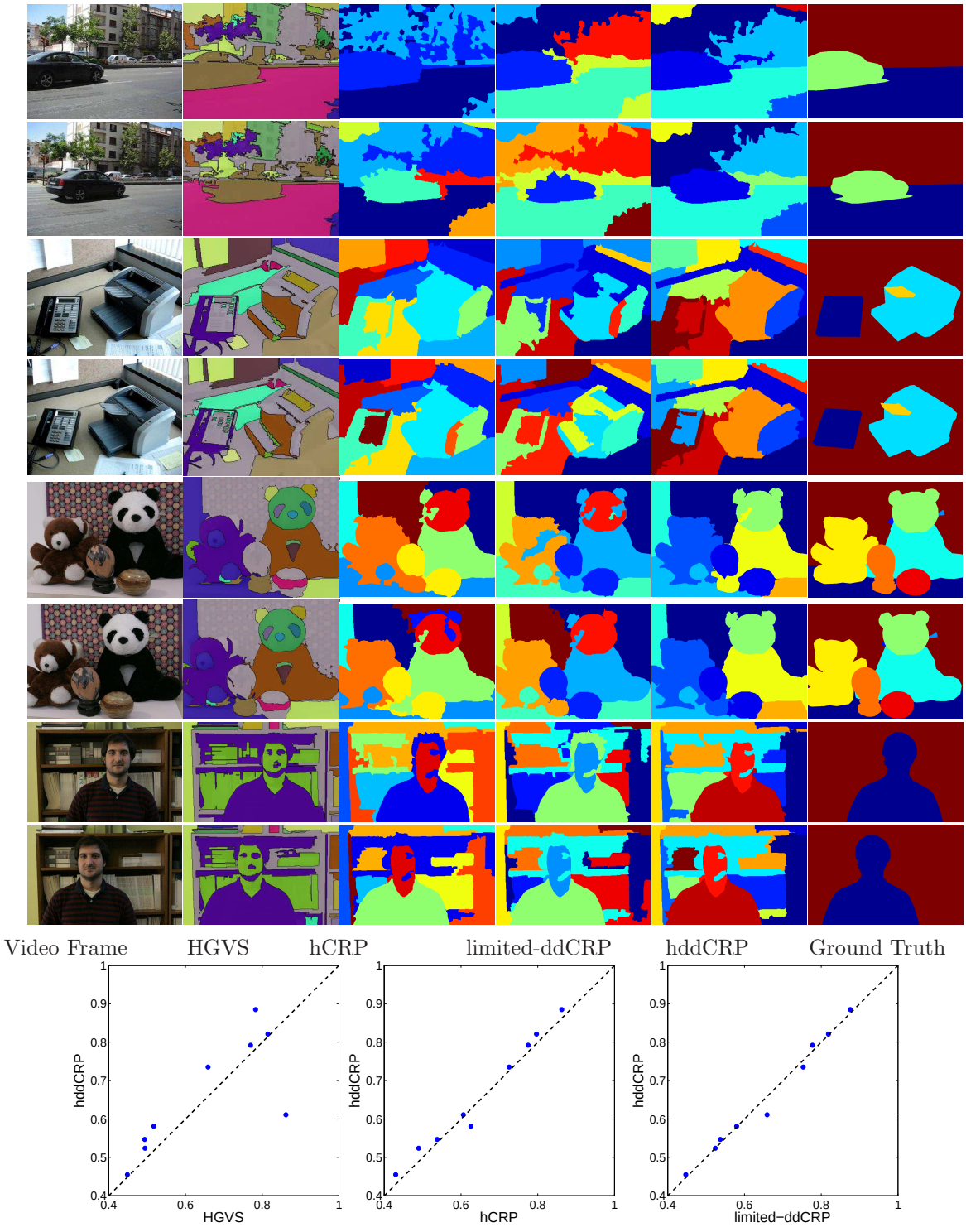

FIGURE 3.8: Video segmentation results. The top eight rows show the first and tenth frames of four videos from the MIT dataset. From *left to right* we have the original video frames, segmentations produced by HGVS, CRF, limited-ddCRP and hddCRP and the ground truth segmentations. The last row displays scatter plots comparing hddCRP, HGVS, limited-ddCRP and hCRP in terms of rand index achieved on all nine human annotated videos.

Chapter 4

Spatially Dependent Pitman-Yor processes

4.1 Introduction

Image segmentation algorithms partition images into spatially coherent, approximately homogeneous regions. Segmentations provide an important mid-level representation which can be leveraged for various vision tasks including object recognition [54], motion estimation [50], and image retrieval [55]. Despite significant research [56–60], segmentation remains a largely unsolved problem. One major challenge is to move beyond seeking a single "optimal" image partition, and to recognize that while there are commonalities among multiple human segmentations of the same image, there is also substantial variability [61].

Most existing segmentation algorithms are endowed with a host of tunable parameters; a particular configuration may work well on some images, and poorly on others. Often these parameters are tuned via manual experimentation, or expensive validation experiments. Noting this issue, Russell et al. [62] produced a "soup of segments" by varying the parameters of the normalized cuts algorithm, and collecting the range of observed outputs. Others have used agglomerative clustering methods to produce a nested tree of segmentations [60]. A limitation of these approaches is that they do not provide any image-specific estimate of which particular segmentations are most accurate.

In this paper, we instead pursue a Bayesian nonparametric statistical approach to modeling segmentation uncertainty. We reason about prior and posterior distributions on the space of image partitions, and thus consider segmentations of all possible resolutions. In contrast with parametric segmentation models based on finite mixtures [55, 63, 64] or

Markov random fields [65], we do *not* need to pre-specify the number of segments. Our inference algorithm automatically provides calibrated estimates of the relative probabilities of segmentations with varying numbers of regions.

Because we define a consistent probabilistic model and not just a segmentation procedure, our approach is a natural building block for more sophisticated models. We improve earlier work on spatially dependent Pitman-Yor (PY) processes [3], which was motivated by the problem of jointly segmenting multiple related images. This PY model was later extended to allow prediction of semantic segment labels, given supervised annotations of objects in training images [66]. Here we focus on the problem of segmenting single images containing unknown object categories.

The model we consider is a minor variation on the dependent PY process of Sudderth and Jordan [3], which captures the power law distribution of human image segments via a stick-breaking construction, and uses Gaussian processes (GPs) to induce spatial dependence. Our first major contribution is a new posterior inference algorithm that is far less susceptible to local optima than previous mean field variational methods [3]. Our algorithm combines a discrete stochastic search, capable of making large moves in the space of image partitions, with an accurate higher-order variational approximation (based on expectation propagation [67]) to marginalize latent GPs. We improve computational efficiency via a low rank representation of the GP covariance, an innovation that could be applicable to many other models with high-dimensional Gaussian variables.

Our second major contribution is a procedure for learning the various model hyperparameters, including image-dependent GP covariance functions, from example human segmentations. Using training images from the Berkeley segmentation dataset [61], we calibrate our model, and then evaluate its accuracy in segmenting various images of natural scenes [61, 68]. Our results show significant improvements over prior work with PY process models [3], and demonstrate segmentations that are both qualitatively and quantitatively competitive with state-of-the-art methods.

4.2 Nonparametric Bayesian Segmentation

We have two primary requirements of any segmentation model – a) it should adapt to image complexity and automatically select the appropriate number of segments and b) it should encourage spatial neighbors to cluster together. Furthermore, human segmentations of natural scenes consist of segments of widely varying sizes. It has been observed that histograms over segment areas [61] and contour lengths [69] are well explained by power law distributions. Thus a third requirement is to model this power-law behavior.

In this section, we first describe our image representation and then review increasingly sophisticated models which satisfy these requirements. Finally, in Sec. 4.2.4, we propose a novel low-rank model which improves computational efficiency while retaining the above desiderata .

4.2.1 Image Representation

Each image is dicided into roughly 1,000 *superpixels* [70] using the normalized cuts spectral clustering algorithm [56]. The color of each superpixel is described using a histogram of HSV color values with $W_c = 120$ bins. We choose a non-regular quantization to more coarsely group low saturation values. Similarly, the texture of each superpixel is modeled via a local $W_t = 128$ bin texton histogram [71], using quantized band-pass filter responses. Superpixel n is then represented by histograms $x_n = (x_n^t, x_n^c)$ indicating its texture x_n^t and color x_n^c.

4.2.2 Pitman-Yor Mixture Models

Pitman-Yor mixture models extend traditional finite mixture models by defining a Pitman-Yor (PY) process [72] prior over the distribution of mixture components. The distributions sampled from a PY process are countably infinite discrete distributions which place mass on infinitely many mixture components. Furthermore, these discrete distributions follow a power law distribution and previous work [3] has shown that they model the distribution over human segment sizes well. There are various ways of formally defining the PY process, here we consider the stick breaking representation. Let $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \ldots)$, $\sum_{k=1}^{\infty} \pi_k = 1$, denote an infinite *partition* of a unit area region (in our case, an image). The Pitman-Yor process defines a prior distribution on this partition via the following *stick-breaking* construction:

$$\begin{aligned} \pi_k &= w_k \prod_{\ell=1}^{k-1} (1 - w_\ell) = w_k \left(1 - \sum_{\ell=1}^{k-1} \pi_\ell\right) \\ w_k &\sim \text{Beta}(1 - \alpha_a, \alpha_b + k\alpha_a) \end{aligned} \tag{4.1}$$

This distribution, denoted by $\boldsymbol{\pi} \sim \text{GEM}(\alpha_a, \alpha_b)$, is defined by two hyperparameters (the discount and the concentration parameters) satisfying $0 \leq \alpha_a < 1$, $\alpha_b > -\alpha_a$. It can be shown that $\mathbb{E}\pi_k \propto k^{-1/\alpha_a}$, thus exhibiting the aforementioned power law distribution.

For image segmentation, each index k is associated with a different segment or region with its own appearance models $\theta_k = (\theta_k^t, \theta_k^c)$ parameterized by multinomial distributions on the W_t texture and W_c color bins, respectively. Each superpixel n then independently

selects a region $z_n \sim \text{Mult}(\boldsymbol{\pi})$, and a set of quantized color and texture responses according to

$$p(x_n^t, x_n^c \mid z_n, \boldsymbol{\theta}) = \text{Mult}(x_n^t \mid \theta_{z_n}^t, M_n)\,\text{Mult}(x_n^c \mid \theta_{z_n}^c, M_n) \tag{4.2}$$

The multinomial distributions themselves are drawn from a symmetric Dirichlet prior with hyper-paramter ρ. Note that conditioned on the region assignment z_n, the color and texture features for each of the M_n pixels within superpixel n are sampled independently. The appearance feature channels provide weak cues for grouping superpixels into regions. Since, the model doesn't enforce any spatial neighborhood cues, we refer to it as the "bag of features" (BOF) model.

4.2.3 Spatially Dependent PY Mixtures

Next, we review the approach of Sudderth and Jordan [3] which extends the BOF model with spatial grouping cues. The model combines the BOF model with ideas from layered models of image sequences [73], and level set representations for segment boundaries [74].

We begin by elucidating the analogy between PY processes and layered image models. Consider the PY stick-breaking representation of Eq. (4.1). If we sample a random variable z_n such that $z_n \sim \text{Mult}(\boldsymbol{\pi})$ where $\pi_k = w_k \prod_{\ell=1}^{k-1}(1 - w_\ell)$, it immediately follows that $w_k = \mathbb{P}[z_n = k \mid z_n \neq k-1, \ldots, 1]$. The stick-breaking proportion w_k is thus the *conditional* probability of choosing segment k, given that segments with indexes $\ell < k$ have been rejected. If we further interpret the ordered PY segments $\{k = 1, \ldots \infty\}$ as a sequence of layers, z_n can be sampled by proceeding through the layers in order, flipping biased coins (with probabilities w_k) until a layer is chosen. Given this, the probability of assignment to subsequent layers is zero; they are effectively *occluded* by the chosen "foreground" layer.

The spatially dependent Pitman-Yor process of [3] preserves this PY construction, while adding spatial dependence among super-pixels by associating a layer (real valued function) drawn from a zero mean *Gaussian process* (GP) $\mathbf{u}_k \sim GP(\mathbf{0}, \Sigma)$ with each segment k. Σ captures the spatial correlation amongst super-pixels, and without loss of generality we assume that it has a unit diagonal. Each super-pixel can now be associated with a layer following the procedure described in the previous paragraph, n.e.,

$$z_n = \min\left\{k \mid u_{kn} < \Phi^{-1}(w_k)\right\},\ u_{kn} \sim \mathcal{N}(0, \Sigma_{nn} = 1) \tag{4.3}$$

Here, $u_{kn} \perp u_{\ell n}$ for $k \neq \ell$ and $\Phi(u)$ is the standard normal *cumulative distribution function* (CDF). Let $\delta_k = \Phi^{-1}(w_k)$ denote a threshold for layer k. Since $\Phi(u_{kn})$ is

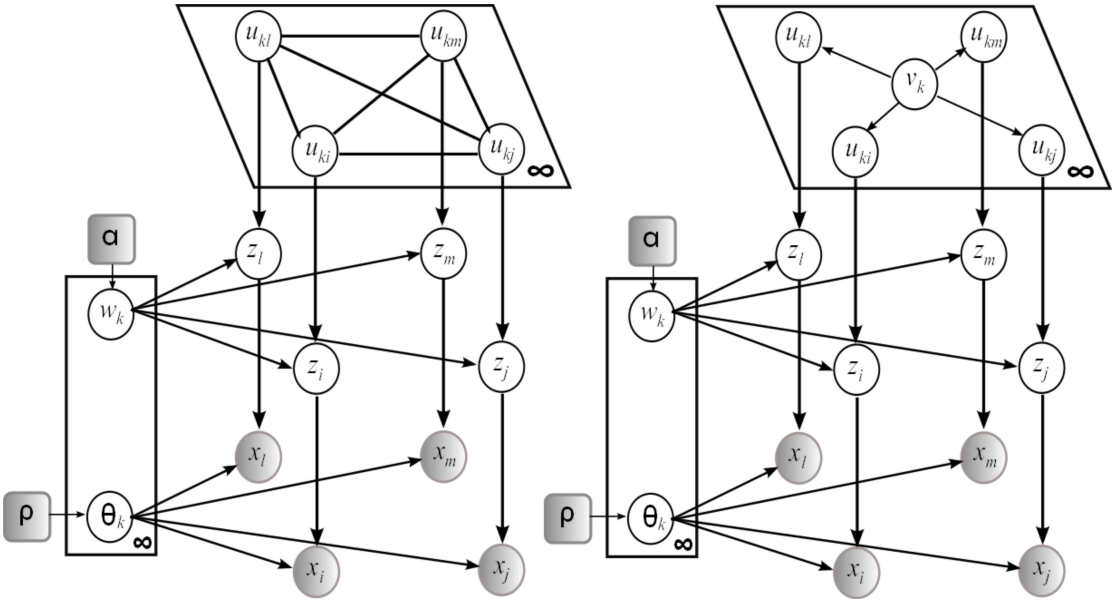

FIGURE 4.1: **Generative models of image partitions.** *Left.* Spatially dependent PY model, *(right)* low rank model. Shaded nodes represent observed random variables. $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$ is a low dimensional Gaussian random variable and $\mathbf{u}_k$ is the corresponding N dimensional layer. $w_k \sim \text{Beta}(1-\alpha_a, \alpha_b + k\alpha_a)$ controls expected layer size and are governed by Pitman-Yor hyper-parameters $\alpha = (\alpha_a, \alpha_b)$. The Dirichlet hyper-parameters $\rho = (\rho^t, \rho^c)$ parametrize appearance distributions. Finally, the color and texture histograms describing super-pixel n are represented as $x_n = (x_n^t, x_n^c)$

uniformly distributed on $[0, 1]$, we have

$$\begin{aligned} \mathbb{P}z_n = 1 &= \mathbb{P}u_{1n} < \delta_1 = \mathbb{P}\Phi(u_{1n}) < w_1 = w_1 = \pi_1 \\ \mathbb{P}z_n = 2 &= \mathbb{P}u_{1n} > \delta_1 \mathbb{P}u_{2n} < \delta_2 = (1-w_1)w_2 = \pi_2 \end{aligned} \tag{4.4}$$

and so on. The extent of each layer is determined via the region on which a real-valued function lies below the threshold δ_{layer}, akin to level set methods. If $\Sigma = \mathbf{I}$, we recover the BOF model. More general covariances can be used to encode the prior probability that each feature pair occupies the same segment; developing methods for learning these probabilities is a major contribution of this paper.

The power law prior on segment sizes is retained by transforming priors on stick proportions $w_k \sim \text{Beta}(1-\alpha_a, \alpha_b + k\alpha_a)$ into corresponding randomly distributed thresholds $\delta_k = \Phi^{-1}(w_k)$:

$$p(\delta_k \mid \alpha) = \mathcal{N}(\delta_k \mid 0, 1) \cdot \text{Beta}(\Phi(\delta_k) \mid 1-\alpha_a, \alpha_b + k\alpha_a) \tag{4.5}$$

Figure 4.1 displays corresponding graphical model. Image features are generated as in the BOF model.

4.2.4 Low-Rank Representation

In the preceding generative model, the layer support functions $\mathbf{u}_k \sim \mathcal{N}(0, \Sigma)$ are samples from a Gaussian distribution over N super-pixels. Inference involving GPs involve inverting Σ which is in general a $O(N^3)$ operation and thus scales poorly with increasing image sizes. To cope, we employ a low-rank representation based on $D \leq N$ dimensions, analogous to factor analysis models. We proceed by defining a Gaussian distributed D dimensional latent variable $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$, we then set $\mathbf{u}_k = A\mathbf{v}_k + \epsilon_k$, where A is a N-by-D dimensional factor loading matrix and $\epsilon_k \sim \mathcal{N}(0, \Psi)$, with Ψ being a diagonal matrix. Observe that marginalizing over $\mathbf{v}_k$ results in a model equivalent to the full rank model of the preceding section with $\Sigma = AA^T + \Psi$. The low rank model replaces the $O(N^3)$ operation with an $O(ND^2)$ operation, thus scaling linearly with N[1]. Figure 4.1 displays the corresponding graphical model.

4.3 Inference

This section describes a novel, robust to local optima, inference algorithm which is an example of a Maximization Expectation (ME) [75] technique. In contrast to the popular Expectation Maximization algorithms, ME algorithms marginalize model parameters and directly maximize over the latent variables. In our model, the latent variables correspond to segment assignments of super-pixels (z_n). Any configuration of these variables defines a partition of the image. Our strategy is to explore the space of these image partitions by climbing the posterior $p(\mathbf{z} \mid \mathbf{x}, \eta)$ surface, where $\eta = \{\alpha, \rho, A, \Psi\}$. It is worth noting that since different partitions will have different numbers of segments, we are in fact searching over models of varying complexities akin to traditional model selection techniques.

The algorithm proceeds by first evaluating the posterior for an initial image partition $\mathbf{z}$. It then modifies the partition in an interesting fashion to generate a new partition $\mathbf{z}'$ which is accepted if $p(\mathbf{z}' \mid \mathbf{x}, \eta) \geq p(\mathbf{z} \mid \mathbf{x}, \eta)$. This process is repeated until convergence. By caching the various mutated partitions, we approximate the posterior distribution over partitions (Figure 4.5). In what follows, we first describe the innovations required for evaluating the posterior marginal and then the procedure for mutating a partition.

[1] A complete time complexity analysis is available in the supplement.

4.3.1 Posterior Evaluation

In our model (*Figure 4.1*), the posterior $p\left(\mathbf{z} \mid \mathbf{x}, \eta\right)$ factorizes as $p\left(\mathbf{z} \mid \mathbf{x}, \eta\right) \propto p\left(\mathbf{x} \mid \mathbf{z}, \rho\right) p\left(\mathbf{z} \mid \alpha, A, \Psi\right)$. The likelihood:

$$p\left(\mathbf{x} \mid \mathbf{z}, \rho\right) = \int_{\Theta} p\left(\mathbf{x} \mid \mathbf{z}, \Theta\right) p\left(\Theta \mid \rho\right) d\Theta \tag{4.6}$$

is a standard Dirichlet-multinomial integral and can be evaluated in closed form[2].

Unfortunately, the prior can't similarly be evaluated in closed form. Significant innovations are required for its computation and the remainder of this section details a major contribution of this paper, an algorithm for evaluating $p\left(\mathbf{z} \mid \eta\right)$.

$$\begin{aligned} p\left(\mathbf{z} \mid \eta\right) = \prod_{k=1}^{K(\mathbf{z})} \int_{\mathbf{u}_k} \int_{\delta_k} \int_{\mathbf{v}_k} & p\left(\mathbf{z} \mid \delta_k, \mathbf{u}_k\right) \\ & p\left(\mathbf{u}_k, \mathbf{v}_k \mid A, \Psi\right) p\left(\delta_k \mid \alpha\right) d\mathbf{v}_k d\mathbf{u}_k d\delta_k \end{aligned} \tag{4.7}$$

where $K(\mathbf{z})$ represents the number of layers in partition $\mathbf{z}$. To simplify notation in the remainder of this paper we denote $K(\mathbf{z})$ simply by K. Note that in the BOF model $\mathbf{z}$ depends only on α and $p(\mathbf{z}|\alpha)$ can be calculated in closed form:

$$p(\mathbf{z} \mid \alpha) = \alpha_a^K \frac{\Gamma\left(\alpha_b/\alpha_a + K\right)\Gamma(\alpha_b)}{\Gamma(\alpha_b/\alpha_a)\Gamma(N + \alpha_a)} \left(\prod_{k=1}^{K} \frac{\Gamma(M_k - \alpha_a)}{\Gamma(1 - \alpha_a)} \right) \tag{4.8}$$

where N is the number of super-pixels in the partition and M_k is the number of super-pixels in layer k.

Spatial prior evaluation. The integrals in equation 4.7 can be evaluated independently for each layer. In the following analysis, it is implied that we are dealing with the k^{th} layer and we drop the explicit dependence on k in our notation. We approximate the joint distribution $p(\mathbf{u}, \mathbf{v}, \delta, z \mid \eta)$ with a Gaussian distribution $q(\mathbf{u}, \mathbf{v}, \delta, \mathbf{z} \mid \eta)$ and the corresponding marginal $p(\mathbf{z} \mid \eta)$ with $q(\mathbf{z} \mid \eta)$, which is easy to compute. We use expectation propagation (EP) [67] to estimate the Gaussian "closest" to the true joint distribution.

Recall that our model assigns super-pixel n to the first layer k whose value is less than the layer's threshold (δ), thus setting $z_n = k$. Equivalently, we can introduce a binary random variable t_n for each layer k, whose value is deterministically related to z_n as follows:

$$t_n = \begin{cases} +1 & \text{if } z_n = k \implies u_n < \delta \\ -1 & \text{if } z_n > k \implies u_n > \delta \end{cases} \tag{4.9}$$

[2]The result follows from Dirichlet multinomial conjugacy. Please see the supplement for relevant details

Note that super-pixels with $z_n < k$ have already been assigned to preceding layers and can be marginalized out before inferring the latent Gaussian layer for the k^{th} layer. We can now express the joint distribution in terms of $\mathbf{t}$:

$$p(\mathbf{u}, \mathbf{v}, \delta, \mathbf{t} \mid \eta) = p(\mathbf{v})\; p(\delta \mid \alpha) \prod_{n=1}^{N} p(u_n \mid \mathbf{v}) p(t_n \mid u_n, \delta) \tag{4.10}$$

Furthermore, since for a given partition $\mathbf{t}$ is known, we can condition on it to get

$$\begin{aligned} p(\mathbf{u}, \mathbf{v}, \delta \mid \mathbf{t}, \eta) = \frac{1}{Z} \mathcal{N}(\mathbf{v} \mid 0, I)\; p(\delta|\alpha) \\ \prod_{n=1}^{N} \mathcal{N}(u_n \mid a_n^T \mathbf{v}, \psi_n) \mathbb{I}(t_n(\delta - u_n) > 0) \end{aligned} \tag{4.11}$$

where Z is the appropriate normalization constant. Note that the indicator functions $\mathbb{I}(t_n(\delta - u_n) > 0)$ and the threshold prior $p(\delta \mid \alpha)$ are the only non Gaussian terms. We approximate these with un-normalized Gaussians, leading to the following approximate posterior

$$q(\mathbf{u}, \mathbf{v}, \delta \mid \mathbf{t}, \eta) = \frac{1}{Z_{\text{EP}}} \mathcal{N}([\mathbf{v}^T\; \mathbf{u}^T\; \delta]^T \mid \mu_{\approx}, \mathbf{\Sigma}_{\approx}) \tag{4.12}$$

where Z_{EP} ensures appropriate normalization. We now iteratively refine the Gaussian approximation using EP [3]. At convergence we compute $Z_{\text{EP}} = \int_{\mathbf{u}} \int_{\mathbf{v}} \int_{\delta} q(\mathbf{u}, \mathbf{v}, \delta, \mathbf{t} \mid \eta)$ which is prior for the k^{th} layer. Finally, we have $p(\mathbf{z}|\eta) \approx \prod_{k=1}^{K} Z_{\text{EP}_k}$.

With the expression for prior in hand, we can now compute the log posterior marginal

$$\log\; p(\mathbf{z} \mid \mathbf{x}, \eta) = \gamma\; \log\; p(\mathbf{x} \mid \mathbf{z}, \rho) + \sum_{k=1}^{K} \log Z_{\text{EP}_k} \tag{4.13}$$

The parameter γ is used to weight the likelihood appropriately. We set $\gamma = \frac{1}{\bar{m}}$, where $\bar{m}$ is the average number of pixels per super-pixel. Recall that our likelihood treats pixels within a super-pixel as independent random variables, necessitating the above down weighting.

[3] Applying EP to our low dimensional model requires an interesting combination of Gaussian belief propagation and expectation propagation. Due to space limitations we haven't included the details of EP here, but all relevant details can be found in the supplement.

4.3.2 Search over partitions

Armed with the ability to evaluate the posterior probability mass for a given image partition, we explore the space of partitions using discrete search. The search performs hill climbing on the posterior surface and explores high probability regions of the partition space. This is similar in spirit to MCMC techniques. Perhaps most similar to our approach is the data driven MCMC approach of Tu *et al.,* [76], which uses a version of the Metropolis-Hastings algorithm along with clever data driven proposals to explore the posterior space. Here, we forgo the requirement of *eventually* converging to the true posterior distribution in exchange for the ease of incorporating flexible search moves and the ability to quickly explore high probability regions of the posterior.

Given a partition we propose a new candidate partition by stochastically choosing one of the following moves:

Merge. Two layers in the current partition are merged into a single layer.

Split. A layer is split into two layers, which are adjacent in layer order. We employ two types of shift moves. Given a layer to be split, the first move works by randomly selecting two seed super-pixels and then assigning all remaining super-pixels to the closest (in appearance space) seed. The initial seeds are chosen such that with high probability they are far in appearance space. The second move employs a connected component operation. If the given layer has disconnected components then one such disconnected component is sampled at random and deemed to be a new layer.

Swap. The swap move reorders the layers in the current partition, by selecting two layers and exchanging their order.

Shift. The shift move refines the partitions found by the other moves. It iterates over all super-pixels in the image assigning each to a segment which maximizes the posterior probability [4]. Observe that the merge and split moves change the number of layers in a partition performing model selection, while swap and shift attempt to find the optimal partition given a model order.

4.4 Learning from Human Segmentations

In this section, we provide methods for quantitatively calibrating the proposed models to appropriate human segmentation biases. Recall that our model has four hyper-parameters, the PY region size hyper-parameter (α), the appearance hyper-parameter

[4] A naive shift move would evaluate the posterior probability of the partition after every super-pixel shift. This proves to be prohibitively expensive, instead we develop an alternative which allows us to evaluate the posterior after one complete sweep through the super-pixels while ensuring that each individual shift by-and-large increases the posterior. Please see the supplement for details.

(ρ) and the GP covariance parameters (A and Ψ). We tune these to the human segmentations from the 200 training images of the Berkeley Segmentation Dataset (BSDS) [61]. We show that in spite of the inherent uncertainty in the segmentations of an image, we are able to learn important low level grouping cues.

Learning size and appearance hyper-parameters. The optimal region size hyper-parameters are the ones that best describe the statistics of the training data. We select $\hat{\alpha} = (\hat{\alpha}_a, \hat{\alpha}_b)$ by performing a grid search over 20 evenly spaced α_a and α_b candidates in the intervals $[0, 1]$ and $[0.5, 20]$ respectively and choosing values which maximize the model's likelihood of the training partitions according to equation 4.8. The appearance hyper-parameters $\hat{\rho} = (\hat{\rho}^t, \hat{\rho}^c)$ are tuned through cross validation on a subset of the training set. For BSDS, the estimated parameters equal $\hat{\alpha}_a = 0.15, \hat{\alpha}_b = 1$ $\hat{\rho}^t = 0.01$ and $\hat{\rho}^c = 0.01$

Learning covariance kernel hyper-parameters. The covariance kernel governs the type of layers that can be expressed by the model. Estimating it accurately is crucial for accurately partitioning images. In [3, 66] the authors use various heuristics to specify this kernel. Here, we take a more data driven approach and learn the kernel from human segmentations. While we cannot expect our training data to provide examples of all important region appearance patterns, it does provide important cues. In particular like [77], we learn to predict the probability that *pairs* of super-pixels occupy the same segment via human segmentations.

For every pair of super-pixels, we consider several potentially informative low-level cues: (i) pairwise Euclidean distance between super-pixel centers; (ii) intervening contours, quantified as the maximal response of the probability of boundary (Pb) detector [71] on the straight line linking super-pixel centers; (iii) local feature differences, estimated via log empirical likelihood ratios of χ^2 distances between super-pixel color and texture histograms [70]. To model non-linear relationships between these four raw features and super-pixel groupings, each feature is represented via the activation of 20 radial basis functions, with the appropriate bandwidth chosen by cross-validation. Concatenating these gives a feature vector ϕ_{ij} for every super-pixel pair i, j. We then train a L_2 regularized logistic regression model to predict the probability of two super-pixels occupying the same segment q_{ij}. Figure 4.2 illustrates the effect of these cues on partitions preferred by the model.

When probabilities are chosen to depend only on the distance between super-pixels the distribution constructed defines a generative model of image features. When these probabilities also incorporate contour cues, the model becomes a conditionally specified distribution on image partitions, analogous to a conditional random field [78].

From probabilities to correlations. Recall that our layers are functions sampled from multivariate Gaussian distributions, with covariance Σ with unit variance and a potentially different correlation c_{ij} for each super-pixel pair i, j. For each super-pixel pair, q_{ij} is *independently* determined by the corresponding correlation coefficient c_{ij}. As detailed in the supplement there exists an one-to-one mapping between the pairwise probabilities and correlations, allowing us to go from the logistic regression outputs (q_{ij}) to correlation matrices. These correlation matrices (C), learned from pairwise probabilities will in general not be positive semi-definite (PSD). We cope by finding the closest PSD unit diagonal matrix to the correlation matrix. We use the recently proposed technique of Borsdorf *et al.*, [79], which solves for A and Ψ by minimizing the Frobenius norm$||C - (AA^T + \Psi)||_F$. It should be noted that even the heuristic approaches of Sudderth and Jordan [3] and Shyr textitet al., [66] can yield non PSD correlation matrices. There the authors ensure positive semi-definiteness by performing an eigen-decomposition of C and retaining only non-negative eigenvalues. This is a cruder approximation and leads to poor results (Figure 4.2).

4.5 Spatially dependent PY model properties

In this section, we explore various properties of our model which may not be immediately obvious.

Prior samples. Our model defines a distribution over image partitions, which can be partially assessed by visualizing partitions sampled from the prior. Figure 4.2 displays such samples. Note that the samples from the conditionally specified models better reflect the structure of the image.

Layers. Our model produces partitions made up of layers, not segments. These layers can have multiple connected components, due to either occlusion by a foreground layer, or a layer support function with multimodal shape. The inferred partitions illustrated in the second row of figure 4.2 illustrate this point. The model groups all buffaloes (in the first image), non-contiguous portions of sky, grass and trees (in the second and third images) in the same layer. Traditional segmentation algorithms, having no notion of layers, would assign each non contiguous region to a separate segment. Our layered representation provides a higher level representation of the scene than is possible with a collection of segments, which allows us to naturally deal with complex visual phenomena such as occlusion.

Implicit prior on layer order. Recall that a partition is an ordered sequence of layers, and the likelihood of a partition is governed by the likelihood of its constituent layers.

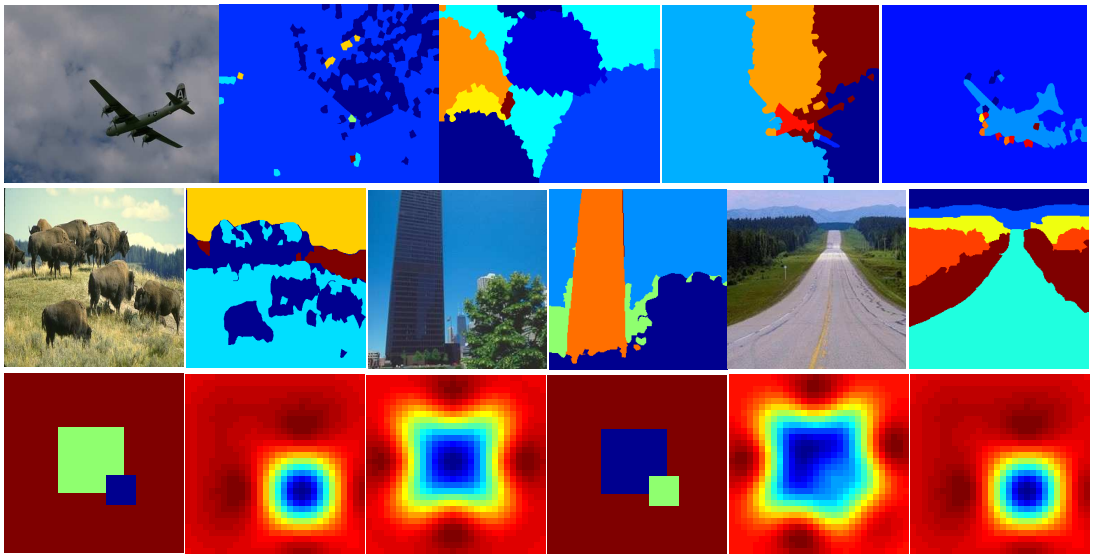

FIGURE 4.2: **Model Properties**. *TOP-* Prior samples from models employing heuristic distance+pb [3], learned distance (PYdist) , learned distance+pb and all cues (PYall) based covariances. *CENTER-* Layered segmentations produced by our method. *BOTTOM* - Three layer synthetic partitions illustrating preferred layer orderings, Layer 1 is displayed in blue and Layer 2 in green. *Left to right:* Partition 1 (*blue = low; red = high*), the inferred Gaussian function for layers 1 and 2, partition 2 and the corresponding Gaussian functions. Under our model, partition 1 has a log probability of -77 while partition 2 has a log probability of -90.

Note that reordering layers can change the set of support functions which produce those layers, which in turn makes certain orderings preferable to others. In general, our GP priors prefer simple shapes over complicated ones and hence our model prefers explaining complicated shapes via an occlusion process. Figure 4.2 illustrates these ideas using two synthetic partitions with the order of layers 1 and 2 flipped. The model [5] prefers the partition in the first column over the one in fourth. As can be seen from the inferred layers, partition 1 is explained by the model using simpler Gaussian functions, while partition 2 has to be explained using more complicated and hence less likely Gaussian functions.

4.6 Experimental Results

In this section we present quantitative evaluations of various aspects of the proposed model along with qualitative results. In all experiments, our model (PYall) used a 200 dimensional low rank representation and ran 200 discrete search iterations, with three random restarts.

[5]Here, we have used a squared exponential covariance kernel with length scale set to half of the partition's diagonal length.

Experimental Setup. We benchmark the algorithm on the Berkeley Image Segmentation Dataset (BSDS300 [61]) and a subset of of Oliva and Torralba's [68] eight natural categories dataset. We sampled the first 30 images from each of the eight categories to create a 240 image dataset.

The performance of the algorithms are quantified using the probabilistic Rand Index (*PRI*) [80], and the segmentation covering (*SegCover*) metric [60]. The partitions produced by our model are made up of layers, which may not be spatially contiguous. However, the benchmarks we evaluate on, define segments to be spatially contiguous regions. To produce these we run connected components on the layers splitting them into spatially contiguous segments.

Quantifying model enhancements. This paper improves on both the model (PYheur) and the corresponding inference algorithm presented in [3] . To quantify the performance gains solely from model enhancements we devise the following test. On BSDS300 test images, we compare the log-posterior assigned to the ground truth human segmentations $p(z_{gt}|x, \eta)$ under both models. Since, we already have access to z_{gt} no inference is required and the model which assigns higher probability mass to the ground truth, models the data better. Figure 4.3 presents a scatter plot comparing both models. It is easy to see that PYall models human segmentations significantly better.

Evaluating inference enhancements. Next, we evaluate the performance improvements resulting from the novel inference algorithm[6]. Figure 4.3 displays the result of running mean field and search based inference from 10 random initializations for a given test image. The log-likelihood plots clearly demonstrate mean field being susceptible to local minima. In contrast, EP based search exhibits robustness and all chains converge to high probability partitions. The bottom row displays the best and worst partitions found by mean field and search. As one would expect, there is wide variability in the quality of mean field partitions, while the search partitions are consistently good. The rightmost top row plot displays randomly chosen partitions from the 10 EP search runs. It demonstrates a high correlation between log likelihoods and rand indexes, again verifying that the partitions favored by our model are also favored by humans.

Comparison against competing methods. In this paper, our goal is not to produce one "optimal" segmentation but to provide a tractable handle on the posterior distribution over image partitions. Nevertheless, here we demonstrate that by summarizing the posterior with the MAP partition we produce results which are competitive with the state-of-the-art segmentation techniques. We compare against four popular segmentation techniques: Mean Shift (MS) [57], Felzenszwalb and Huttenocher's graph

[6] 100 search iterations takes about 30 minutes on a standard quadcore with 4GB of ram.

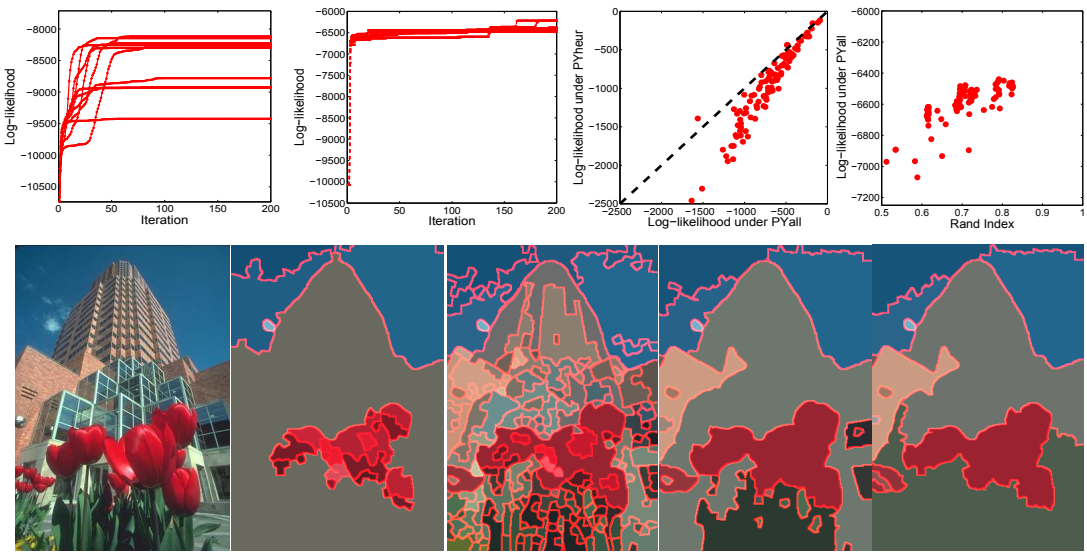

FIGURE 4.3: **Model and inference comparison**. *TOP (Left to right)* Log-likelihood (ll) trace plots of mean field runs, search runs, scatter plot comparing PYall and PYheur, scatter plot of ll vs Rand index. *BOTTOM (Left to right)* Test image, partitions with highest and lowest ll found by mean field, best and worst search partitions.

	BSDS300							LabelMe	
	Ncuts	MS	FH	gPb	PYheur	PYdist	PYall	gPb	PYall
PRI	0.73	0.77	0.77	0.80	0.60	0.69	0.76	0.74	0.73
segCover	0.40	0.48	0.53	0.58	0.45	0.50	0.54	0.54	0.55

TABLE 4.1: Quantitative performance of various algorithms on BSDS300 and LabelMe.

based segmentation (FH) [58], Normalized cuts [56] and gPb contour based segmentation [60][7]. In addition, we also compare against a version of our model which uses only distance cues for learning the covariance kernel (PYdist). Table 4.1 displays the quantitative numbers achieved on the BSDS300 test set. Figure 4.4 demonstrates qualitative differences amongst the methods. PYall is significantly better than both PYheur and PYdist. According to a Wilcoxon's signed rank test (at an 0.01 significance level) it is also significantly better than Ncuts and MS (on segCover metric, within noise on PRI), within noise of FH and statistically worse than gPb on the BSDS300 dataset.

Next, in order to test generalizability, we compare PYall against the top performing method on BSDS – gPb on the LabelMe dataset. The parameters for either method were tuned on BSDS and were not re-tuned to the LabelMe dataset. Table 4.1 displays the results. PYall and gPb are now statistically indistinguishable.

Posterior Summary. Perhaps, a more accurate assessment of our model involves exploring the posterior distribution over partitions. In Figure 4.5 we summarize the posterior distributions, for a few randomly chosen test images, by presenting a set of

[7] All model parameters were tuned by performing a grid search on the training set. See supplement for more details.

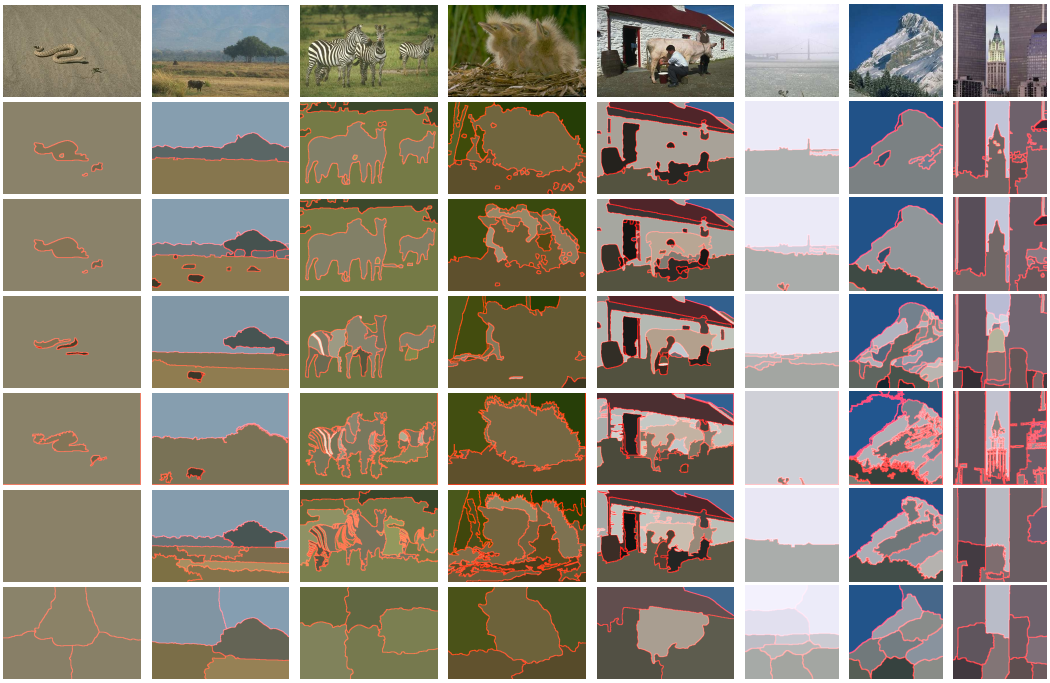

FIGURE 4.4: **Comparisons across models.** From Top to Bottom: PYdist, PYall, gPb, FH, MS, Ncuts

high probability partitions discovered by our algorithm. It is worth noting that the set of multiple partitions produced by our method is richer than those produced by a single multi-resolution segmentation tree [60]. For instance, the partitions in the third and fourth columns of the first two rows of Figure 4.5 are mutually inconsistent with any one segmentation tree, but are nonetheless produced by our algorithm. More interesting ways of leveraging the distribution over partitions is an important direction of future work.

4.7 Discussion

Starting with a promising Bayesian nonparametric model of images partitions, we have developed substantially improved algorithms for learning from example human segmentations, and robustly inferring multiple plausible segmentations of novel images. By defining a consistent distribution on segmentations of varying resolution, this dependent PY process provides a promising building block for other high-level vision tasks.

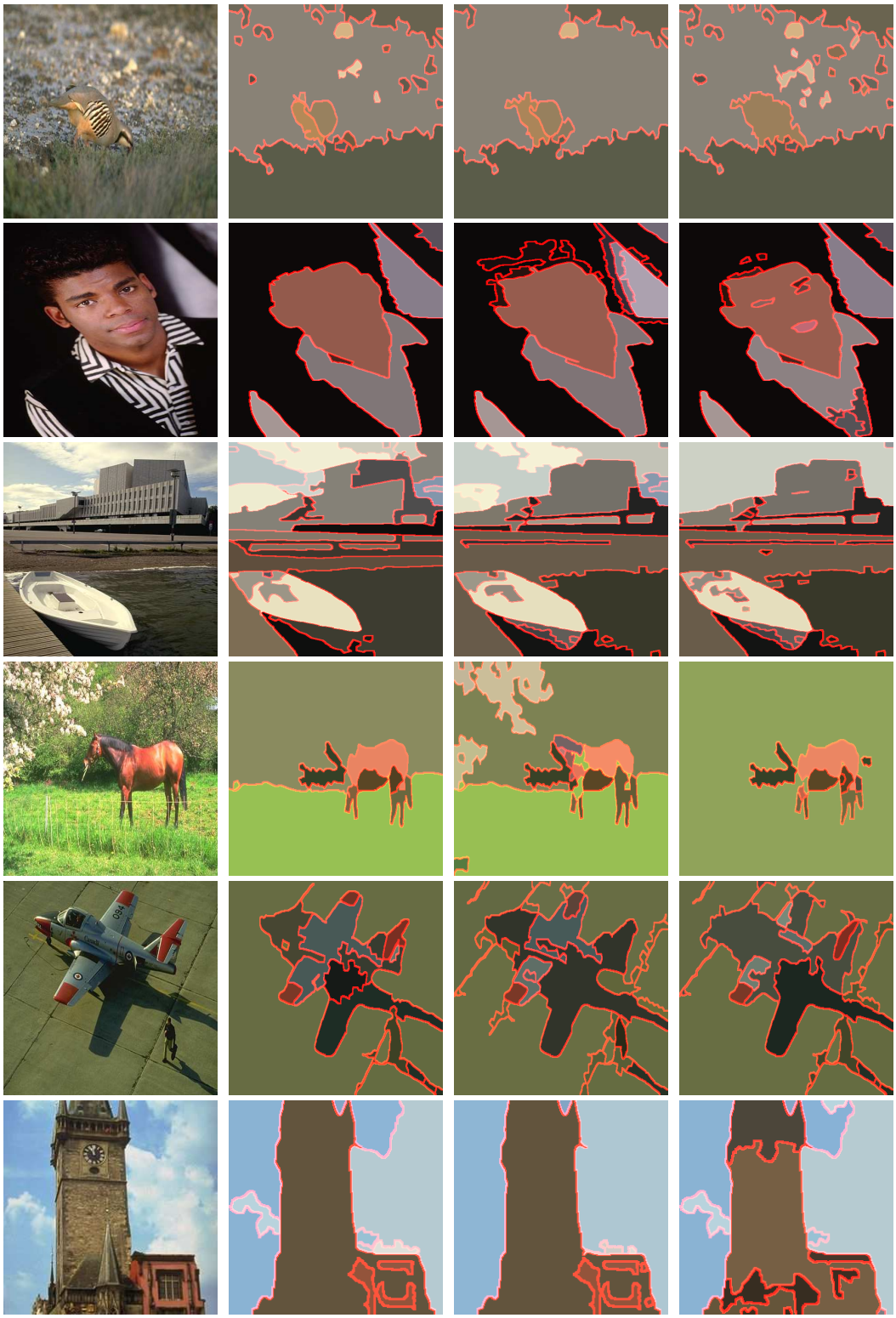

FIGURE 4.5: **Diverse Segmentations.** Each row depicts multiple partitions for a given image. Partitions in the second column are the MAP estimates. Other partitions with significant probability masses are shown in the third and fourth columns.

Chapter 5

Proposed Work

Scene understanding is arguably the holy grail of computer vision. Recent years have seen significant progress in object recognition and image labeling [81, 82]. State-of-the-art systems place bounding boxes around objects in images and optionally produce a dense labeling into one of K predefined classes. Such labelings while useful only provide a superficial understanding of the image. Contrast this to a human analyzing an image, in addition to identifying regions and objects she is able to infer rich geometric structure and disambiguate occlusion and support relationships by reasoning in the 3D scene space rather than the 2D image space. The inability to reason in 3D is one plausible explanation for why despite recent progress computer vision systems fall well short of human performance on comparable tasks. In this chapter, we propose statistical models for simultaneously recovering coarse scene structure while performing semantic image parsing. The goal of this proposal is to illustrate the usefulness of the proposed models by developing efficient inference algorithms and demonstrating improved image parsing and 3D structure recovery performance.

5.1 Background

3D object representations played a crucial role in early vision systems [83–85]. However, they lacked robustness resulting in limited success and over the years fell out of preference for robust statistical methods that worked solely in the image space. With the recent advances in statistical methods there has been a renewed push towards incorporating 3D geometry in object and scene recognition systems. Seminal works along this line of research include [86–88] of Hoiem et al., [86] performed simultaneous object detection, 3D scene geometry and camera viewpoint estimation improving the performance of all three components over independently solving each problem. Sudderth and Torralba [88]

developed a probabilistic model for jointly modeling scene appearance and 3D geometry while accounting for the uncertainty in the number of object and instance uncertainty and showed that coarse scene geometry helped disambiguate among object classes and conversely object categories improved geometry inference. An alternate line of work by Saxena et al., [87, 89] presents a system for recovering depth and 3D geometry estimates given a monocular image without attempting to parse the image into semantic categories.

The advent of cheap depth sensing hardware has further intensified interest in incorporating geometric constraints in scene understanding systems. Several recent works [90–92] use depth as an additional cue for image parsing and demonstrate improved performance over appearance based schemes. They assume the availability of reliable depth maps and no attempt at refining or estimating the scene geometry is made. Furthermore, relying on structured supervised classifiers they are limited to labeling scenes with one of K (usually a small number) classes and provide no mechanism for discovering new classes.

A more sophisticated use of depth has been demonstrated by [93, 94] and [95], who estimate meshes and point clouds from RGB-D data using either off the shelf techniques [96] or by stitching together techniques developed over decades of vision and graphics research [97, 98]. The resulting 3D representations are classified into predefined semantic categories. While estimation of the 3D scene structure is interesting and demonstrably improves image parsing, it is difficult and noise-prone. The techniques in [93–95] all estimate 3D structure in a feed forward fashion and are unable to recover from errors in 3D structure estimation.

The semantic structure from motion framework proposed in [99–101] is an interesting direction which jointly performs object recognition, region segmentation and scene geometry estimation. Initial image segmentations, object detections and camera poses are jointly refined to be consistent with each other. The framework requires the availability of multiple viewpoints of a scene and is again only able to parse an image into a predefined small set of semantic categories.

Another line of work [102–104] attempts to recover volumetric representations from images, by fitting a physically plausible configuration of predefined volumetric primitives. While we don't attempt volumetric reconstructions, instead sticking with simpler pop-up representations [77] we do model in plane shape variations not captured by simple convex 3D primitives – cubes and cylinders.

An interesting orthogonal direction of research [105–107] has focused on synthesizing 2D objects from 3D CAD models. Here during training, viewpoint dependent 3D to 2D mappings are learned. The mappings are then used to localize objects and estimate

object viewpoints in test images. These papers have demonstrated encouraging multi-view object detection and viewpoint estimation results. The focus of our work, while related, is more ambitious. In addition to localizing well defined objects and diffuse regions in 2D, we are also interested in coarse 3D localizations of objects and regions and in producing a dense labeling of the image.

In this proposal, we aim to jointly estimate coarse 3D structure and parse images into meaningful segments from partially labeled RGB-D images of related but not necessarily the same scene. Additionally, we will model the uncertainty in the number of possible object categories and the number of instances populating an observed image. This allows us to discover semantic categories not seen in the training set. For training, we will primarily consider indoor scenes and use RGB-D data from the NYU-depth dataset [108].

5.2 Hierarchical Models for 3D Scenes

Building on the layered representation introduced in the previous chapter we propose a joint model of objects, their appearance in 2D and their shape and pose in the encompassing 3D scene. By jointly analyzing a collection of images we are able to share statistical strength among images and better characterize ambiguous image regions. Crucially, we share information between images by reusing latent 3D representations of objects instead of sharing observed appearance features between images. This provides us a degree of robustness to within object category appearance variation stemming from viewpoint changes.

5.2.1 Counting Objects in Scenes

We explicitly account for the uncertainty in the number of objects populating an image, by placing broad distributions over object counts. Given a collection of N images, object counts can be naturally represented as a matrix Z with rows corresponding to images and columns to object categories, with entry $Z_{i,j}$ containing the number of instances of object category j appearing in image i. If the number of objects appearing in the image corpus is known apriori (K), then the observed number of instances of a specific category in an image can be easily modeled by treating $Z \in \mathbb{R}^{N \times K}$ as a random variable with $Z_{i,j}$ distributed according to a standard exponential family distribution with support over natural numbers, such as the Poisson or the Geometric distribution. In realistic scenarios, additional uncertainty over K needs to be modeled. Here, we leverage recent advances in Bayesian nonparametrics to define broad priors over matrices

with countably infinite columns $Z \in \mathbb{R}^{N\times\infty}$. Through careful construction these priors guarantee that each row of Z will only have a finite sparse subset of columns with non zero counts, which in turn guarantees that, for a finite number of images, the posterior will concentrate probability mass on matrices with random but finite number of columns $\hat{K}$, best supported by data.

Which Bayesian nonparametric prior? There are two popular Bayesian nonparametric priors for modeling count data – the beta Geometric process and the gamma Poisson process. To assess the process best suited for our purposes we resort to empirical analysis of the NYU depth dataset [108]. We count the number of appearances of a semantic category in each image of the dataset and compare the marginal likelihoods of the observed counts under the two processes, over a reasonable range of hyper-parameters. Figure (5.1) displays marginal likelihoods computed using a finite approximation of the two processes. The beta geometric process produces higher marginal likelihoods (note the different scales in the plots) and fits the observations better. Intuitively, this makes sense. The beta geometric distribution (the finite counter part to the beta geometric process) is a better fit for counts when the probability of occurrence decays geometrically. This is certainly true for most semantic categories. While we will see some images with a large number of instances of a category, we will see far more images with a single or no occurrence of an object category (Figure (5.1)).

Beta Geometric process: The Beta Geometric process is defined as

$$\begin{aligned} B &\sim \mathrm{BP}(c, B_0) \\ Z_i \mid B &\sim \mathrm{GeoP}(B) \end{aligned} \tag{5.1}$$

where c is the concentration parameter and B_0 is a base measure over a space Ω. Teh et al. [109] showed that when $c = 1$ and B_0 is continuous, the expected probability of instantiation of an object j is given by the following stick breaking process:

$$\begin{aligned} \mu_j &\sim \mathrm{Beta}(\alpha, 1) \\ \pi_j &= \prod_{l=1}^{j} \mu_l \end{aligned} \tag{5.2}$$

These weights can be used to construct a sample from the Beta process $B = \sum_{j=1}^{\infty} \pi_j \delta_{\omega_j}$, where $\omega_j \sim \frac{1}{\gamma} B_0$, $\gamma = B_0(\Omega)$ is the mass parameter of the Beta process. The Geometric

process then provides us a mechanism for generating rows of Z:

$$\begin{aligned} Z_i \mid B &= \sum_j z_{ij}\delta_{\omega_j} \\ z_{ij} &\sim Geom(1-\pi_j), \end{aligned} \tag{5.3}$$

where the Geometric distribution is defined as

$$p(m|\gamma) = (1-\gamma)^m(\gamma); \text{ for } m \in \{0,1,2,\dots,\} \tag{5.4}$$

5.2.2 2D Images from 3D Scenes

With the distribution over object counts in hand, we can now proceed to define distributions over object shapes and 3D locations with respect to the camera. For each object category $j \in \{1,\dots,\infty\}$ sample its expected appearance $\theta_j \sim \mathrm{H}(\lambda)$ and expected 3D location $\mathbf{l}_j = [l^x, l^y, l^z]^T$

$$\begin{aligned} \ln(l^z) &\sim \mathcal{N}(\mu_z, \sigma_z^2) \\ \begin{bmatrix} l^x \\ l^y \end{bmatrix} &\sim \mathcal{N}(\mu_{xy}, \Sigma_{xy}) \end{aligned} \tag{5.5}$$

We then determine the number of instances of each object category j present in image i by sampling $z_{ij} \sim \mathrm{Geom}(1-p_j)$, where $p = \{p_j \mid j = 1,\dots,\infty\}$ is sampled according to equation 5.2. Almost surely, only a finite number of elements of $Z_i = \{z_{ij} \mid j = 1,\dots,\infty\}$ are non zero. These non zero elements indicate the object categories present in the image with the number of instantiations of each category being z_{ij}.

In order to populate the image with these object instances we need to further determine the location and shape of each instance. In particular, for each instance t of category j present in image i, we sample its 3D location: $\mathbf{l}_{tj} = \mathbf{l}_j + \epsilon_j$, where $\epsilon_j \sim \mathcal{N}(0, \Psi)$ is a category specific noise term.

To simplify the model, we don't attempt a volumetric shape representation. Following [77] we adopt a pop-up representation and model 2d shape variations of instance tj over a "billboard" $\mathcal{X} \in \mathbb{R}^2$ at fixed depth l^z_{tj}. Object shapes are described through shape silhouettes obtained through thresholding of smooth functions sampled from Gaussian

processes (GP). The shape silhouette for instance tj is generated as follows:

$$\begin{aligned} m_{tj}(\mathcal{X}) &\sim \mathrm{GP}(0, K_j(\mathcal{X})) \\ s_{tj}(\mathcal{X}) &\sim \mathrm{GP}(m_{tj}(\mathcal{X}), C_j(\mathcal{X})) \\ v_{tj} &= \{x, y \mid s_{tj}(x, y) > 0; x, y \in \mathcal{X}\}. \end{aligned} \tag{5.6}$$

The intensity of the mean function ($m_{tj}(\mathcal{X})$) governs the expected size of instance tj, by dictating where the shape function s_{tj} exceeds the threshold. By design, the mean intensity peaks at $\ell = [l^x_{tj}, l^y_{tj}]^T$ and decays with increasing distance from ℓ. The rate of decay is controlled by a category specific parameter σ^2_j. C_j is a covariance function controlling the smoothness of the sampled GP layers which in turn controls the variance from the mean shape of the object category. The parameters of the mean and covariance functions can be learned from partially labeled training data. Figure (5.2) provides a pictorial illustration of these ideas.

These silhouettes are then projected onto image i using perspective projection:

$$u^x = \eta \frac{v^x_{tj}}{l^z_{tj}} \; ; \; u^y = \eta \frac{v^y_{tj}}{l^z_{tj}} \tag{5.7}$$

where $u_i = (u^x, u^y)$ represents a pixel in image i and η denotes the pixel scale magnification corresponding to the camera's focal length.

Occlusion and Background. In the above construction, several objects may map to the same pixel u_i. When this happens, pixel u_i is assigned to the object instance tj closest to the camera, effectively occluding other objects.

$$r_{u_i} = \operatorname*{argmin}_{tj} \{l^z_{tj} \mid v_{tj} \to u_i\} \tag{5.8}$$

where $v_{tj} \to u_i$ denotes the projection of object instance tj onto pixel u_i. It might also happen that no object maps to a particular pixel u_i. We sidestep this issue by maintaining a billboard infinitely far from the camera corresponding to the background class. All unassigned pixels are allocated to this background class.

Finally, observed image features are generated as follows:

$$w_{u_i} \sim f(\theta_j); \quad j = h(r_{u_i}) \tag{5.9}$$

where f is an appropriate likelihood function and $h(a)$ is a function that returns the object category of instance a. Determining the exact form of the appearance likelihoods is an open research question that this proposal seeks to answer. An initial idea along

this direction, is to maintain a category-specific distribution over HOG feature cells placed along the contour [110] of category-specific mean shapes. Rotated and translated versions of these HOG cells then describe instance-specific scale and shape variations.

Having described the model's building blocks we can now state the full model as follows:

$$
\begin{aligned}
p(.) = p(Z \mid \alpha) \Bigg(& \prod_j p(l_j \mid \mu, \Sigma) p(\theta_j \mid \lambda) \\
& \prod_i \Bigg\{ \prod_{t=1}^{z_{ij}} \left[p(l^i_{tj} \mid l_j) p(m^i_{tj} \mid l^i_{tj}) p(s^i_{tj} \mid m^i_{tj}, \tau_j) \right] \\
& p(r_{u_i} \mid S_i) p(w_{u_i} \mid \theta_{h(r_{u_i})}) \Bigg\} \Bigg)
\end{aligned}
\tag{5.10}
$$

where $S_i = \{s^i_{11}, \ldots, s^i_{z_{i1}1}, \ldots s^i_{12}, \ldots, s^i_{z_{i2}2}, \ldots\}$ and τ_j represents the mean and covariance hyper-parameters of the category specific Gaussian processes over shape functions. Additionally, to make the notation unambiguous we have indexed the image specific random variables with the super-script i.

5.3 Timeline

Our aim is to develop efficient and reliable inference algorithms for the model described in the previous section. We envision this happening in two steps. First, we will focus on a simpler parametric version (replacing the stochastic process in Equation (5.1) with a beta geometric distribution) of the model. Given a collection of images with semantic labels, color and depth information the model will learn category specific distribution over layers, depths, and appearances. Subsequently, the learned model will be used to decompose unlabeled test images into semantic layers that will be placed at depths predicted by the model. We will validate the model by benchmarking its performance on a train-test split of the NYU depth dataset that contains images annotated with both labels and depth estimates (Figure (5.1)). We plan to submit our findings to ECCV-2014 (early March).

The next step would involve developing the nonparametric extension. This more sophisticated model would not only learn to parse images into semantic layers, but also discover visual categories constituting a semantic category. A semantic category might have several visual categories owing to view point changes and within category appearance variations. We hope to have this wrapped up by early June and submit a paper reporting our progress to NIPS-2014.

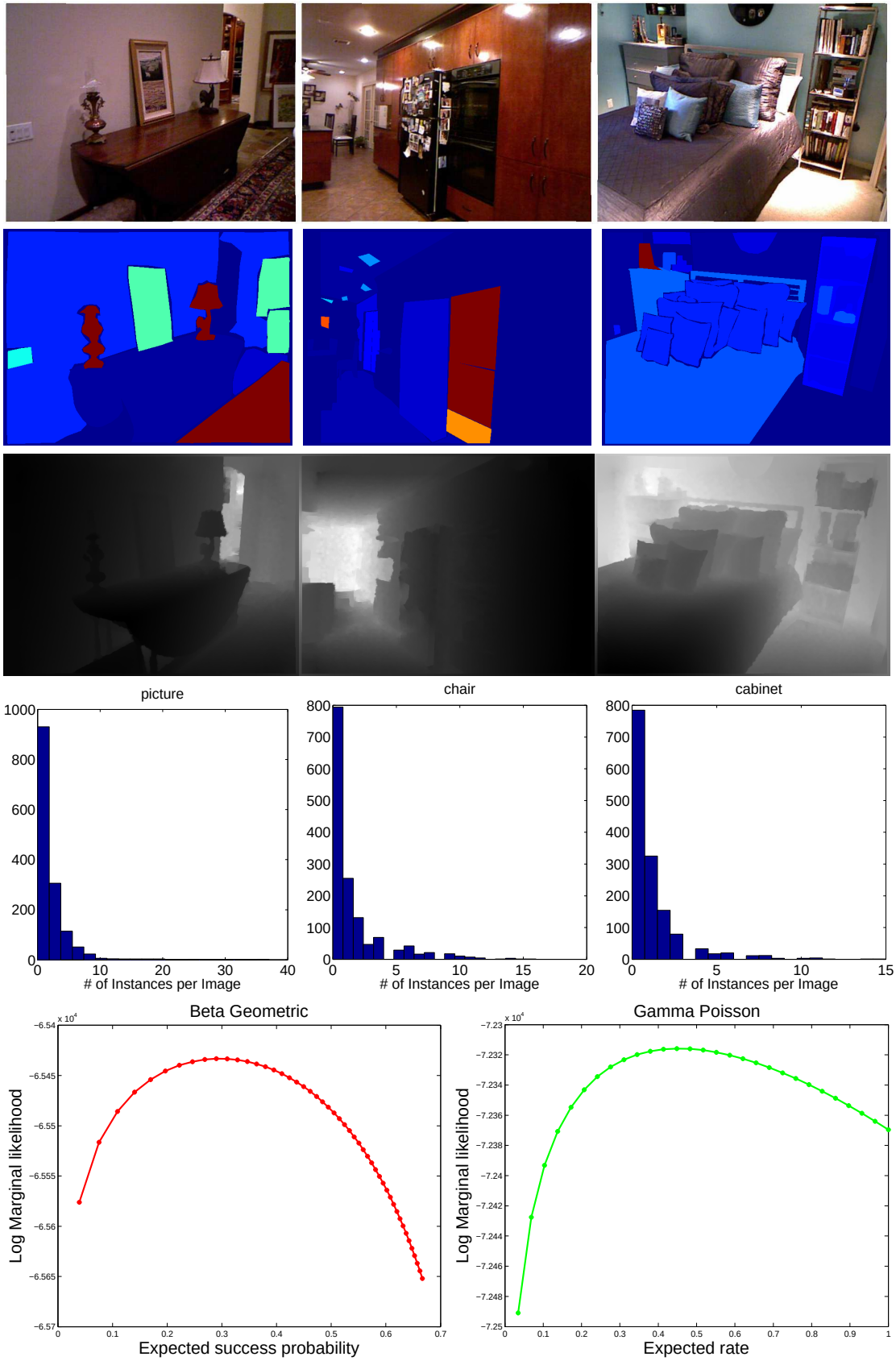

FIGURE 5.1: NYU Depth dataset. The top three rows displays example images, corresponding labels and depth information (lighter shades imply larger depths) respectively. The fourth row displays histograms of object count per image for three popular semantic categories, "picture", "chair" and "cabinet". Notice that most images exhibit a small number of instances of each category. The bottom row compares marginal likelihoods of empirical object counts under the Beta Geometric and Gamma Poisson processes.

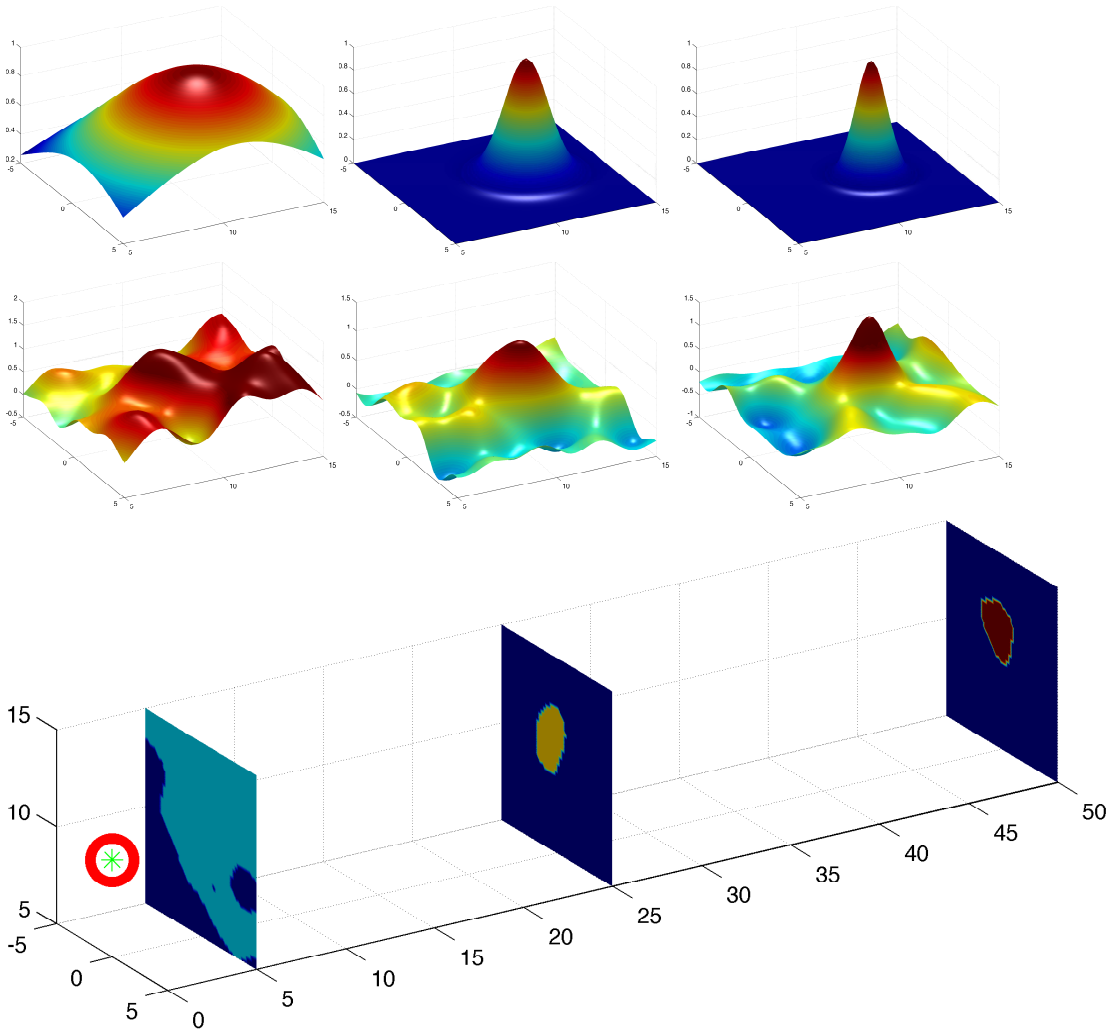

FIGURE 5.2: Silhouettes and Billboards. The figure illustrates instances from three object categories whose locations have been sampled according to Equation (5.5). The columns have been sorted according to depth from the camera. The top two rows illustrate the mean (m_{tj}) and GP sampled (with squared exponential covariance kernels) shape functions (s_{tj}) specified in Equation (5.6). The bottom row displays the billboards propped up at l^z_{tj}. The dark blue represents regions where the shape function is less than the threshold and the colored regions represent areas where the shape function exceeds the threshold. The shape silhouette corresponds to the boundaries of the colored regions. These billboards are then projected into the image plane (visualized here as the circled asterisk) through perspective projection. Note that the background billboard hasn't been visualized here, but is assumed to exist at infinity.

Appendix A

Body Segmentation Details

A.1 Marginalizing over $\mathcal{A}$

Let $Y = [\mathbf{y}_1, ...\mathbf{y}_N] \in R^{3\times N}$ denote the coordinates of mesh faces assigned to the same part in a given pose. Let $X = [\mathbf{x}_1, ...\mathbf{x}_N] \in R^{4\times N}$ represent the corresponding reference (homogeneous) coordinates. The distribution of $Y|X$ (for a given part and pose) is then given by

$$Y|X \sim \mathcal{MN}(\mathcal{A}X, \Sigma, \mathbf{I}) \tag{A.1}$$

From [15] - F.10 we have

$$p(Y|X,\Sigma) = \int p(Y,\mathcal{A}|X,\Sigma)d\mathcal{A} = \frac{|K|^{3/2}}{|2\pi\Sigma|^{N/2}|S_{xx}|^{3/2}} exp\{-\frac{1}{2}tr(\Sigma^{-1}S_{y|x})\} \tag{A.2}$$

and

$$S_{xx} = XX^T + K \tag{A.3}$$

$$S_{yx} = YX^T + MK \tag{A.4}$$

$$S_{y|x} = YY^T + MKM^T - S_{yx}(S_{xx})^{-1}S_{yx}^T \tag{A.5}$$

Finally, the marginal likelihood is given by

$$p(Y|X) = \int p(Y|X,\Sigma)p(\Sigma|n_0,S_0)d\Sigma \tag{A.6}$$

$$= \int \frac{|K|^{3/2}}{|2\pi\Sigma|^{3N/2}|S_{xx}|^{3/2}} exp\{-\tfrac{1}{2}tr(\Sigma^{-1}S_{y|x})\} \tag{A.7}$$

$$\frac{|S_0|^{n_0/2}|\Sigma|^{-(4+n_0)/2}}{2^{3n_0/2}\Gamma_3(n_0/2)} exp\{-\tfrac{1}{2}tr(\Sigma^{-1}S_0)\}d\Sigma \tag{A.8}$$

$$p(Y|X) = \int \frac{|K|^{3/2}|S_0|^{n_0/2}|\Sigma|^{-(4+n_0)/2}}{|2\pi\Sigma|^{3N/2}|S_{xx}|^{3/2}2^{3n_0/2}\Gamma_3(n_0/2)} exp\{-\frac{1}{2}tr(\Sigma^{-1}(S_{y|x}+S_0))\}d\Sigma \tag{A.9}$$

$$p(Y|X) = \frac{|K|^{3/2}|S_0|^{n_0/2}}{|2\pi|^{3N/2}|S_{xx}|^{3/2}2^{3n_0/2}\Gamma_3(n_0/2)} \int |\Sigma|^{-(3+N+n_0+1)/2} exp\{-\frac{1}{2}tr(\Sigma^{-1}(S_{y|x}+S_0))\} \tag{A.10}$$

$$p(Y|X) = \frac{|K|^{3/2}|S_0|^{n_0/2}2^{(N+n_0)3/2}\Gamma_3((N+n_0)/2)}{|2\pi|^{3N/2}|S_{xx}|^{3/2}2^{3n_0/2}\Gamma_3(n_0/2)|S_0+S_{y|x}|^{(N+n_0)/2}} \int IW(N+n_0, S_{y|x}+S_0)d\Sigma \tag{A.11}$$

The marginal likelihood for one part in one pose is then given by

$$p(Y|X,K,n_0,S_0) = \frac{|K|^{\frac{3}{2}}|S_0|^{\frac{n_0}{2}}\Gamma_3\left(\frac{N+n_0}{2}\right)}{\pi^{\frac{3N}{2}}|S_{xx}|^{\frac{3}{2}}|S_0+S_{y|x}|^{\frac{(N+n_0)}{2}}\Gamma_3(\frac{n_0}{2})} \tag{A.12}$$

Appendix B

Hierarchical ddCRP details

B.1 Inference Details

Algorithm 1: Iterative sampling of customer and table links.

for $i \in 1 \dots N$ **do**
- $C^*, \mathcal{K}^* \longleftarrow$ CustLinkProposal$(i, \mathbf{X}, \mathcal{K}, C, \alpha, D, \alpha_0, D_0)$
- Compute acceptance ratio ρ ; `/*See supplement*/`
- With probability $\propto \min(1, \rho)$, accept $C, \mathcal{K} \longleftarrow C^*, \mathcal{K}^*$

for $t \in T(C)$ **do**
- $k_t \sim p(k_t \mid \mathcal{K}_{-t}, C, \mathbf{X}, \alpha_0, D_0)$; `/*Gibbs update` k_t`*/`

B.1.1 Acceptance Ratios

Notational details:

1. i is a customer, k_{t_i} denotes the link of the table on which customer i sits.
2. If two tables, one with customer i and another with j are merged, the resulting table is denoted t_{ij} and the corresponding table link is $k_{t_{ij}}$
3. $C = \{c_1, \dots, c_{i-1}, c_i = i', \dots, c_N\}$
   $C^o = \{c_1, \dots, c_{i-1}, c_i = i, \dots, c_N\}$
   $C^* = \{c_1, \dots, c_{i-1}, c_i = i^*, \dots, c_N\}$ and $c_i = i^*$ is denoted as c_i^*

The proposed algorithm changes an existing partition by either

Algorithm 2: CustLinkProposal

input : $i, \mathbf{X}, \mathcal{K}, C, \alpha, D, \alpha_0, D_0$
output: $\mathcal{K}^*, C^*, q_\cdot(C^*, \mathcal{K}^* \mid \mathcal{K}, C)$
$i' \longleftarrow c_i$
Set $c_i = i$ and update $C^o = \{c_1, \ldots, c_{i-1}, c_i = i, \ldots, c_N\}$;
$L_{t_i} = \{t_\ell \mid k_{t_\ell} = t_i \;\&\; t_\ell \neq t_i\}$; `/*Set of all tables pointing to` t_i`, except self loops.*/`
if *A new table is created by setting* $c_i = i$ **then**
  Set split = true; `/*Record the occurrence of a split.*/`
  $\mathcal{K} \longleftarrow$ ReassignLinks (L_{t_i})
  $k^*_{t_{i'}} \longleftarrow k_{t_{i,i'}}$; `/*A split table retains the current table's link.*/`
Sample $c_i^* \sim q(c_i^*)$
if c_i^* *causes two existing tables to merge* **then**
  Set $t_{i,i^*} = t_i \cup t_{i^*}$
  $L_{t_{i,i^*}} = L_{t_i} \cup L_{t_{i^*}}$ and Update $\mathcal{K}$ to reflect the merge
  if *split* **then** `/* Split+Merge */`
    $k^*_{t_{i,i^*}} \longleftarrow k_{t_{i^*}}$
    $q_{sm}(C^*, \mathcal{K}^* \mid C, \mathcal{K}, X) = (0.5)^{|L_{t_i}|} q(c_i^*)$
  else `/* No split + Merge */`
    Delete k_{t_i}
    $k^*_{t_{i,i^*}} \longleftarrow k_{t_{i^*}}$
    $q_m(C^*, \mathcal{K}^* \mid C, \mathcal{K}, X) = q(c_i^*)$
else
  if *split* **then** `/* Split+No Merge */`
    Sample $k^*_{t_i} \sim p(k^*_{t_i} \mid \alpha_0, D_0(C^*), \mathbf{X}, \mathcal{K}_{-t_i})$;
    $q_s(C^*, \mathcal{K}^* \mid C, \mathcal{K}, X) = (0.5)^{|L_{t_i}|} q(c_i^*) p(k^*_{t_i} \mid D_0(C^*), \mathbf{X}, \mathcal{K}^*_{-t_i})$
  else `/* No Split+No Merge */`
    `/*No change to partition - Do Nothing. */`
    $q_{nc}(C^*, \mathcal{K}^* \mid C, \mathcal{K}, X) = q(c_i^*)$

Algorithm 3: ReassignLinks

input : L_{t_i}
output: $\mathcal{K}$
`/*Reassign links pointing to a split table. Links are assigned to one of the two split tables. */`
for $\ell \in L_{t_i}$ **do**
  $b_\ell \sim \text{Ber}(0.5)$
  if $b_\ell = 1$ **then**
    $L_{t_i} = L_{t_i} / t_\ell$
    $L_{t_{i'}} = L_{t_{i'}} \cup t_\ell$
    $k_{t_j} = t_{i'}$;

1. Merging existing tables : No new table is created when $c_i = i$ and exiting tables are merged after sampling c_i^*. The transition probability of this move is:

$$q_m(C^*, \mathcal{K}^* \mid C, \mathcal{K}, X) = q(c_i^*) \tag{B.1}$$

2. Or splitting an existing table: A new table is created when $c_i = i$ and and no tables are merged after sampling c_i^*.

$$q_s(C^*, \mathcal{K}^* \mid C, \mathcal{K}, X) = (0.5)^{|L_{t_i}|} q(c_i^*) p(k_{t_i}^* \mid D_0(C^*), \mathbf{X}, \mathcal{K}_{-t_i}^*) \tag{B.2}$$

3. Or both merging and splitting tables (Figure B.1) : A new table is created when $c_i = i$ and tables are merged after sampling c_i^*.

$$q_{sm}(C^*, \mathcal{K}^* \mid C, \mathcal{K}, X) = (0.5)^{|L_{t_i}|} q(c_i^*) \tag{B.3}$$

Finally, the move might not change a partition at all: No new table is created when $c_i = i$ and no tables are merged after sampling c_i^*.

$$q_{nc}(C^*, \mathcal{K}^* \mid C, \mathcal{K}, X) = q(c_i^*) \tag{B.4}$$

with $q(c_i^*) = p(c_i^* \mid \alpha, D)$ for the prior proposal and $q(c_i^*) = p(c_i^* \mid \mathbf{X}, \mathcal{K}_{-t_i}, C_{-i})$ for the posterior predictive proposal. Moves 1 and 4 are reverses of each other, while moves 2 and 3 are their own reverses.

Recall that a proposal is accepted with probability $\propto \min(1, \rho)$, where

$$\rho = \frac{p(\mathbf{X}, C^*, \mathcal{K}^*)}{p(\mathbf{X}, C, \mathcal{K})} \frac{q_{rev}(C, \mathcal{K} \mid C^*, \mathcal{K}^*, \mathbf{X})}{q_{fwd}(C^*, \mathcal{K}^* \mid C, \mathcal{K}, \mathbf{X})} \tag{B.5}$$

B.1.2 Split move

Let us first consider the split move under the prior proposal. The merge move is the reverse of a split move. Hence we have:

$$\rho_s = \frac{p(\mathbf{X}, C^*, \mathcal{K}^*)}{p(\mathbf{X}, C, \mathcal{K})} \frac{q_m(C, \mathcal{K} \mid C^*, \mathcal{K}^*, \mathbf{X})}{q_s(C^*, \mathcal{K}^* \mid C, \mathcal{K}, \mathbf{X})} \tag{B.6}$$

Substituting Equations B.2 and B.1 above we get:

$$\rho_s = \frac{p(\mathbf{X}, C^*, \mathcal{K}^*)}{p(\mathbf{X}, C, \mathcal{K})} \frac{p(c_i = i' \mid \alpha, D)}{p(c_i = i^* \mid \alpha, D) p(k_{t_i}^* \mid D_0(C^*), \mathbf{X}, \mathcal{K}_{-t_i}^*) (0.5)^{|L_{t_{i,i'}}|}} \tag{B.7}$$

Dropping dependence on α, D, D_0 for notational convenience we have:

$$\rho_s = \frac{1}{(0.5)^{|L_{t_{i,i'}}|}} \frac{p(C^*)p(\mathcal{K}^* \mid C^*)p(\mathbf{X} \mid \mathcal{K}^*, C^*)}{p(C)p(\mathcal{K} \mid C)p(\mathbf{X} \mid \mathcal{K}, C)} \frac{p(c_i = i')}{p(c_i = i^*)p(k^*_{t_i} \mid \mathbf{X}, \mathcal{K}^*_{-t_i}, C^*)} \tag{B.8}$$

The customer links cancel out between the likelihood and hastings ratios:

$$\rho_s = \frac{1}{(0.5)^{|L_{t_{i,i'}}|}} \frac{p(\mathcal{K}^* \mid C^*)p(\mathbf{X} \mid \mathcal{K}^*, C^*)}{p(\mathcal{K} \mid C)p(\mathbf{X} \mid \mathcal{K}, C)} \frac{1}{p(k^*_{t_i} \mid \mathbf{X}, \mathcal{K}^*_{-t_i}, C^*)} \tag{B.9}$$

$$\rho_s = \frac{1}{(0.5)^{|L_{t_{i,i'}}|}} \frac{p(\mathcal{K}^* \mid C^*)p(\mathbf{X} \mid \mathcal{K}^*, C^*)}{p(\mathcal{K} \mid C)p(\mathbf{X} \mid \mathcal{K}, C)} \frac{p(\mathbf{X}, \mathcal{K}^*_{-t_i}, C^*)}{p(\mathbf{X}, \mathcal{K}^*, C^*)} \tag{B.10}$$

$$\rho_s = \frac{1}{(0.5)^{|L_{t_{i,i'}}|}} \frac{\cancel{p(\mathcal{K}^* \mid C^*)}\cancel{p(\mathbf{X} \mid \mathcal{K}^*, C^*)}}{p(\mathcal{K} \mid C)p(\mathbf{X} \mid \mathcal{K}, C)} \frac{p(\mathbf{X} \mid \mathcal{K}^*_{-t_i}, C^*)p(\mathcal{K}^*_{-t_i} \mid C^*)\cancel{p(C^*)}}{\cancel{p(\mathbf{X} \mid \mathcal{K}^*, C^*)}\cancel{p(\mathcal{K}^* \mid C^*)}\cancel{p(C^*)}} \tag{B.11}$$

$$\rho_s = \frac{1}{(0.5)^{|L_{t_{i,i'}}|}} \frac{p(\mathcal{K}^*_{-t_i} \mid C^*)}{p(\mathcal{K} \mid C)} \frac{p(\mathbf{X} \mid \mathcal{K}^*_{-t_i}, C^*)}{p(\mathbf{X} \mid \mathcal{K}, C)} \tag{B.12}$$

Note that the number of table links in $\mathcal{K}$ = number of table links in $\mathcal{K}^*_{-t_i}$, and depending on the distance between tables the ratio of table links may be further simplified. For instance, for the intersection over union distance used in the video segmentation algorithm all but the links of the affected tables cancel out with the above ratio simplifying to:

$$\frac{p(\mathcal{K}^*_{-t_i} \mid C^*)}{p(\mathcal{K} \mid C)} = \frac{p(k^*_{t_{i'}} \mid C^*) \prod_{t_j \in L_{t_{i,i'}}} p(k^*_{t_j} \mid C^*)}{p(k_{t_{i,i'}} \mid C) \prod_{t_j \in L_{t_{i,i'}}} p(k_{t_j} \mid C)} \tag{B.13}$$

Similarly, likelihood terms of dishes not affected by the split cancel out. If the two tables serve different dishes we have

$$\frac{p(\mathbf{X} \mid \mathcal{K}^*_{-t_i}, C^*)}{p(\mathbf{X} \mid \mathcal{K}, C)} = \frac{p(\mathbf{X}_{\mathbf{z}=z_{i'}} \mid \mathcal{K}^*_{-t_i}, C^*, \lambda)p(\mathbf{X}_{\mathbf{z}=z_{i^*}} \mid \mathcal{K}^*_{-t_i}, C^*, \lambda)}{p(\mathbf{X}_{\mathbf{z}=z_i} \mid \mathcal{K}, C, \lambda)} \tag{B.14}$$

where $\mathbf{X}_{\mathbf{z}=z_{i'}}$ refers to all customers sharing a dish with i'. Note that since in the pre spilt state the i and i' are sitting at the same table, $\{\mathbf{X}_{\mathbf{z}=z_{i'}} \mid C, \mathcal{K}\} = \{\mathbf{X}_{\mathbf{z}=z_i} \mid C, \mathcal{K}\}$, but $\{\mathbf{X}_{\mathbf{z}=z_{i'}} \mid C^*, \mathcal{K}^*\}$ may not equal $\{\mathbf{X}_{\mathbf{z}=z_i} \mid C^*, \mathcal{K}^*\}$ in the new split state. If the two tables do serve the same dish we have:

$$\frac{p(\mathbf{X} \mid \mathcal{K}^*_{-t_i}, C^*)}{p(\mathbf{X} \mid \mathcal{K}, C)} = \frac{p(\mathbf{X}_{\mathbf{z}=z_i} \mid \mathcal{K}^*_{-t_i}, C^*, \lambda)}{p(\mathbf{X}_{\mathbf{z}=z_i} \mid \mathcal{K}, C, \lambda)} \tag{B.15}$$

Note that due to the missing k_{t_i} link in the numerator the two sets of customers in general will not be the same.

Pseudo Gibbs Proposals. Recall that we sample the customer link from the following proposal distribution:

$$q(c_i^*) \propto p(c_i^* \mid \alpha, D)\Gamma(\mathbf{X}, \mathbf{z}, \lambda), \tag{B.16}$$

where

$$\Gamma(\mathbf{X}, \mathbf{z}, \lambda) = \begin{cases} \dfrac{p(\mathbf{X}_{\mathbf{z}(\Delta)=m_a} \cup \mathbf{X}_{\mathbf{z}(\Delta)=m_b} \mid \lambda)}{p(\mathbf{X}_{\mathbf{z}(\Delta)=m_a} \mid \lambda)p(\mathbf{X}_{\mathbf{z}(\Delta)=m_b} \mid \lambda)} & \text{if } c_i^* \text{ merges dishes } m_a \text{ and } m_b \\ p(c_i^* \mid \alpha, D) & \text{otherwise,} \end{cases} \tag{B.17}$$

where $\Delta = \{C_{-i}, k_{t_i} = t_i, \mathcal{K}_{-t_i}\}$.

In the above equations, $\mathcal{K}_{-t_i}$ represents the set of all table links excluding k_{t_i}. Observe that in algorithm 2 a new customer link is proposed only after resetting the current link, which may cause a table to split. If a split does occur, $\mathcal{K}_{-t_i}$ represents the set of table links appropriately reassigned to account for the split.

Split move acceptance ratio for pseudo Gibbs proposals First, observe that when a particular link proposal doesn't cause two dishes to merge (either because it doesn't merge tables or because it merges two tables serving the same dish) the prior proposal and the pseudo Gibbs proposals are identical. This implies that when a split is proposed that splits a table but not a dish the pseudo Gibbs acceptance ratio is equal to the prior proposal acceptance ratio given in Equation (B.12).

Now let us consider the case when the proposed customer link causes dishes to be split. The reverse move must then cause two distinct dishes (m_a and m_b) to be merged. Thus the reverse transition probability is:

$$q_m(C, \mathcal{K} \mid C^*, \mathcal{K}^*, X) = \frac{1}{\mathcal{C}_i} p(c_i = i' \mid \alpha, D) \frac{p(\mathbf{X}_{m_a} \cup \mathbf{X}_{m_b} \mid \lambda)}{p(\mathbf{X}_{m_a} \mid \lambda)p(\mathbf{X}_{m_b} \mid \lambda)}, \tag{B.18}$$

where $\mathcal{C}_i$ is the appropriate normalization constant for the discrete pseudo Gibbs proposal. Under, the pseudo Gibbs proposal, the probability of a link that doesn't cause a merge of dishes is given by $\frac{1}{\mathcal{C}_i} p(c_i = i^* \mid \alpha, D)$ which is then combined with the probability of sampling a new table link to give the forward transition probability for the split move:

$$q_s(C^*, \mathcal{K}^* \mid C, \mathcal{K}, X) = (0.5)^{|L_{t_{i,i'}}|} \frac{1}{\mathcal{C}_i} p(c_i = i^* \mid \alpha, D) p(k_{t_i}^* \mid D_0(C^*), \mathbf{X}, \mathcal{K}_{-t_i}) \tag{B.19}$$

Plugging these values in Equation (B.5) leads to the following ratio:

$$\eta_s = \frac{1}{(0.5)^{|L_{t_{i,i'}}|}} \frac{p(\mathcal{K}_{-t_i}^* \mid C^*)}{p(\mathcal{K} \mid C)} \tag{B.20}$$

Finally the acceptance ratio for the split move under the pseudo Gibbs proposal is:

$$\rho_s^{pg} = \begin{cases} \rho_s & \text{if the split tables share the same dish} \\ \eta_s & \text{otherwise} \end{cases} \tag{B.21}$$

where η_s is Merge move ratios are analogously computed.

B.1.3 Split+Merge moves

These moves allow for customers to shift between tables. Figure B.1 illustrates such a move. Colors correspond to dishes served. The acceptance ratio for the prior proposal

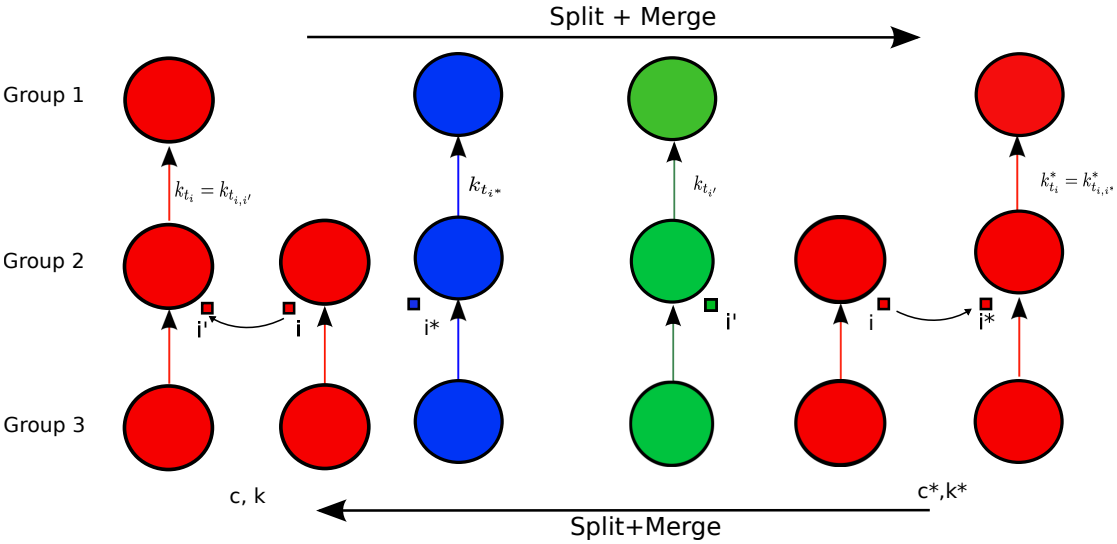

FIGURE B.1: Split+Merge move.

works out to

$$\rho_{sm} = (0.5)^{|L_{t_{i,i^*}}|-|L_{t_{i,i'}}|} \frac{p(\mathcal{K}^* \mid C^*)p(\mathbf{X} \mid \mathcal{K}^*, C^*)}{p(\mathcal{K} \mid C)p(\mathbf{X} \mid \mathcal{K}, C)} \tag{B.22}$$

again with table links and likelihood terms not affected by the move canceling out. The pseudo Gibbs acceptance ratio works out to a simple ratio of table links:

$$\rho_{sm}^{pg} = (0.5)^{|L_{t_{i,i^*}}|-|L_{t_{i,i'}}|} \frac{p(\mathcal{K}^* \mid C^*)}{p(\mathcal{K} \mid C)} \tag{B.23}$$

Finally, moves which change customer links but do not cause a change to the partition structure have acceptance ratios of 1 and are always accepted under either proposals.

B.2 Von Mises-Fisher distributions

If a d-dimensional unit vector $\mathbf{X} \in \mathbb{R}^D$ and $||\mathbf{X}||_2 = 1$ follows a von Mises-Fisher (vMF) distribution then:

$$p(\mathbf{X} \mid \mu, \kappa) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa)} e^{\kappa\mu^T\mathbf{X}} \tag{B.24}$$

with $||\mu||_2 = 1$, $\kappa \geq 0$ and $d \geq 2$. The parameter μ corresponds to the mean direction and κ is a concentration parameter. $\mathcal{I}_r(.)$ is a modified Bessel function of the first kind with degree r. The modified Bessel function grows exponentially fast with κ canceling out the exponential growth of the numerator in equation B.24, keeping the density well behaved.

B.3 Likelihood Model - Known κ, Unknown direction μ

$$\mu \sim \text{vMF}(\mu_0, \kappa_0) \tag{B.25}$$

Observed vectors $\mathbf{x}_i$ are then generated according to:

$$\mathbf{X}_i \mid \mu, \kappa \sim \text{vMF}(\mu, \kappa) \tag{B.26}$$

B.3.1 Posterior on μ

$$\begin{aligned}
&p(\mu \mid \mathbf{X}, \mu_0, \kappa_0, \kappa) \propto p(\mu \mid \mu_0, \kappa_0) p(\mathbf{X} \mid \mu, \kappa) \\
&\propto p(\mu \mid \mu_0, \kappa_0) \prod_{i=1}^{N} p(\mathbf{X}_i \mid \mu, \kappa) \\
&\propto \frac{{\kappa_0}^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)} e^{{\kappa_0{\mu_0}^T\mu}} \prod_{i=1}^{N} \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa)} e^{\kappa\mu^T\mathbf{X}_i} \\
&\propto \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{(2\pi)^{\frac{(N+1)d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} e^{\kappa_0\mu_0^T\mu + \kappa\mu^T\sum\mathbf{X}_i}
\end{aligned} \tag{B.27}$$

$$\begin{aligned}
&\propto \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{(2\pi)^{\frac{(N+1)d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} e^{\kappa_0\mu_0^T\mu + \kappa(\sum\mathbf{X}_i)^T\mu} \\
&\propto \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{(2\pi)^{\frac{(N+1)d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} e^{(\kappa_0\mu_0 + \kappa(\sum\mathbf{X}_i))^T\mu}
\end{aligned} \tag{B.28}$$

A valid vMF distribution requires the mean vector to be a unit vector. In general, $(\kappa_0\mu_0 + \kappa\sum \mathbf{X}_i)$ will not be a unit vector and we need to explicitly normalize it.

$$\begin{aligned}
&\propto \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{(2\pi)^{\frac{(N+1)d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} e^{||(\kappa_0\mu_0+\kappa(\sum \mathbf{X}_i))||_2 \frac{(\kappa_0\mu_0+\kappa(\sum \mathbf{X}_i))^T}{||(\kappa_0\mu_0+\kappa(\sum \mathbf{X}_i))||_2}\mu} \\
&\propto \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{(2\pi)^{\frac{(N+1)d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} e^{\tilde{\kappa}\tilde{\mu}^T\mu}
\end{aligned} \tag{B.29}$$

$$\begin{aligned}
&\propto \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{(2\pi)^{\frac{(N)d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} \frac{\mathcal{I}_{\frac{d}{2}-1}(\tilde{\kappa})}{\tilde{\kappa}^{\frac{d}{2}-1}} \frac{\tilde{\kappa}^{\frac{d}{2}-1}}{(2\pi)^{(\frac{d}{2})}\mathcal{I}_{\frac{d}{2}-1}(\tilde{\kappa})} e^{\tilde{\kappa}\tilde{\mu}^T\mu} \\
&\propto \text{vMF}(\tilde{\mu}, \tilde{\kappa})
\end{aligned} \tag{B.30}$$

where $\tilde{\kappa} = \Big|\Big|(\kappa_0\mu_0 + \kappa(\sum \mathbf{X}_i))\Big|\Big|_2$ and $\tilde{\mu} = \frac{(\kappa_0\mu_0 + \kappa(\sum \mathbf{X}_i))^T}{||(\kappa_0\mu_0 + \kappa(\sum \mathbf{X}_i))||_2}$

B.3.2 Marginal Likelihood

$$p(\mathbf{X} \mid \mu_0, \kappa, \kappa_0) = \int p(\mathbf{X}, \mu \mid \mu_0, \kappa, \kappa_0) d\mu \tag{B.31}$$

From equation B.30 we have:

$$\int p(\mathbf{X}, \mu \mid \mu_0, \kappa, \kappa_0) d\mu = \int \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{(2\pi)^{\frac{(N)d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} \frac{\mathcal{I}_{\frac{d}{2}-1}(\tilde{\kappa})}{\tilde{\kappa}^{\frac{d}{2}-1}} \frac{\tilde{\kappa}^{\frac{d}{2}-1}}{(2\pi)^{(\frac{d}{2})}\mathcal{I}_{\frac{d}{2}-1}(\tilde{\kappa})} e^{\tilde{\kappa}\tilde{\mu}^T\mu} d\mu \tag{B.32}$$

$$= \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{(2\pi)^{\frac{(N)d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} \frac{\mathcal{I}_{\frac{d}{2}-1}(\tilde{\kappa})}{\tilde{\kappa}^{\frac{d}{2}-1}} \int \frac{\tilde{\kappa}^{\frac{d}{2}-1}}{(2\pi)^{(\frac{d}{2})}\mathcal{I}_{\frac{d}{2}-1}(\tilde{\kappa})} e^{\tilde{\kappa}\tilde{\mu}^T\mu} d\mu \tag{B.33}$$

$$= \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{(2\pi)^{\frac{(N)d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} \frac{\mathcal{I}_{\frac{d}{2}-1}(\tilde{\kappa})}{\tilde{\kappa}^{\frac{d}{2}-1}} \int \text{vMF}(\mu \mid \tilde{\mu}, \tilde{\kappa}) d\mu \tag{B.34}$$

$$= \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{(2\pi)^{\frac{(N)d}{2}}\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} \frac{\mathcal{I}_{\frac{d}{2}-1}(\tilde{\kappa})}{\tilde{\kappa}^{\frac{d}{2}-1}} \tag{B.35}$$

$$= \frac{1}{(2\pi)^{\frac{Nd}{2}}} \frac{\kappa_0^{\frac{d}{2}-1}\kappa^{N(\frac{d}{2}-1)}}{\tilde{\kappa}^{\frac{d}{2}-1}} \frac{\mathcal{I}_{\frac{d}{2}-1}(\tilde{\kappa})}{\mathcal{I}_{\frac{d}{2}-1}(\kappa_0)(\mathcal{I}_{\frac{d}{2}-1}(\kappa))^N} \tag{B.36}$$

Appendix C

Layered Segmentation Details

C.1 Low rank Expectation Propagation

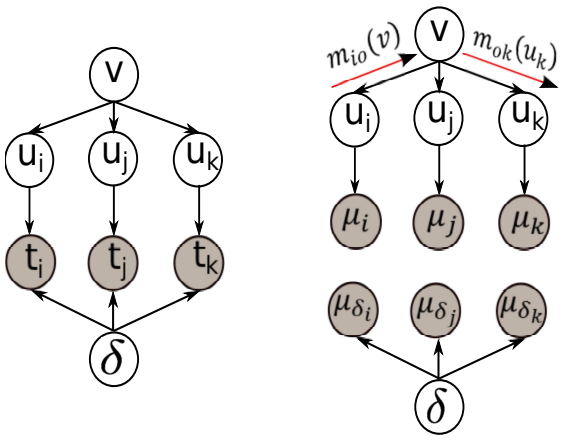

FIGURE C.1: **True and Approximate distributions.** Graphical models representing the distribution of random variables in a layer (*We have left out the hyper-parameters on δ and v*). **Left**: True distribution. **Right**: Approximate distribution.

As previously noted, the random variables associated with each layer of our model can be treated independently of the others. Following the notation introduced in Section 3, we have

$$p(\boldsymbol{u}, \mathbf{v}, \delta \mid \boldsymbol{t}, \alpha) \propto \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I}) p(\delta|\alpha) \prod_{n=1}^{N} N(u_n|a_n^T\mathbf{v}, \psi_n)\mathbb{I}(t_n(\delta - u_n) > 0) \tag{C.1}$$

We approximate this distribution with a Gaussian distribution of the form:

$$q(\boldsymbol{u}, \mathbf{v}, \delta \mid \boldsymbol{t}, \alpha) \propto \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I})\mathcal{N}(\delta \mid \tilde{\mu}_p, \tilde{\sigma}_p^2) \prod_{n=1}^{N} \mathcal{N}(u_n \mid a_n^T\mathbf{v}, \psi_n)\mathcal{N}(u_n \mid \tilde{\mu}_n, \tilde{\sigma}_n^2)\mathcal{N}(\delta \mid \tilde{\mu}_{\delta_n}, \tilde{\sigma}_{\delta_n}^2) \tag{C.2}$$

The graphical models corresponding to the true and approximate distributions are shown in Figure C.1. EP proceeds by removing an approximate factor and substituting it with the corresponding true factor, giving rise to the augmented distribution. The moments of this augmented distribution are then computed and the parameters of the approximate factor is updated by matching the moments of the approximate and augmented distributions. Next, we demonstrate how these quantities are computed for our model.

Firstly, note that our approximation assumes independence between δ and $\{\mathbf{u}, \mathbf{v}\}$. From figure C.1 and using standard Gaussian BP results we have

$$q(\mathbf{v} \mid \boldsymbol{t}) \propto \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I}) \prod_{n=1}^{N} m_{no}(\mathbf{v}) \tag{C.3}$$

with

$$m_{no}(\mathbf{v}) \propto \mathcal{N}(\mathbf{v} \mid \boldsymbol{\tau}_{no}^{-1}\boldsymbol{\nu}_{no}, \boldsymbol{\tau}_{no}^{-1}),\ \boldsymbol{\tau}_{no} = \frac{\tilde{\tau}_n}{1+\psi_n\tilde{\tau}_n} a_n a_n^T \tag{C.4}$$

$$\boldsymbol{\nu}_{no} = \frac{\tilde{\nu}_n}{1+\psi_n\tilde{\tau}_n} a_n,\ \tilde{\nu}_n = \tilde{\tau}_n\tilde{\mu}_n,\ \tilde{\tau}_n = \tilde{\sigma}_n^{-2} \tag{C.5}$$

Thus, we have the following result

$$q(\mathbf{v} \mid \boldsymbol{t}) \propto \mathcal{N}(\mathbf{v} \mid, \boldsymbol{\tau}_{pos}^{-1}\boldsymbol{\nu}_{pos}, \boldsymbol{\tau}_{pos}^{-1}) \tag{C.6}$$

$$\boldsymbol{\tau}_{pos} = \boldsymbol{I} + \sum_{n=1}^{N} \frac{\tilde{\tau}_n}{1+\psi_n\tilde{\tau}_n} a_n a_n^T \tag{C.7}$$

$$\boldsymbol{\nu}_{pos} = \sum_{n=1}^{N} \frac{\tilde{\nu}_n}{1+\psi_n\tilde{\tau}_n} a_n \tag{C.8}$$

We can remove the effect of an approximate factor by dividing out the corresponding message.

$$q(\mathbf{v} \mid \boldsymbol{t}_{-n}) \propto \mathcal{N}(\mathbf{v} \mid, \boldsymbol{\tau}_{-n}^{-1}\boldsymbol{\nu}_{-n}, \boldsymbol{\tau}_{-n}^{-1}) \tag{C.9}$$

$$\boldsymbol{\tau}_{-n}^{-1} = (\boldsymbol{\tau}_{pos} - \boldsymbol{\tau}_{no})^{-1} \tag{C.10}$$

$$\boldsymbol{\nu}_{-n} = \boldsymbol{\nu}_{pos} - \boldsymbol{\nu}_{no} \tag{C.11}$$

Note that $\boldsymbol{\tau}_{-n}^{-1}$ can be efficiently computed using the following rank one update:

$$\boldsymbol{\tau}_{-n}^{-1} = \boldsymbol{\Sigma} - (-m)\frac{\boldsymbol{\Sigma} a_n a_n^T \boldsymbol{\Sigma}}{1 - m a_n^T \boldsymbol{\Sigma} a_n} \tag{C.12}$$

$$m = \frac{\tilde{\tau}_n}{1+\psi_n\tilde{\tau}_n}\ and\ \boldsymbol{\tau}_{-n}^{-1} = \boldsymbol{\Sigma} \tag{C.13}$$

Next observe that

$$q(u_n|\boldsymbol{t}) \propto \mathcal{N}(u_n \mid \tilde{\mu}_n, \tilde{\sigma}_n^2) m_{on}(u_n) \tag{C.14}$$

$$q(u_n|\boldsymbol{t}_{-n}) \propto m_{on}(u_n) \tag{C.15}$$

$$m_{on}(u_n) \propto \mathcal{N}(u_n \mid \tau_{on}^{-1}\nu_{on}, \tau_{on}^{-1}) \tag{C.16}$$

A little algebra reveals that the parameters of m_{on} are given by

$$\tau_{on}^{-1} = \psi_n + a_n^T \boldsymbol{\tau}_{-n}^{-1} a_n \; and \; \tau_{on}^{-1}\nu_{on} = a_n^T \boldsymbol{\tau}_{-n}^{-1} \boldsymbol{\nu}_{-n} \tag{C.17}$$

Similarly, the parameters of the distribution $q(\delta \mid \boldsymbol{t}_{-n}) \propto \mathcal{N}(\delta \mid \tau_{-\delta_n}^{-1}\nu_{-\delta_n}, \tau_{-\delta_n}^{-1})$ can be computed. Finally, the moments of the following augmented distribution need to be computed:

$$q(u_n, \delta \mid \boldsymbol{t}_{-n}) \mathbb{I}(t_n(\delta - u_n) > 0) = q(\delta \mid \boldsymbol{t}_{-n}) q(u_n \mid \boldsymbol{t}_{-n}) \mathbb{I}(t_n(\delta - u_n) > 0) \tag{C.18}$$

A little bit of algebra leads to the following closed form formula for the relevant normalization constants.
Normalization constant of the augmented distribution (0^{th} order moment):

$$P = \Phi\left(\frac{t_n(\mu_{-\delta_n} - \mu_{-n})}{\sqrt{\sigma_{-n}^2 + \sigma_{-\delta_n}^2}}\right) = \Phi(h_n) \tag{C.19}$$

First and Second order moments for δ:

$$E[\delta] = \mu_{-\delta_n} + t_n \frac{\sigma_{-\delta_n}^2 N(h_n)}{\Phi(h_n)\sqrt{\sigma_{-n}^2 + \sigma_{-\delta_n}^2}} \tag{C.20}$$

$$E[\delta^2] = 2\mu_{-\delta_n} E[\delta] - \mu_{-\delta_n}^2 + \sigma_{-\delta_n}^2 - \frac{\sigma_{-\delta_n}^4 h_n N(h_n)}{\Phi(h_n)(\sigma_{-n}^2 + \sigma_{-\delta_n}^2)} \tag{C.21}$$

First and Second order moments for u_n:

$$E[u_n] = \mu_{-n} - t_n \frac{\sigma_{-n}^2 N(h_n)}{\Phi(h_n)\sqrt{\sigma_{-n}^2 + \sigma_{-\delta_n}^2}} \tag{C.22}$$

$$E[u_n^2] = 2\mu_{-n} E[u_n] - \mu_{-n}^2 + \sigma_{-n}^2 - \frac{\sigma_{-n}^4 h_n N(h_n)}{\Phi(h_n)(\sigma_{-n}^2 + \sigma_{-\delta_n}^2)} \tag{C.23}$$

where $\mu_{-n} = \tau_{on}^{-1}\nu_{on}$, $\mu_{-\delta_n} = \tau_{-\delta_n}^{-1}\nu_{-\delta_n}$, $\sigma_{-\delta_n}^2 = \tau_{-\delta_n}^{-1}$, $\sigma_{-n}^2 = \tau_{on}^{-1}$.

The parameters of the approximate factor corresponding to u_n can now be computed and the posterior on $\mathbf{v}$ updated using a rank one update, analogous to standard Gaussian

process classification [111]. A final issue worth noting is that we have a non standard prior on δ which is difficult to deal with. We approximate the prior on δ with another Gaussian factor. The moments required for computing the parameters of this Gaussian are estimated numerically. Since, δ is an unidimensional quantity, numerical moment computation is easy and efficient. Furthermore, these moments are required only once per EP sweep, where a sweep is defined as circling through all the super-pixels. Thus the added computational cost of numerical moment computation is negligible.

C.1.1 Computational Complexity

Observe that we only explicitly maintain a Gaussian posterior distribution on $\mathbf{v}$ which is a D dimensional quantity. Thus, the complexity of one EP sweep is $O(ND^2)$ as opposed to standard Gaussian process classification which has a complexity of $O(N^3)$ where N is the number of super-pixels. Observe that for any candidate partition, the prior for all layers can be evaluated in parallel. Thus, the cost of running T search iterations, each iteration running t sweeps of EP is $O(tTND^2)$.

C.2 Likelihood Evaluation

The likelihood computation involves evaluating the independent color and texture integrals

$$\int_{\Theta} p(\mathbf{x}|\boldsymbol{z},\Theta)p(\Theta|\rho)d\Theta = \int_{\theta^c} p(\mathbf{x^c}|\boldsymbol{z},\theta^c)p(\theta^c|\rho^c)d\theta^c \int_{\theta^t} p(\mathbf{x^t}|\boldsymbol{z},\theta^t)p(\theta^t|\rho^t)d\theta^t \tag{C.24}$$

which is a standard multinomial-Dirichlet integral. We provide the solution to the color integral here for the sake of completeness (*To simplify notation we denote* θ^c, $\mathbf{x}^c$ *by just* θ *and* $\mathbf{x}$).

For K segments and N super-pixels we have,

$$\int_{\theta} p(\mathbf{x}|\boldsymbol{z},\theta^c)p(\theta|\rho^c)d\theta = \prod_{k=1}^{K}\int_{\theta_k} p(\theta_k|\rho^c)\prod_{n=1}^{N} p(\boldsymbol{x}_n|z_n,\theta_k)^{\mathbb{I}(z_n=k)}d\theta_k \tag{C.25}$$

$$= \prod_{k=1}^{K}\int_{\theta_k} \Delta(\rho^c)\prod_{w=1}^{W_c}\theta_{kw}^{\rho_w^c-1}\prod_{n=1}^{N}\prod_{w=1}^{W_c}(\theta_{kw}^{x_{nw}})^{\mathbb{I}(z_n=k)}d\theta_k \tag{C.26}$$

$$= \prod_{k=1}^{K}\Delta(\rho^c)\int_{\theta_k}\prod_{w=1}^{W_c}\theta_{kw}^{\rho_w^c-1}\prod_{w=1}^{W_c}(\theta_{kw})^{\sum_n x_{nw}\times\mathbb{I}(z_n=k)}d\theta_k \tag{C.27}$$

$$= \prod_{k=1}^{K} \Delta(\rho^c) \int_{\theta_k} \prod_{w=1}^{W_c} (\theta_{kw})^{x_w^k + \rho_w - 1} d\theta_k \tag{C.28}$$

$$= \prod_{k=1}^{K} \frac{\Delta(\rho^c)}{\Delta(\rho^c + x^k)} \tag{C.29}$$

In the above derivation $\Delta(\rho^c) = \frac{\Gamma(\sum_w \rho_w^c)}{\prod_w \Gamma(\rho_w^c)}$ and x_w^k = number of times word w occurs with segment k. Putting it all together we have

$$\int_{\Theta} p(\mathbf{x}|\boldsymbol{z}, \Theta) p(\Theta|\rho) d\Theta = \prod_{k=1}^{K} \frac{\Delta(\rho^c)}{\Delta(\rho^c + x_k^{(c)})} \frac{\Delta(\rho^t)}{\Delta(\rho^t + x_k^{(t)})} \tag{C.30}$$

C.3 Search Details

In this section we provide details of our search algorithm.

C.3.0.1 Search Pseudo-code

Get the initial partition $\mathbf{z}^0$ using k-means.
Set maxIter = 200, $i = 1$, bestMode = $\mathbf{z}^0$
while $i \leq$ maxIter **do**
  while $p(\mathbf{z}^i \mid \mathbf{x}, \eta) \geq p(\mathbf{z}^{i-1} \mid \mathbf{x}, \eta)$ **do**
    Apply shift move to $\mathbf{z}^{i-1}$ to get $\mathbf{z}^i$
    bestMode = $\mathbf{z}^i$
    $i = i + 1$
  end while
  if $i \leq$ maxIter **then**
    Select a move from the set { Merge, Swap, Split }
    Apply the selected move to $\mathbf{z}^{i-1}$ to get $\mathbf{z}^i$
    if $p(\mathbf{z}^i \mid \mathbf{x}, \eta) \geq p(\mathbf{z}^{i-1} \mid \mathbf{x}, \eta)$ **then**
      bestMode = $\mathbf{z}^i$
    end if
    $i = i + 1$
  end if
end while
return bestMode

C.3.1 Shift move details

***Notation note**: z_n is a categorical random variable assuming one of K values, where K is the number of components in the partition $\mathbf{z}$. t_n on the other hand is a binary random variable indicating whether super-pixel n is assigned to layer k or not. A is a N-by-D matrix, with rows $a_1^T \dots a_N^T$*

We are interested in optimizing $p(\mathbf{z} \mid \mathbf{x}, \eta)$ with respect to $\mathbf{z} = \{z_1, z_2 ... z_n\}$. In the shift move we assign each $z_n = \hat{k}$ such that $\hat{k} = \underset{k}{argmax}\ p(z_n = k \mid z_{-n}, \alpha, A, \Psi) p(\mathbf{x} \mid \mathbf{z}, \rho)$. Note that this implies we are optimizing $p(\mathbf{z} \mid \mathbf{x}, \eta)$ one z_n at a time.

1. for each super-pixel n
   (a) for each layer k
      i. If super-pixel n is defined for layer k; Compute the approximate posterior cavity distribution on $\mathbf{v}$; $q(\mathbf{v}|\boldsymbol{t}_{-n}) \propto \mathcal{N}(\mathbf{v}|\boldsymbol{\mu}_{-n}, \Sigma_{-n})$ and the approximate posterior cavity distribution for the layer's threshold δ_k; $q(\delta_k|\boldsymbol{t}_{-n}) = N(\delta_k|\mu_{-\delta_n}, \sigma^2_{-\delta_n})$
      ii. If super-pixel n is not defined for layer k (ie it has already been assigned to a previous layer) the posterior distributions on $\mathbf{v}$ and δ_k are themselves the cavity distributions.
      iii. Next, compute the parameters of the conditional distribution $q(u_n|\mathbf{v}, \boldsymbol{t}_{-n}) = q(u_n|\mu_*, {\sigma^2}_*)$, given by

$$\mu_* = a_n^T \boldsymbol{\mu}_{-n} \tag{C.31}$$
$$\sigma_*^2 = \Psi_n + a_n^T \Sigma_{-n} a_n \tag{C.32}$$

      iv. Finally, compute $\pi_{nk} = p(t_n = 1|t_{-n})$ as follows

$$\pi_{nk} = E_q[\mathbb{I}(u_n < \delta_k)] \tag{C.33}$$
$$= \int\int \mathbb{I}(u_n < \delta_k) N(u_n|\mu_*, \sigma_*^2) N(\delta_k|\mu_{-\delta_n}, \sigma^2_{-\delta_n}) du_n d\delta_k \tag{C.34}$$
$$= \Phi\left(\frac{\mu_{-\delta_n} - \mu_*}{\sqrt{\sigma_*^2 + \sigma^2_{-\delta_n}}}\right) \tag{C.35}$$

      v. The probability of super-pixel n getting assigned to layer k is given by

$$p(z_n = k \mid z_{-n}) = p(u_n < \delta_k \mid u_n > \delta_l) = \pi_{nk} \prod_l (1 - \pi_{nl});\ with\ l = 1 ... k-1 \tag{C.36}$$

vi. Compute the posterior probability of the super-pixel assignment

$$p(\mathbf{z} \mid \mathbf{x}, \rho, \alpha) \propto p(z_n = k \mid z_{-n}) \int p(\mathbf{x} \mid \mathbf{z}, \theta) p(\theta \mid \rho) d\theta \tag{C.37}$$

(b) Finally, assign n to layer $\hat{k}$ which maximizes posterior probability

$$\hat{k} = \underset{k}{argmax}\; p(z_n = k \mid z_{-n}) \int p(\mathbf{x} \mid \mathbf{z}, \theta) p(\theta \mid \rho) d\theta \tag{C.38}$$

(c) For all layers affected by the shift of super-pixel n, update the corresponding posterior distribution on $\mathbf{v}$ by a EP projection for the relevant super-pixel. Care is taken such that when a previously invalid super-pixel gets shifted into a layer, the old posterior is treated as the new cavity distribution. Likewise when a super-pixel is shifted out of a layer, the old cavity distribution is treated as the new posterior.

C.4 Probability to Correlation mapping details

Covariance Calibration. We are interested in estimating a mapping between the correlation (c) of a pair of Gaussian random variables (u_i and u_j), and the conditionally learned probability q_{ij} of the corresponding super-pixels i and j being assigned to the same layer. According to our generative model, two super-pixels i and j can be assigned to the same layer k iff both u_i and u_j are less than the threshold δ_k. Hence, the probability of two super-pixels being assigned to layer k is

$$p_-|\delta_k = \int_{-\infty}^{\delta_k} \int_{-\infty}^{\delta_k} \mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix} \right) du_i du_j \tag{C.39}$$

Furthermore, we can marginalize out the unknown thresholds δ_k

$$q_-^k(\alpha, \rho) = \int_{-\infty}^{\infty} \int_{-\infty}^{\delta_k} \int_{-\infty}^{\delta_k} \mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix} \right) p(\delta_k|\alpha) du_i du_j d\delta_k \tag{C.40}$$

Let us further define

$$q_+^k(\alpha, \rho) = \int_{-\infty}^{\infty} \int_{\delta_k}^{\infty} \int_{\delta_k}^{\infty} \mathcal{N}\left(\begin{bmatrix} u_i \\ u_j \end{bmatrix} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix} \right) p(\delta_k|\alpha) du_i du_j d\delta_k \tag{C.41}$$

which is the probability that both u_i and u_j are greater than the δ_k. Note that neither q_- nor q_+ can be computed in closed form and are both numerically approximated.

Now observe that two super-pixels i and j can be assigned to the same layer, if they are both assigned to the first layer or if neither is assigned to the first layer but both are assigned to the second layer or if neither is assigned to the first two layers but both are assigned to the third layer and so on. We can thus express p_{ij} as

$$q_{ij} = q_-^1(\alpha,\rho) + q_-^2(\alpha,\rho)q_+^1(\alpha,\rho) + q_-^3(\alpha,\rho)q_+^1(\alpha,\rho)q_+^2(\alpha,\rho) + \dots \quad \text{(C.42)}$$

$$\approx \sum_{k=1}^{K} q_-^k(\alpha,\rho) \prod_{l=1}^{K-1} q_+^l(\alpha,\rho) \quad \text{(C.43)}$$

where we have explicitly truncated our model to have K (some large number) layers. The above equation defines the sought relationship and allows us to map conditionally learned q_{ij} to pairwise correlations of Gaussian random variables. The mapping is visualized in figure C.2.

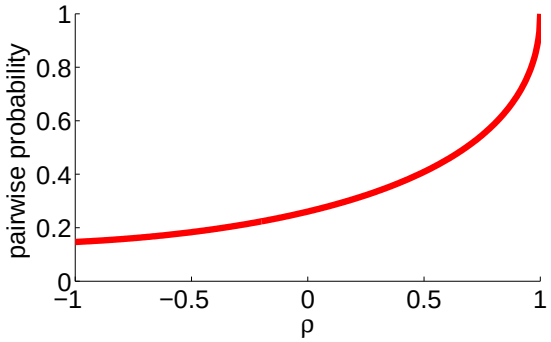

FIGURE C.2: Mapping between correlation coefficients and pairwise probabilities

Bibliography

[1] D. Hirshberg, M. Loper, E. Rachlin, and M.J. Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In *ECCV*, pages 242–255, 2012.

[2] Alexander Bronstein, Michael Bronstein, and Ron Kimmel. Calculus of non-rigid surfaces for geometry and texture manipulation. *IEEE Tran. on Viz. and Computer Graphics*, 13:902–913, 2007. ISSN 1077-2626. doi: http://doi.ieeecomputersociety.org/10.1109/TVCG.2007.1041.

[3] Erik B. Sudderth and Michael I. Jordan. Shared segmentation of natural scenes using dependent Pitman-Yor processes. In *NIPS*, pages 1585–1592, 2008.

[4] Erik B. Sudderth and Michael I. Jordan. Shared segmentation of natural scenes using dependent pitman-yor processes. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1585–1592. 2008.

[5] David M. Blei and Peter I. Frazier. Distance dependent Chinese restaurant processes. *J. Mach. Learn. Res.*, 12:2461–2488, November 2011.

[6] J. Pitman. *Combinatorial Stochastic Processes*. Lecture Notes for St. Flour Summer School. Springer-Verlag, New York, NY, 2002.

[7] D. M. Blei and P. I. Frazier. Distant dependent chinese restaurant process. *arXiv:0910.1022v1*, 2009.

[8] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, and A. Tal. Mesh segmentation — A comparative study. In *SMI*, 2006.

[9] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28 (3):73:1–73:12, 2009.

[10] Evangelos Kalogerakis, Aaron Hertzmann, and Karan Singh. Learning 3D Mesh Segmentation and Labeling. *ACM Transactions on Graphics*, 29(4):102:1–102:12, July 2010.

[11] S. Ghosh, A. B. Ungureanu, E. B. Sudderth, and D. Blei. Spatial distance dependent Chinese restaurant processes for image segmentation. In *NIPS*, pages 1476–1484, 2011.

[12] D. Anguelov, D. Koller, H. Pang, P. Srinivasan, and S. Thrun. Recovering articulated object models from 3d range data. In *UAI*, pages 18–26, 2004.

[13] J. Franco and E. Boyer. Learning temporally consistent rigidities. In *IEEE CVPR*, pages 1241–1248, 2011.

[14] R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *JCGS*, 9(2):249–265, 2000.

[15] E. B. Fox. *Bayesian Nonparametric Learning of Complex Dynamical Phenomena*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2009.

[16] A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*. Chapman & Hall/CRC, October 2000. ISBN 1584880465. URL http://www.worldcat.org/isbn/1584880465.

[17] Tong-Yee Lee, Yu-Shuen Wang, and Tai-Guang Chen. Segmenting a deforming mesh into near-rigid components. *The Visual Computer*, 22(9):729–739, September 2006. ISSN 0178-2789. doi: 10.1007/s00371-006-0059-6. URL http://dx.doi.org/10.1007/s00371-006-0059-6.

[18] Guy Rosman, Michael M. Bronstein, Alexander M. Bronstein, Alon Wolf, and Ron Kimmel. Group-valued regularization framework for motion segmentation of dynamic non-rigid shapes. In *SSVM'11*, pages 725–736, 2012. ISBN 978-3-642-24784-2. doi: 10.1007/978-3-642-24785-9_61. URL http://dx.doi.org/10.1007/978-3-642-24785-9_61.

[19] Stefanie Wuhrer and Alan Brunton. Segmenting animated objects into near-rigid components. *The Visual Computer*, 26:147–155, 2010. ISSN 0178-2789. URL http://dx.doi.org/10.1007/s00371-009-0394-5.

[20] Richard D. De Veaux. Mixtures of linear regressions. *Computational Statistics and Data Analysis*, 8(3):227 – 245, 1989. ISSN 0167-9473.

[21] Lauren Hannah, David M Blei, and Warren B Powell. Dirichlet process mixtures of generalized linear models. *Journal of Machine Learning Research*, 12:1923–1953, 2011.

[22] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. In *Computer Graphics Forum (Proc. Eurographics 2009)*, volume 2, pages 337–346, March 2009.

[23] R. N. Shepard. Multidimensional scaling, tree-fitting, and clustering. *Science*, 210: 390–398, October 1980.

[24] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel spectral clustering in distributed systems. *IEEE PAMI*, 33(3):568–586, 2011.

[25] Rong Liu and Hao Zhang. Segmentation of 3D meshes through spectral clustering. In *Pacific Conference on Computer Graphics and Applications*, pages 298–305, 2004.

[26] Edilson de Aguiar, Christian Theobalt, Sebastian Thrun, and Hans-Peter Seidel. Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum*, 27(2):389–397, 2008.

[27] Oren Freifeld and Michael J. Black. Lie bodies: A manifold representation of 3D human shape. In *European Conf. on Computer Vision (ECCV)*, Part I, LNCS 7572, pages 1–14. Springer-Verlag, October 2012.

[28] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *PAMI*, 24(8):1026–1038, August 2002.

[29] Dongwoo Kim and Alice Oh. Accounting for data dependencies within a hierarchical Dirichlet process mixture model. In *CIKM*, CIKM '11, pages 873–878, 2011.

[30] X. Ren and J. Malik. Learning a classification model for segmentation. *ICCV*, 2003.

[31] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *ICCV*, 2009. URL http://www.wisdom.weizmann.ac.il/~vision/SingleImageSR.html.

[32] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145 – 175, 2001.

[33] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database web-based tool for image annotation. *IJCV*, 77:157–173, 2008.

[34] E. B. Sudderth and M. I. Jordan. Shared segmentation of natural scenes using dependent pitman-yor processes. *NIPS 22*, 2008.

[35] G. Mori. Guiding model search using segmentation. *ICCV*, 2005.

[36] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. *CVPR*, 2:54–61, 2003.

[37] D. R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. PAMI*, 26 (5):530–549, 2004.

[38] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8):888–905, 2000.

[39] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24: 603–619, May 2002. ISSN 0162-8828. doi: 10.1109/34.1000236. URL http://dl.acm.org/citation.cfm?id=513073.513076.

[40] Sonia Jain and Radford Neal. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182, 2000.

[41] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. PAMI*, 29(6):929–944, 2007.

[42] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, June 2010.

[43] Ce Liu, William T. Freeman, Edward H. Adelson, and Yair Weiss. Human-assisted motion annotation. In *CVPR*. IEEE Computer Society, 2008.

[44] X. Ren and J. Malik. Learning a classification model for segmentation. *ICCV*, 2003.

[45] G. Mori. Guiding model search using segmentation. *ICCV*, 2005.

[46] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. *CVPR*, 2:54–61, 2003.

[47] D. R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. PAMI*, 26 (5):530–549, 2004.

[48] K.V. Mardia and P.E. Jupp. *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley, 2009. ISBN 9780470317815.

[49] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[50] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, pages 2432–2439. IEEE, June 2010.

[51] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph based video segmentation. In *CVPR*, 2010.

[52] Wei-Chen Chiu and Mario Fritz. Multi-class video co-segmentation with a generative multi-video model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[53] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet process. *Journal of American Statistical Association*, 25(2):1566 – 1581, 2006.

[54] Tomasz Malisiewicz and Alexei A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, September 2007.

[55] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *PAMI*, 24(8):1026–1038, August 2002.

[56] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *PAMI*, 22(8), 2000.

[57] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, May 2002.

[58] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. ISSN 0920-5691.

[59] Richard Nock and Frank Nielsen. Statistical region merging. *PAMI*, 26:1452–1458, November 2004. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/TPAMI.2004.110. URL http://dx.doi.org/10.1109/TPAMI.2004.110.

[60] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. *CVPR*, 0:2294–2301, 2009.

[61] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pages 416–423, July 2001.

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

[63] Marco Andreetto, Lihi Zelnik-Manor, and Pietro Perona. Non-parametric probabilistic image segmentation. In *ICCV*, 2007.

[64] Giorgos Sfikas, Christophoros Nikou, and Nikolaos Galatsanos. Edge preserving spatially varying mixtures for image segmentation. *CVPR*, 0:1–7, 2008. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2008.4587416.

[65] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *PAMI*, 6(6):721–741, 1984.

[66] Alex Shyr, Trevor Darrell, Michael I. Jordan, and Raquel Urtasun. Supervised hierarchical Pitman-Yor process for natural scene segmentation. In *CVPR*, pages 2281–2288, 2011.

[67] Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *UAI*, pages 362–369, 2001.

[68] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[69] X. Ren and J. Malik. A probabilistic multi-scale model for contour completion based on image statistics. In *ECCV*, volume 1, pages 312–327, 2002.

[70] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *ICCV*, volume 1, pages 10–17, 2003.

[71] David R. Martin, Charless C. Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26:530–549, May 2004. ISSN 0162-8828.

[72] J. Pitman and M. Yor. The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900, 1997.

[73] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Tran. IP*, 3(5):625–638, September 1994.

[74] D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *IJCV*, 72(2): 195–215, 2007.

[75] M. Welling and K. Kurihara. Bayesian K-means as a "Maximization-Expectation" algorithm. In *SDM*, 2006.

[76] Z. Tu and S.C. Zhu. Image segmentation by data-driven Markov Chain Monte Carlo. *PAMI*, 24:657–673, 2002. ISSN 0162-8828.

[77] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic Photo Pop-up. In *ACM SIGGRAPH*, SIGGRAPH '05, pages 577–584, 2005.

[78] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289, 2001.

[79] Rüdiger Borsdorf, Nicholas J. Higham, and Marcos Raydan. Computing a nearest correlation matrix with factor structure. *SIAM J. Matrix Analysis App.*, 31(5):2603–2622, 2010. URL http://dblp.uni-trier.de/db/journals/siammax/siammax31.html#BorsdorfHR10.

[80] William M. Rand. Objective criteria for the evaluation of clustering methods. *JASA*, 66:846–850, 1971.

[81] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1106–1114. 2012. URL http://books.nips.cc/papers/files/nips25/NIPS2012_0534.pdf.

[82] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:1–8, 2008. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2008.4587503.

[83] Rodney A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artif. Intell.*, 17(1-3):285–348.

[84] D. Marr and H.K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. In *Proc. of the Royal Society of London, series B*, volume 200, pages 269–294, February 1978.

[85] Alex Pentland. Perceptual organization and the representation of natural form. *Artif. Intell.*, 28(3):293–331.

[86] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2137 – 2144, June 2006.

[87] A. Saxena, M. Sun, and A.Y. Ng. Learning 3-d scene structure from a single still image. In *ICCV workshop on 3dRR*, 2007.

[88] Erik B. Sudderth, Antonio Torralba, William T. Freeman, and Alan S. Willsky. Depth from familiar objects: A hierarchical model for 3d scenes. In *Proc. IEEE*

Computer Vision and Pattern Recognition (CVPR), pages 2410–2417. IEEE Computer Society, 2006.

[89] A. Saxena, M. Sun, and A.Y. Ng. Make3d: Learning 3d scene structure from a single still image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):824–840, 2009.

[90] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2012.

[91] L. Bo, X. Ren, and D. Fox. Depth Kernel Descriptors for Object Recognition. In *IROS*, September 2011.

[92] Chenxi Zhang, Liang Wang, and Ruigang Yang. Semantic segmentation of urban scenes using dense depth maps. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 708–721, 2010.

[93] Sunando Sengupta, Eric Greveson, Ali Shahrokni, and Philip HS Torr. Urban 3d semantic modelling using stereo vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[94] Julien P. C. Valentin, Sunando Sengupta, Jonathan Warrell, Ali Shahrokni, and Philip H. S. Torr. Mesh based semantic modelling for indoor and outdoor scenes. In *CVPR*, pages 2067–2074, 2013.

[95] H Koppula, Abhishek Anand, Thorsten Joachims, and Ashutosh Saxena. Semantic labeling of 3d point clouds for indoor scenes. *25th annual conference on neural information processing systems*, 2011.

[96] Albert S. Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Int. Symposium on Robotics Research (ISRR)*, Aug. 2011.

[97] Yutaka Ohtake, Alexander G. Belyaev, and Hans-Peter Seidel. An integrating approach to meshing scattered point data. In *Symposium on Solid and Physical Modeling*, pages 61–69, 2005.

[98] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, 1996.

[99] Sid Yingze Bao and Silvio Savarese. Semantic structure from motion. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2011.

[100] Sid Yingze Bao, Mohit Bagra, and Silvio Savarese. Semantic structure from motion with object and point interactions. In *IEEE Workshop on Challenges and Opportunities in Robot Perception (in conjunction with ICCV)*, 2011.

[101] Sid Yingze Bao, Mohit Bagra, Yu-Wei Chao, and Silvio Savarese. Semantic structure from motion with points, regions, and objects. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2012.

[102] Abhinav Gupta, Alexei A. Efros, and Martial Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *European Conference on Computer Vision(ECCV)*, 2010.

[103] David Changsoo Lee, Abhinav Gupta, Martial Hebert, and Takeo Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. *Advances in Neural Information Processing Systems (NIPS)*, 24, November 2010.

[104] Varsha Hedau, Derek Hoiem, and David Forsyth. Thinking inside the box: using appearance models and context based on room geometry. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV'10, pages 224–237, 2010.

[105] Bojan Pepik, Michael Stark, Peter Gehler, and Bernt Schiele. Teaching 3d geometry to deformable part models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[106] Michael Stark, Michael Goesele, and Bernt Schiele. Back to the future: Learning shape models from 3d cad data. In *British Machine Vision Conference (BMVC)*, 08/2010 2010.

[107] Jörg Liebelt and Cordelia Schmid. Multi-view object class detection with a 3d geometric model. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 1688–1695, jun 2010.

[108] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[109] Yee-Whye Teh, Dilan Gorur, and Zoubin Ghahramani. Stick-breaking construction for the indian buffet process. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 11, 2007.

[110] S. Zuffi, O. Freifeld, and M. J. Black. From pictorial structures to deformable structures. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3546–3553. IEEE, June 2012.

[111] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.