Bayesian Adaptive Superpixel Segmentation

Roy Uziel | Meitar Ronen | Oren Freifeld

Computer Science, Ben-Gurion University

uzielr@post.bgu.ac.il | meitarr@post.bgu.ac.il | orenfr@cs.bgu.ac.il

Abstract

Superpixels provide a useful intermediate image representation. Existing superpixel methods, however, suffer from at least some of the following drawbacks: 1) topology is handled heuristically; 2) the number of superpixels is either predefined or estimated at a prohibitive cost; 3) lack of adaptiveness. As a remedy, we propose a novel probabilistic model, self-coined Bayesian Adaptive Superpixel Segmentation (BASS), together with an efficient inference. BASS is a Bayesian nonparametric mixture model that also respects topology and favors spatial coherence. The optimization-based and topology-aware inference is parallelizable and implemented in GPU. Quantitatively, BASS achieves results that are either better than the state-of-the-art or close to it, depending on the performance index and/or dataset. Qualitatively, we argue it achieves the best results; we demonstrate this by not only subjective visual inspection but also objective quantitative performance evaluation of the downstream application of face detection. Our code is available at https://github.com/uzielroy/BASS.

1. Introduction

Superpixels [37], relatively-small image segments, form a compact intermediate image representation that is a key preprocessing step in many computer-vision tasks, *e.g.*, [6, 14, 18, 22, 24, 25, 27, 30, 33, 34, 35, 46, 54]. Existing superpixel methods, however, suffer from at least some of the following shortcomings: 1) Topological constraints are handled only via post-processing heuristics, and thus the resulting superpixels are no longer directly related to their objective; 2) the number of superpixels, K, is either defined manually or estimated at a usually-prohibitive computational cost (at least for methods of superixels with flexible shapes, as opposed to those that support only limited geometric primitives); 3) Limited adaptiveness to the global and local levels of image details, causing important details to be lost while wasting many superpixels in uninformative regions.

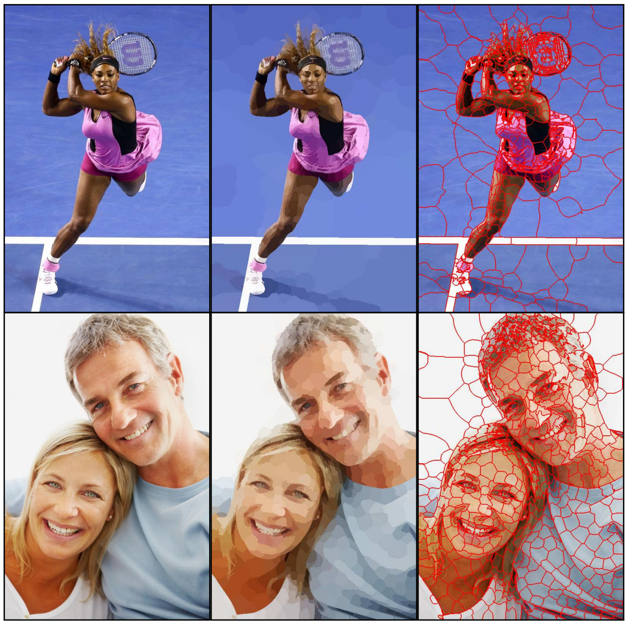

Figure 1: BASS adapts the size, shapes, and number of the superpixels to the image content, thereby providing an unsupervised and compact detail-preserving intermediate image representation. From left to right: original images; superpixels colored by their mean colors, superpixel boundaries. The same hyperparameters were used for both the images.

Here we propose a novel adaptive model of superpixels (Fig. 1) together with an efficient inference. This model, self-coined Bayesian Adaptive Superpixel Segmentation (BASS), is based on the Dirichlet-Process Gaussian Mixture Model (DPGMM). The latter is an important example of Bayesian Non-Parametric (BNP) mixture models. Such models provide a flexible and principled approach to the unsupervised-learning task of clustering and were first introduced to the computer-vision community over a decade ago; see, *e.g.* [45]. A key benefit of the BNP approach is that the inference procedure finds not only the clusters but also their number, thus letting these models adapt their complexity to the complexity of the data. Particularly, BASS automatically adapts K to the complexity of the image both globally and locally. BASS

This work was partially supported by the Lynn and William Frankel Center for Computer Science at BGU.

Table 1: A comparison of BASS with key methods. (a) SLIC [1], (b) reSEEDS [43], (c) ETPS [52], (d) TSP [9], (e) FSCSP [17], (f) BASS.

	(a)	(b)	(c)	(d)	(e)	(f)
Connectivity	×	✓	✓	✓	✓	✓
Parellelism	✓	×	×	×	✓	✓
Adaptive K	×	×	×	✓	×	✓
Anisotropic covs	×	✓	✓	×	✓	✓
Spatial coherence	×	×	✓	×	×	✓

differs from a DPGMM in that it respects topology (*e.g.*, it explicitly disallows a superpixel consisting of disconnected regions) and encourages spatial coherence (*i.e.*, it favors smoother inter-superpixel boundaries). The proposed efficient optimization-based inference is inspired by a DPGMM parallel sampler [8]. Seemingly, BASS' topological constraint prevents parallelizing the computations of that algorithm. A careful design, however, lets the proposed inference support massive parallelization without violating the topology, facilitating a massively-parallel GPU implementation. BASS' quantitative results are either better than the state-of-the-art or close to it, depending on the performance index and/or dataset. Qualitatively, we argue it achieves the best results and show it by not only subjective visual inspection but also objective *quantitative evaluation* of the performance of a face-detection downstream application. Finally, we show that BASS improves a deep-net-based segmentation.

2. Related Work

Below we focus on methods most related to ours; for more comprehensive surveys of superpixels, see [44, 49]. The most widely-used superpixel method is SLIC [1], which is based on spatio-color K-means clustering; recall K-means is a restricted case of the Gaussian Mixture Model (GMM). In fact, SLIC can be seen as a particular case of 4 categories: 1) methods based on spatio-color clustering using a K-component GMM; 2) methods with a parallelized implementation [36]; 3) methods whose model and inference do not respect topology and thus resort to post-processing heuristics to fix connectivity issues such as holes and disconnected regions; 4) methods of fixed K. The proposed method, BASS, belongs only to the first two categories. Examples of spatio-color GMMs are [7, 1, 9, 17]. Excluding [9], these methods fix K. Despite its varying K, [9] is a parametric model. Moreover, BASS, like [17], makes weaker assumptions on the covariances than [9] (see § 4) and thus, *e.g.*, has more flexibility in modeling the shape of each superpixel. Unlike [7, 1, 9, 17], BASS favors spatial coherence. It also differs from [7, 1], as well as from many other superpixel methods, in that it preserves connectivity. Several superpixel methods explicitly handle connectivity,

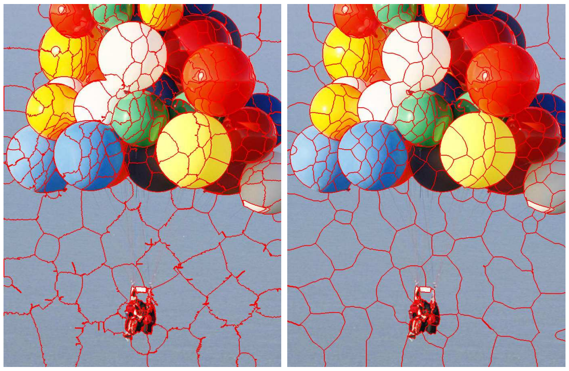

Figure 2: BASS without (left) and with (right) the Potts term.

e.g. [28, 47, 48, 52, 9, 17]; SEEDS [47] and ETPS [52] do it by considering updates only of boundary pixels while penalizing segmentations that harm connectivity. A newer version of SEEDS, reSEEDS [43], also models region compactness. Chang *et al.* [9] proposed a connectivity-constrained probabilistic model. The connectivity constraints in [47, 52, 9], however, complicate parallelization so these works propose only serial implementations. Freifeld et al. [17] improved on [9] by adding a flexible Bayesian prior on the spatial covariance and by proposing a parallelized implementation, though, unlike BASS, it was too conservative in the sense did not exploit the full extent of the potential parallelism.

With the exception of [9], all the works above use a user-defined K and do not alter it. This is problematic both locally and globally. Locally, different regions of the image might require different amounts of superpixels. As these methods are local in nature, since the global K is fixed, the number of superpixels they tend to allocate to each region depends almost entirely on its area, not content. Thus, fine details are lost while uninformative areas are over segmented. Globally, manually fixing K per image is infeasible for large datasets, while using the same K for all images is far from optimal. Thus, several methods proposed an adaptive K. The two main approaches for this are top-down and bottom-up. In the first, one starts with a few large segments that are gradually split into smaller ones. Works that do it via graph-based methods [39, 41, 15] usually suffer from high computational costs. Another top-down approach [32] incrementally partitions the image by horizontal and vertical splits; however, as noted in [1], computing the optimal paths is expensive. The bottom-up approach starts from many small segments, and then gradually merges them; *e.g.*, Alpert *et al.* [2] use a probabilistic method, Shen *et al.* [40] relied on DBSCAN, and in [38] merges are done via a model trained to detect feature similarity. Such an agglomerative approach may also be achieved via Gestalt grouping principles [26]. Some methods combine the top-down and bottom-up approaches using splits and merges. These allow to adapt not only K but also

```
Algorithm 1: BASS INFERENCE
  Input: Data: $(\boldsymbol{x}_i)_{i=1}^N$ where $\boldsymbol{x}_i = (\boldsymbol{l}_i, \boldsymbol{c}_i)$, $\boldsymbol{l}_i \in \mathbb{R}^2$,
         $\boldsymbol{c}_i \in \mathbb{R}^3$;
  Output: $K$, $(z_i)_{i=1}^N \subset \{1, ..., K\}$, $(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \boldsymbol{\pi}_j)_{j=1}^K$
  Init()
  $N_{\text{split}} \leftarrow 0$ , $N_{\text{merge}} \leftarrow 0$
  for $it \leftarrow 1$ to $N_{\text{iter}}$ do
      for every j do in parallel
          Update $(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \boldsymbol{\pi}_j)$ by Eqs. (8)–(12)
      for Row $\pmod 2 \in \{0, 1\}$ do in parallel
          for Col $\pmod 2 \in \{0, 1\}$ do in parallel
              Update $z_i$ by Eq. (13)
      if $it \pmod{32} == 0$ then
          $(z_i)_{i=1}^N$, $K \leftarrow$ Split($j$, $N_{\text{split}}$) by Alg. 2
          $N_{split}$++
      if $it \pmod{32} == 15$ then
          $(z_i)_{i=1}^N$, $K \leftarrow$ Merge($j$, $N_{\text{merge}}$) by Alg. 3
          $N_{\text{merge}}$++
```

superpixels' sizes. Such flexible-size methods can partition the image into meaningful segments by preserving details in complex regions on the one hand and not over-segmenting homogeneous areas on the other hand. An example for such methods is SMURFS [31]; however, SMURFS fails to retain compactness. Methods that achieve more compactness include [9, 12, 29]. Note that the superpixels from [12] are restricted to polygons. Our approach uses splits and merges to generate an adaptive amount of flexible-size superpixels in an efficient manner that allows preserving details. Due to its shape prior and favoring of spatial coherence, BASS maintains compactness. Thus, BASS enjoys both the benefits of compactness and detail-preserving segmentation.

The proposed efficient optimization-based inference is closely related to, and inspired by, a parallel DPGMM sampler proposed (unrelated to superpixels) in [8]. The key differences are that we replace their sampling with deterministic operations, that we respect topological constraints, and that we account for an additional prior term that favors spatial coherence. Finally, Table 1 summarizes some of the key differences between BASS and a few key methods.

3. Preliminaries

Let N be the number of pixels in the image, and let $\boldsymbol{x}_i = (\boldsymbol{l}_i, \boldsymbol{c}_i) \in \mathbb{R}^5$ denote the measurement associated with pixel i where $\boldsymbol{l}_i \in \mathbb{R}^2$ is the 2D location and $\boldsymbol{c}_i \in \mathbb{R}^3$ is the color. As is common in works on superpixels we use the *Lab* color space. Extensions to other color spaces, RGBD data, or even deep features, are straightforward and thus not discussed here. Spatio-color clustering methods, ours included, partition $(\boldsymbol{x}_i)_{i=1}^N$ into K disjoint groups, where

```
Algorithm 2: SPLIT
  Input: $j$, $N_{\text{split}}$
  Output: $(z_i)_{i=1}^N$, $K$
  switch $N_{\text{split}} \pmod 2$ do
      case 0 do
          $c_j^1 \leftarrow \mu_j^l + (0, 1)$ , $c_j^2 \leftarrow \mu_j^l - (0, 1)$
      case 1 do
          $c_j^1 \leftarrow \mu_j^l + (1, 0)$ , $c_j^2 \leftarrow \mu_j^l - (1, 0)$
  for $i \leftarrow 1$ to 2 do in parallel
      $d_j^i \leftarrow$ Run BFS to get the distance of each pixel
        from the center $c_j^i$
  Assign each pixel to the sub-superpixel closer to it.
  if $H_{\text{split}} > 1$(see Sup. Mat.) then
      Accept split and update $(z_i)_{i=1}^N$
  Update $(z_i)_{i=1}^N$
  Update $K$
```

z_i is the measurement-to-cluster assignment, known as the *label*, of $\boldsymbol{x}_i$, and thus also of pixel i. Cluster j is the set of measurements labeled as j; *i.e.*, $\{\boldsymbol{x}_i : z_i = j\}$. The associated superpixel is {pixel i : $z_i = j$}, and thus z_i is also a pixel-to-superpixel assignment. The number of clusters, $K \triangleq |\{j : j \in \boldsymbol{z}\}|$, is the number of unique elements in $\boldsymbol{z} = (z_i)_{i=1}^N$; *i.e.*, $\boldsymbol{z}$ is a K-region image segmentation. Let $\mathcal{Z}_K \triangleq \{1, \ldots, K\}^N$ denote the set of all segmentations into (no more than) K regions. A superpixel is called *connected*, if it is (path-) connected in the topological sense. It is called *simply-connected* if it is connected and has no holes. A segmentation is *valid* if each of its superpixels is connected. The pdf of a K-component GMM, in $\mathbb{R}^n$, has the form

$$p(\boldsymbol{x}; (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \pi_j)_{j=1}^K) = \sum\nolimits_{j=1}^K \pi_j \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{1}$$

where $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ is a Gaussian pdf (of mean $\boldsymbol{\mu}_j \in \mathbb{R}^n$ and an $n \times n$ covariance matrix $\boldsymbol{\Sigma}_j$) evaluated at $\boldsymbol{x} \in \mathbb{R}^n$ and where the weights $(\pi_j)_{j=1}^K$ form a convex combination. Let $\boldsymbol{\theta}_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ denote the parameters of Gaussian j. In a *Bayesian GMM*, $(\theta_j)_{j=1}^K$ and $(\pi_j)_{j=1}^K$ are viewed as random variables, drawn from a prior distribution, $p((\boldsymbol{\theta}_j, \pi_j)_{j=1}^K)$. A standard independence assumption implies the factorization

$$p((\boldsymbol{\theta}_j, \pi_j)_{j=1}^K) = p((\pi_j)_{j=1}^K) \prod\nolimits_{j=1}^K p(\boldsymbol{\theta}_j) \,. \tag{2}$$

Standard choices are a Normal-Inverse Wishart (NIW) distribution for $p(\boldsymbol{\theta}_j)$ and a Dirichlet *distribution* for $p((\pi_j)_{j=1}^K)$ [20]. The key reason is that, with these conjugate priors, the posterior distributions have the same functional form as the priors, and their updates from the priors are given in closed form via *sufficient statistics* [42, 20]. The Bayesian

```
Algorithm 3: MERGE
    Input: $j$, $N_{\text{split}}$
    Output: $(z_i)_{i=1}^N$, $K$
    Direction = [north,west,east,south]
    Choose the smallest adjacent neighbor from
      direction[$m \pmod 4$]
    if $H_{\text{merge}} > 1$ (see Sup. Mat.) then
        Add $j$ to the proposed merge group
    if Merge 3 or more adjacent SP was suggested then
        Update the proposed merge group, by keeping
          the smallest two
    Update $(z_i)_{i=1}^N$
    Update $K$
```

GMM inference is often done by alternating between

$$p((\pi_j)_{j=1}^K|\boldsymbol{z},(\boldsymbol{x}_i)_{i=1}^N)\prod\nolimits_{j=1}^K p((\boldsymbol{\theta}_j,\pi_j)|\boldsymbol{z},(\boldsymbol{x}_i)_{i=1}^N) \tag{3}$$

$$\text{and } p(\boldsymbol{z}|(\boldsymbol{\theta}_j,\pi_j)_{j=1}^K,(\boldsymbol{x}_i)_{i=1}^N)\,; \tag{4}$$

e.g., Gibbs sampling [21, 20] alternates between sampling $(\boldsymbol{\theta}_j,\pi_j)_{j=1}^K$ using Eq. (3) and sampling $\boldsymbol{z}$ given Eq. (4). By conditional independence, the N labels $(z_i)_{i=1}^N$ can be sampled in parallel. Similarly, the K parameters $(\boldsymbol{\theta}_j)_{j=1}^K$ can be sampled in parallel, and at the same time with the sampling of the weight vector $(\pi_1,\dots,\pi_K)$. Another approach (more relevant to the proposed method), is Iterated Conditional Modes (ICM) [5], a greedy optimization method with fast convergence. Its computational structure and parallelism are similar to Gibbs sampling, except that instead of conditional sampling, one uses conditional modes.

A DGPMM [16, 3] is a Bayesian nonparametric extension of the Bayesian GMM. Loosely speaking, it entertains the notion of infinitely-many Gaussians. Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_j)_{j=1}^\infty$ and $\boldsymbol{\pi} = (\pi_1,\pi_2,\dots)$ such that $\pi_j > 0\ \forall j$, and $\sum_{j=1}^\infty \pi_j = 1$. Given $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, the $\boldsymbol{x}_i$'s are assumed to be *i.i.d*. draws from $p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\pi}) = \sum_{j=1}^\infty \pi_j\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$. In analogy to the Bayesian GMM, the infinite-length $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ themselves are assumed to be drawn (independently) from their own prior distributions: the weights, $\boldsymbol{\pi}$, are drawn using a stick-breaking process (whose details are inessential to understanding our paper) with a so-called concentration parameter, α, while the parameters, $(\boldsymbol{\theta}_j)_{j=1}^\infty$, are *i.i.d*. draws from their prior $p(\boldsymbol{\theta}_j)$, typically an NIW distribution.

Let z_i be the index of the Gaussian from which $\boldsymbol{x}_i$ was drawn, and let $\boldsymbol{z} = (z_i)_{i=1}^N$. Recall $K = |\{j : j \in \boldsymbol{z}\}|$ and note that $K \leq N$. By possibly renaming indices, we may assume without loss of generality that $\{j : j \in \boldsymbol{z}\} = \{1,2,\dots,K\}$. Here K is random and is determined by α: the higher α is, the larger the expected value of K is.

As in the finite case, one often alternates between $p(\boldsymbol{\theta},\boldsymbol{\pi}|\boldsymbol{z},(\boldsymbol{x}_i)_{i=1}^N)$ and $p(\boldsymbol{z}|\boldsymbol{\theta},\boldsymbol{\pi},(\boldsymbol{x}_i)_{i=1}^N)$. One efficient sampling-based DPGMM inference method (unrelated to superpixels) was proposed in [8]. This Markov-Chain Monte-Carlo method alternates between 1) changes to K using splits and merges of clusters and 2) given K, Gibbs-sampling GMM inference. Importantly, within each iteration, all computations in their algorithm, including the splits and merges, are massively parallelizable. In fact, the computations can be not only parallelized but also distributed [11].

Finally, we touch upon connectivity. Changing a label in a valid segmentation might break connectivity. A point whose label can be changed without breaking connectivity is called a *simple point* [9]; see also our **Sup. Mat.** It can be shown that the answer to the question whether a pixel is a simple point or not is a function of only the labels of its neighbors in a 3×3 neighborhood when considering simply-connected regions, or a 5×5 neighborhood when considering connected regions. For more details, see [9].

4. BASS: Model and Inference

BASS is a variant of the DPGMM. Unlike the latter, BASS: 1) disallows (topologically-) invalid segmentations; 2) favors spatially-coherent segmentations. More formally, BASS differs from the standard DPGMM in that the labels, $\boldsymbol{z}$, given the weights $\boldsymbol{\pi}$, are no longer conditionally-independent. Rather, BASS assumes a 2-step process for generating $\boldsymbol{z}$ given $\boldsymbol{\pi}$. First, N temporary labels, $\widetilde{\boldsymbol{z}} = (\widetilde{z}_1,\dots,\widetilde{z}_N)$, are drawn *i.i.d*. from $\boldsymbol{\pi}$. Next, the actual labels, $\boldsymbol{z} = (z_1,\dots,z_N)$, are defined via the action of $\boldsymbol{\Pi}$, an $N \times N$ random permutation matrix (namely, $\boldsymbol{\Pi}$ has one entry of 1 in each row and each column and zeros elsewhere), on $\widetilde{\boldsymbol{z}}$; *i.e*., $\boldsymbol{z} = \boldsymbol{\Pi}\widetilde{\boldsymbol{z}}$ where $\boldsymbol{\Pi}$ is drawn, given $\widetilde{z}$, by

$$p(\boldsymbol{\Pi}|\widetilde{\boldsymbol{z}}) \propto \mathbb{1}_{\text{valid}}(\boldsymbol{z})\exp\left(-\beta\sum\nolimits_{i\sim i'}\mathbb{1}_{z_i\neq z_{i'}}(\boldsymbol{z})\right). \tag{5}$$

We now explain Eq. (5). If A is an event, then $\mathbb{1}_A$ is its indicator function. The event valid stands for a valid segmentation. The notation $i \sim i'$ indicates that pixels i and i' are neighbors according to to some predefined graph G, *e.g*., a regular 2D grid with a 4- or 8-connectivity. Thus, BASS assigns zero probability to any invalid segmentation. The role of $\exp\left(-\beta\sum_{i\sim i'}\mathbb{1}_{z_i\neq z_{i'}}(\boldsymbol{z})\right)$, which has the form of the Potts model [50], is encouraging spatial coherence of $\boldsymbol{z}$; see, *e.g*., Fig. 2. The parameter $\beta > 0$ controls the importance of that term. As for modeling each Gaussian, we follow other works on spatio-color GMMs [7, 1, 9, 17] and assume that 1) given z_i, color and location are conditionally independent; 2) the color covariance is isotropic. This implies:

$$\begin{gathered}\mathcal{N}(\boldsymbol{x}_i;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j) = \mathcal{N}(\boldsymbol{l}_i;\boldsymbol{\mu}_{j,\boldsymbol{l}},\boldsymbol{\Sigma}_{j,\boldsymbol{l}})\mathcal{N}(\boldsymbol{c}_i;\boldsymbol{\mu}_{j,\boldsymbol{c}},\sigma_{j,\boldsymbol{c}}^2\boldsymbol{I}_{3\times3}) \\ \boldsymbol{\mu}_j = \begin{bmatrix}\boldsymbol{\mu}_{j,\boldsymbol{l}} \\ \boldsymbol{\mu}_{j,\boldsymbol{c}}\end{bmatrix}, \boldsymbol{\Sigma}_j = \begin{bmatrix}\boldsymbol{\Sigma}_{j,\boldsymbol{l}} & \boldsymbol{0}_{2\times3} \\ \boldsymbol{0}_{3\times2} & \sigma_{j,\boldsymbol{c}}^2\boldsymbol{I}_{3\times3}\end{bmatrix}.\end{gathered} \tag{6}$$

Differently from [7, 1, 9], however, our Gaussians are more flexible since 1) we do not assume the spatial co-

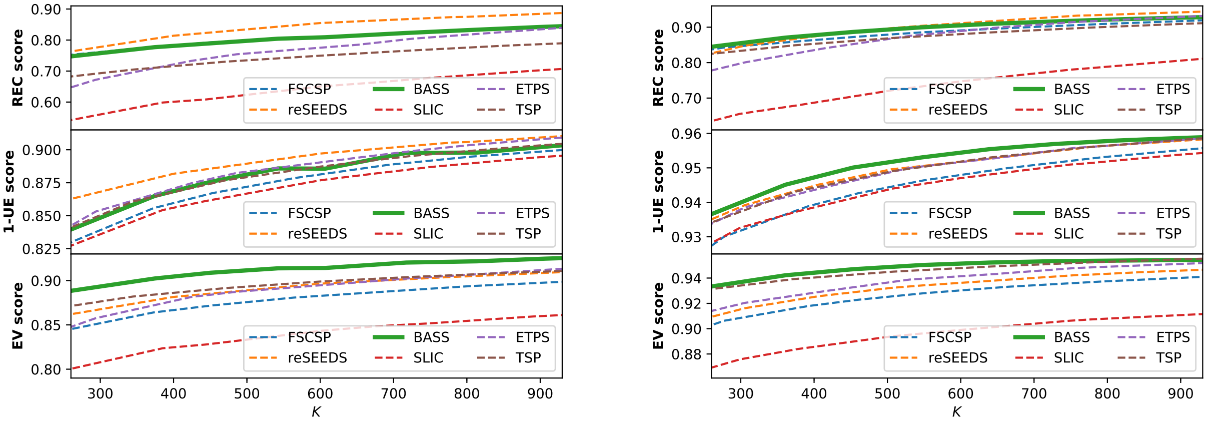

Figure 3: Comparing methods for a range of 10 different K values on the BSD (left) and SBD (right) datasets. For each such value, all methods were initialized, and ended with, about the same K (see text). For all 3 metrics, the higher the better.

variances, $\boldsymbol{\Sigma}_{j,\boldsymbol{l}}$'s, are isotropic and 2) we treat all 4 quantities, $(\boldsymbol{\mu}_{j,\boldsymbol{l}}, \boldsymbol{\Sigma}_{j,\boldsymbol{l}}, \boldsymbol{\mu}_{j,\boldsymbol{c}}, \sigma^2_{j,\boldsymbol{c}})$ as latent, random, and possibly-dependent on j. Particularly, we place an NIW prior on $(\boldsymbol{\mu}_{j,\boldsymbol{l}}, \boldsymbol{\Sigma}_{j,\boldsymbol{l}})$ and a multivariate Normal-Inverse-Gamma (NIG) prior on $(\boldsymbol{\mu}_{j,\boldsymbol{c}}, \sigma^2_{j,\boldsymbol{c}})$. In contrast, in [7, 1, 9] $(\boldsymbol{\Sigma}_{j,\boldsymbol{l}}, \sigma^2_{j,\boldsymbol{c}})$ were assumed to be known, $\boldsymbol{\Sigma}_{j,\boldsymbol{l}}$ was isotropic, and the unknown variables $\boldsymbol{\mu}_{j,\boldsymbol{l}}$ and $\boldsymbol{\mu}_{j,\boldsymbol{c}}$ were treated as deterministic. Our modeling of a Gaussian is akin to [17] except that in [17]: 1) the $\sigma^2_{j,\boldsymbol{c}}$'s were assumed to be known and identical to each other; 2) instead of an NIW prior on $(\boldsymbol{\mu}_{j,\boldsymbol{l}}, \boldsymbol{\Sigma}_{j,\boldsymbol{l}})$, they used a uniform prior on both $\boldsymbol{\mu}_{j,\boldsymbol{l}}$ and $\boldsymbol{\mu}_{j,\boldsymbol{c}}$ and an Inverse-Wishart (not to be confused with NIW) on $\boldsymbol{\Sigma}_{j,\boldsymbol{l}}$. These differences from [17] are subtle, and, if we were to treat K as fixed (as [17] did), would be almost immaterial. However, when we infer K, the NIW prior and NIG priors are essential for computing the associated Hastings ratios.

Most superpixel methods, ours included, specify how to weight location versus color; consequently, the results of most methods are often very sensitive to the weight's value. The fact we, like [17], estimate the spatial covariance (as opposed to assuming it is known) gives us some robustness to picking the "wrong" weight. The Potts term further increases this robustness beyond that of [17].

Given the data, $(\boldsymbol{x}_i)_{i=1}^N$, we seek to infer the latent K, as well as the latent $\boldsymbol{z} = (z_i)_{i=1}^N$ where each $z_i \in \{1, .., K\}$. A natural question is whether (and how) an efficient DPGMM inference method can be used for the more complicated BASS model. Our proposed inference is inspired by [8]. The key differences are as follows: 1) we replace all their sampling operations with deterministic operations. Particularly, given K, we replace their Gibbs sampling with ICM [5]. Likewise, when proposing splits and/or merges, instead of deciding on acceptance/rejection by flipping a coin biased by Hastings' ratio, we deterministically accept the proposal if and only if the ratio exceeds 1. The rationale is that this greedy approach converges faster than sampling. 2) Both our label updates and our proposed splits/merges respect topological constraints. 3) Our label updates also take into account spatial coherence. We now provide the details for the proposed inference, summarized in Alg. 1. Since $p(\boldsymbol{\theta}, \boldsymbol{\pi} | \boldsymbol{z}, (\boldsymbol{x}_i)_{i=1}^N)$ depends on $\boldsymbol{z}$ only through $(n_j)_{j=1}^K$ (where $n_j = \sum_{i=1} \mathbb{1}_{z_i=j}(z_i)$), which are invariant under the action of $\boldsymbol{\Pi}$, working with $p(\boldsymbol{\theta}, \boldsymbol{\pi} | \boldsymbol{z}, (\boldsymbol{x}_i)_{i=1}^N)$ in BASS is the same as in DPGMM. Working with $p(\boldsymbol{z} | \boldsymbol{\theta}, \boldsymbol{\pi}, (\boldsymbol{x}_i)_{i=1}^N)$, however, is harder in BASS since, due the loss of conditional independence, $p(\boldsymbol{z} | \boldsymbol{\theta}, \boldsymbol{\pi}, (\boldsymbol{x}_i)_{i=1}^N) \neq \prod_{i=1}^N p(z_i | \boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{x}_i)$. This difficulty affects not only the label updates but also the splits/merges. It has no effect on estimating $(\boldsymbol{\theta}_j)_{j=1}^K$ and the cluster weights. The required computations are as follows. Consider first the case of a fixed K. The priors are:

$$\begin{aligned} p(\boldsymbol{\mu}_{j,\boldsymbol{l}}, \boldsymbol{\Sigma}_{j,\boldsymbol{l}}) =& \mathrm{NIW}(\boldsymbol{\mu}_{j,\boldsymbol{l}}, \boldsymbol{\Sigma}_{j,\boldsymbol{l}}; \boldsymbol{m}_{j,\boldsymbol{l}}, \kappa_{j,\boldsymbol{l}}, \boldsymbol{\Lambda}_{j,\boldsymbol{l}}, \nu_{j,\boldsymbol{l}})\,; \\ p(\boldsymbol{\mu}_{j,\boldsymbol{c}}, \sigma^2_{j,\boldsymbol{c}}) =& \mathrm{NIG}(\boldsymbol{\mu}_{j\ \boldsymbol{c}}, \sigma^2_{j,\boldsymbol{c}}; \boldsymbol{m}_{j,\boldsymbol{c}}, \kappa_{j,\boldsymbol{c}}, a_{j,\boldsymbol{c}}, b_{j,\boldsymbol{c}})\,; \\ p((\boldsymbol{\pi}_j)_{j=1}^K) =& \mathrm{Dir}((\boldsymbol{\pi}_j)_{j=1}^K; \alpha)\,. \end{aligned} \tag{7}$$

Parameter updates. The conditional modes are (see [20])

$$\left(\frac{\alpha_j^* - 1}{\sum_{j'=1}^K \alpha_{j'}^* - K} \right)_{j=1}^K = \operatorname*{arg\,max}_{(\pi_j)_{j=1}^K} p((\pi_j)_{j=1}^K | \boldsymbol{z}, (\boldsymbol{x}_i)_{i=1}^N) \tag{8}$$

$$\frac{\boldsymbol{\Lambda}^*_{j,\boldsymbol{l}}}{\nu^*_{j,\boldsymbol{l}}+3} = \operatorname*{arg\,max}_{\boldsymbol{\Sigma}_{j,\boldsymbol{l}}} p(\boldsymbol{\Sigma}_{j,\boldsymbol{l}} | \boldsymbol{z}, (\boldsymbol{x}_i)_{i=1}^N) \tag{9}$$

$$\boldsymbol{m}^*_{j,\boldsymbol{l}} = \operatorname*{arg\,max}_{\boldsymbol{\mu}_{j,\boldsymbol{l}}} p(\boldsymbol{\mu}_{j,\boldsymbol{l}} | \boldsymbol{\Sigma}_{j,\boldsymbol{l}}, \boldsymbol{z}, (\boldsymbol{x}_i)_{i=1}^N) \tag{10}$$

$$\frac{b^*}{a^*+5/2} = \operatorname*{arg\,max}_{\sigma^2_{j,\boldsymbol{c}}} p(\sigma^2_{j,\boldsymbol{c}} | \boldsymbol{z}, (\boldsymbol{x}_i)_{i=1}^N) \tag{11}$$

$$\boldsymbol{m}^*_{j,\boldsymbol{c}} = \operatorname*{arg\,max}_{\boldsymbol{\mu}_{j,\boldsymbol{c}}} p(\boldsymbol{\mu}_{j,\boldsymbol{c}} | \sigma^2_{j,\boldsymbol{c}}, \boldsymbol{z}, (\boldsymbol{x}_i)_{i=1}^N) \tag{12}$$

where the closed-form expressions for the "starred" quantities on the LHS of Eqs. (8)–(12) appear in our **Sup. Mat.**

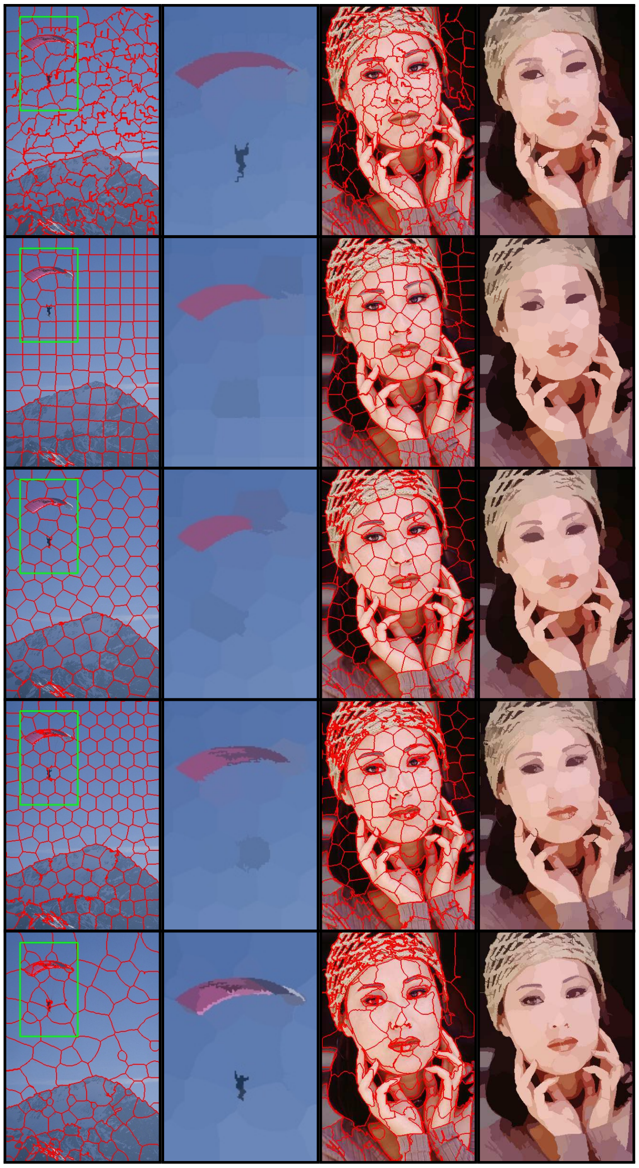

Figure 4: Superpixels. Rows, top to bottom: reSEEDS [43]; ETPS [52]; TSP [9]; FSCSP [17]; BASS. All methods were initialized, and ended with, $K \approx 150$. See text for details. Columns 1 and 3 show superpixel boundaries overlaid over original images. Columns 2 and 4 show mean colors. BASS captures fine details (*e.g.* parachutist and parachute; the women's nostrils) while keeping regular boundaries.

A single label update. Fix $z_{/i} \triangleq (z_{i'} : i' \neq i)$. If pixel i is a simple point then z_i is updated by the conditional mode of $p(z_i|(\boldsymbol{\theta}_j, \pi_j)_{j=1}^K, (\boldsymbol{x}_i)_{i=1}^N, z_{/i})$:

$$z_i = \operatorname*{arg\,max}_{j:\exists i' \in \eta_4(i) \text{ s.t. } z_{i'}=j} \pi_j \mathcal{N}(\boldsymbol{l}_i; \boldsymbol{\mu}_j^{\boldsymbol{l}}, \boldsymbol{\Sigma}_j^{\boldsymbol{l}}) \mathcal{N}(\boldsymbol{c}_i; \boldsymbol{\mu}_j^{\boldsymbol{c}}, \sigma_j^2 \boldsymbol{I}_{3\times 3}) \times \exp\left(-\beta \sum\nolimits_{i'':i\sim i''} \mathbb{1}_{j\neq z_{i''}}(\boldsymbol{z})\right) \quad (13)$$

where the last term penalizes label j according to how many neighboring pixels of pixel i have a label other than j.

Parallel label updates. Naive parallel label updates can, and usually do, break connectivity: even if two neighbors are simple points, updating *both* their labels simultaneously can violate connectivity. This difficulty prevented parallel label updates in methods such as [9, 52, 43]. However, as noted in [17], a large portion of the labels of simple points *can* be updated in parallel, provided the points are sufficiently far from each other; however, [17] was too conservative, spacing these points needlessly far from each other. This led to parallelization over only $N/9$ label updates. We note here that any pair of simple points that are at least one pixel apart from each other can be updated in parallel. This implies that $N/4$ label updates can be considered at once. This parallelization scheme is also consistent with the Potts term when the latter uses a 3×3 neighborhood; empirically, we find this scheme obtains good results even if it uses a larger neighborhood (despite the violation of conditional independence).

Changing K via Splits and Merges. At the core of the algorithm from [8], for each cluster one maintains a pair of sub-clusters, symbolically denoted l and r ("left" and "right"). In other words, for each Gaussian j with parameters $\boldsymbol{\theta}_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ and weight π_j, one also maintains two Gaussians, with parameters $\boldsymbol{\theta}_{jl} = (\boldsymbol{\mu}_{jl}, \boldsymbol{\Sigma}_{jl})$ and $\boldsymbol{\theta}_{jr} = (\boldsymbol{\mu}_{jr}, \boldsymbol{\Sigma}_{jr})$, and weights (π_{jl}, π_{jr}). During inference the corresponding sufficient statistics are computed as they are computed for their parent Gaussian. In [8], splits and merges were proposed using draws from proposal distributions and then were accepted or rejected according to the corresponding Hastings ratio. The expressions for Hastings ratios (derived in [8]), for splits and merges, denoted by H_{split} and H_{merge}, are too lengthy to be included here but appear in our **Sup. Mat.** The take-home message is that a higher value of α encourages more splits and less merges and vice versa. Our connectivity-aware parallel splits and merges algorithms are summarized in Alg. 2 and Alg. 3. First, we discuss splits. For each suprpixel we run a Breadth-First Search (BFS) starting from two pixels, called roots, near its centroid. This gives graph distances from all the superpixel's pixels to the roots. We set the subcluster label of each pixel to the root closer to it. By construction, each subcluster remains (simply-) connected. As in [8], the split proposals are parallelizable. Once a split is proposed (deterministically), we accept it (again, deterministically) if and only if its Hastings ratio, H_{split}, exceeds 1. For merges, we consider only pairs of adjacent superpixels (to maintain connectivity), proposing them in a deterministic order. To enable parallel merges, one must ensure no 3 or more superpixels are merged together. Here our situation scales better than [8]: since every superpixel has only few adjacent superpixels, this check is fast. A legit merge is accepted according to its Hastings ratio, H_{merges}. While we disallow breaking *simple-*

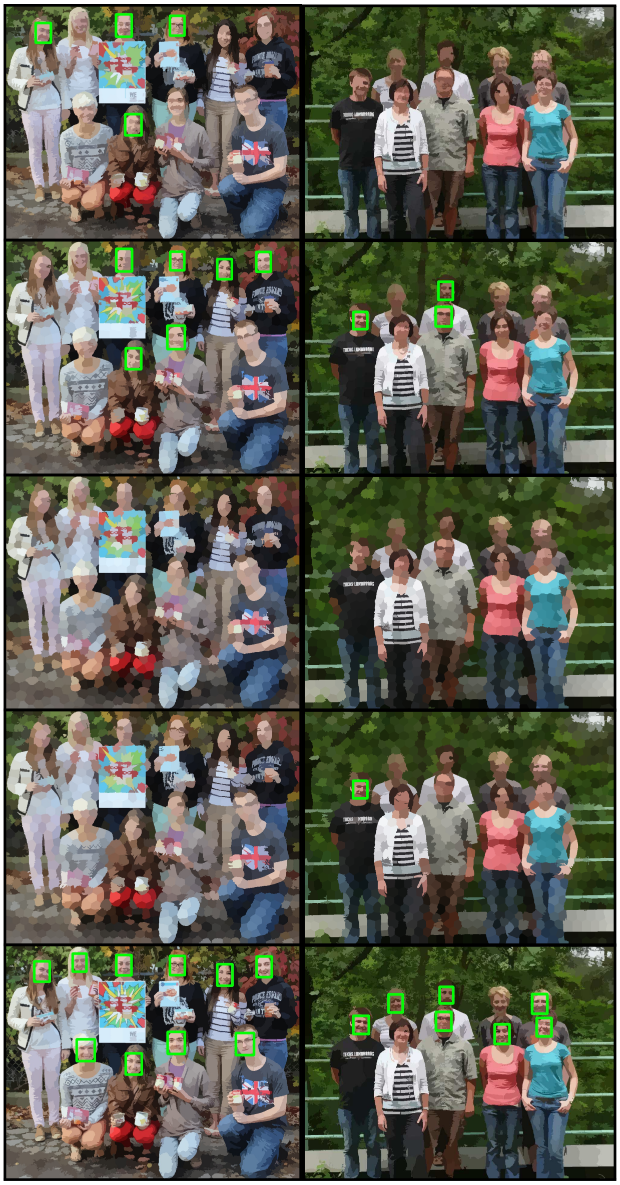

Figure 5: Face-detection examples. Rows, top to bottom: reSEEDS [43]; ETPS [52]; TSP [9]; FSCSP [17]; BASS. All methods were initialized, and ended with, $K \approx 1050$. See text for details. Detected faces are shown in a green. Please see our **Sup. Mat.** for examples in a higher resolution.

connectivity during label updates, we allow it during merges (while still disallowing breaking regular connectivity).

Table 2: Success rates of a face detector applied to superpixels of different methods. See text for more details.

Method \ $SP\#$	550	1100
SLIC [1]	10.543%	16.298%
reSEEDS [43]	17.102%	25.733%
ETPS [52]	11.390%	22.553%
TSP [9]	5.210%	9.817%
FSCSP [17]	14.209%	10.854%
BASS	**32.123%**	**45.469%**

Hyperparamters and Initialization. There are 2 important user-defined parameters, α (the most important one) and β; the others are fixed as explained below. Let K_0 be the initial number of superpixels and let K_{final} be their final number. A high α encourages splits and discourages merges and vice versa. Part of BASS' elegance is that K_0 hardly affects K_{final}; for a fixed α, initializing with different K_0's (e.g., ranging between 200 and 800) on the same image produce similar K_{final}'s. As for β, a high β encourages shorter inter-superpixel boundaries. The K_0 superpixels are initialized as squares in a regular grid. Let $A_0 = N/K_0$ denote their average size. For every j, we set $\nu_{j,\boldsymbol{l}} = \eta A_0$ where $\eta = 50$ in all our experiments. All the $\mathbf{\Lambda}_{j,\boldsymbol{l}}$'s are initially set to $A_0^2 \boldsymbol{I}_{2\times2}$. The higher η is (and thus $\nu_{j,\boldsymbol{l}}$), the more likely the inferred (and usually anisotropic) $\mathbf{\Sigma}_{j,\boldsymbol{l}}$ is to be isotropic (circular). The value of η has little impact on the segmentation's quality: if it is, *e.g.*, too high, then BASS, instead of trying to fit a circle-like supeprixel to an elongated shape, will simply split the superpixel. Let $L_j = \#\text{splits} - \#\text{merges}$ of superpixel j. To encourage superpixels of the same L to have similar sizes, we adjust the prior on the fly by setting $A_{j,L} = \frac{N}{2^{L_j}K}$, $\Lambda_{j,\boldsymbol{l}} = A_{j,L}^2 \boldsymbol{I}_{2\times2}$, and $\nu_{j,\boldsymbol{l}} = \eta A_{j,L}$. The other hyperparameters are set as follows. To make the Normal parts of the NIW and NIG almost uniform, we set $\kappa_{j,\boldsymbol{l}}$ and $\kappa_{j,\boldsymbol{c}}$ to a small number close to zero (*e.g.*, 0.001). As for the Inverse-Gamma, since $E(\sigma_{j,\boldsymbol{l}}^2) = (b_{j,\boldsymbol{c}}/(a_{j,\boldsymbol{c}} - 1))$ and $\text{VAR}(\sigma_{j,\boldsymbol{l}}^2) = b_{j,\boldsymbol{c}}^2/((a_{j,\boldsymbol{c}} - 1)^2(a_{j,\boldsymbol{c}} - 2))$, we set $a_{j,\boldsymbol{c}}$ to be very large, and then set $(b_{j,\boldsymbol{c}}/(a_{j,\boldsymbol{c}} - 1)) \approx 10$. The effect is that $p(\sigma_{j,\boldsymbol{c}}^2)$ is concentrated near 10.

Convergence and Implementation. Theoretical convergence analysis is hard; empirically, the greedy algorithm converges fast. Our current GPU implementation was written in PyTorch; for $K_0 = 1000$ and a 481-by-321 image, it runs in ∼2 [sec] on a good graphics card. Profiling suggests that the implementation suffers from some significant overhead (due to certain deep-learning-related functionalities that are unneeded here); we speculate that a pure CUDA implementation (as was done in [17]) will be much faster.

5. Experiments and Results

The Competing Methods. We compared BASS with ETPS [52] and reSEEDS [43] that represent the state-of-the-art (*e.g.* [44]), with the popular SLIC [1], and with TSP [9] and FSCSP [17] that share some similarities with BASS. For SLIC/ETPS/reSEEDs, we used the optimal parameters reported in [44]; for TSP/FSCSP we used the defaults. For BASS, in all the experiments below we set $\beta = 1.4$, and

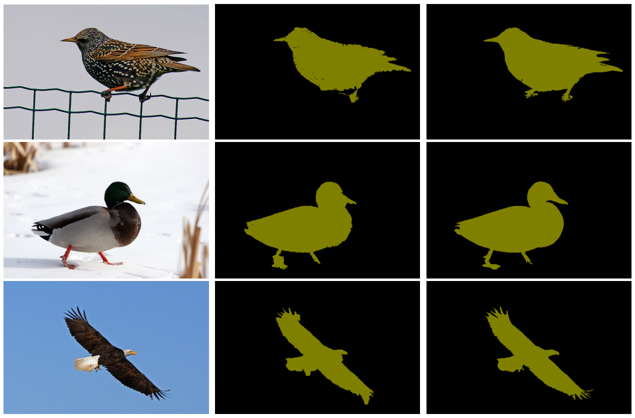

Figure 6: Object segmentation using DeepLab [10]+BI [19] using superpixels. Left: Original images. Middle: results using SLIC. Right: results using BASS.

then tuned only α (using the same nominal value of α in all the experiments). All methods were initialized to the same K_0. For a fair comparison, we made BASS, after some iterations, converge back to $K_{\text{final}} \approx K_0$ by automatically increasing/decreasing α as needed. TSP [9] performed only few splits/merges and thus usually had $K_{\text{final}} \approx K_0$.

Quantitative Evaluation. We evaluated the methods on two standard benchmarks, BSD [23] and SBD [4], using the evaluation framework proposed in [44]. The SBD dataset is considered more difficult than BSD. We used 3 standard evaluation metrics: Boundary Recall (Rec), Undersegmentation Error (UE) and Expected Variance (EV). The first two, Rec and UE, rely on ground-truth segmentation to measure boundary adherence. The latter, EV, is independent of ground truth and quantifies the amount of image variance explained by the model. There is a trade-off between Rec and compactness: segmentations with a high Rec tend to have lower compactness [44]. The results are summarized in Fig. 3. On BSD: BASS outperforms the others in the EV metric; in the UE metric, BASS is close to the state of the art; in the Rec metric, BASS is second only to reSEEDS. Note that, unlike reSEEDs, BASS' high Rec does not come at the expense of preserving regular shape (Fig. 4). Interestingly, on the more complicated SBD dataset, BASS performs even better, and almost uniformly dominates in all metrics.

Face Detection. We took group images from [51]. For each image, we counted the number of faces detected by a state-of-the-art face detector [53] when run on the original image. Next, for each superpixel method, we created a mean-color image, by coloring each superpixel with its mean color; see Fig. 5 for examples. Then we computed the Success Rate (SR) of a method, per image, by dividing the number of faces detected on the mean-color images by the number found on the original images. Table 2 shows BASS outperforms the others dramatically. In fact, for both $K = 550$ and $K = 1100$, the median SR for all other methods was zero; *i.e.*, in most images they missed all the faces. In contrast, BASS scored median values of 18.182% and 41.667%.

Table 3: Ablation study – results on the BSD dataset. (a) no Potts' term; (b) no splits/merges; (c) no Potts & no splits/merges; (d) DPGMM (*i.e.*, *i.i.d.* labels: no connectivity & no Potts); (e) BASS.

SP #		(a)	(b)	(c)	(d)	(e)
950	*Rec*	0.858	0.843	**0.874**	0.682	0.849
	1-UE	0.905	0.902	0.902	0.746	**0.908**
	EV	0.926	0.916	0.919	0.764	**0.931**
600	*Rec*	0.766	0.814	**0.887**	0.574	0.817
	1-UE	0.878	0.887	0.887	0.599	**0.890**
	EV	0.906	0.906	0.909	0.685	**0.923**
350	*Rec*	0.740	0.764	**0.810**	0.454	0.782
	1-UE	0.856	0.852	0.858	0.430	**0.861**
	EV	0.896	0.884	0.892	0.545	**0.910**

Object Segmentation. As another example of the applications of BASS, we considered the downstream task of object segmentation. We used the pre-trained *deep nets* of DeepLab [10] and the Bilateral Inception (BI) model [19], but during test, replaced the input superpixel segmentation from the popular SLIC [36] to BASS and evaluated the results on the Pascal-VOC2012 [13] test set. The baseline net (*i.e.*, no superpixels, hence no BI), DeepLab, scored a mean IoU of 68.9. DeepLab+BI+SLIC scored 74.42. DeepLab+BI+BASS (set to have a similar K to SLIC) scored 74.68. For typical results, see Fig. 6.

Ablation study. This experiment studies the contribution of each part of the model. It also highlights how much BASS differs from a DPGMM. As Table 3 shows, dropping either Potts' term or the splits/merges usually hurts performance. Dropping both hurts two indices and improves only the boundary recall (but then boundaries become too irregular). Dropping Potts' term and connectivity gives back DPGMM, whose results are by far the worst. In other words, the proposed modifications to the generic DPGMM were critical in order to create a variant suitable for superpixels.

6. Conclusion

BASS yields state-of-the-art superpixels while adapting to image content, as we showed in our experiments. The face-detection experiment provides a quantitative support to our qualitative claim that BASS preserves details better than other methods. Finally, unsupervised and task-independent compact intermediate image representations, such as the superpixels presented in this paper, are complementary to supervised and task-dependent methods such as most deep-learning (DL) methods. In fact, superpixels can even improve the latter. For example, precomputed superpixels can improve DL-based segmentation (as we showed here) or to drastically speed up DL computations (*e.g.*, by treating superpixels as nodes on a small graph).

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 2, 4, 5, 7

[2] Sharon Alpert, Meirav Galun, Achi Brandt, and Ronen Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE transactions on pattern analysis and machine intelligence*, 34(2):315–327, 2012. 2

[3] Charles E Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, pages 1152–1174, 1974. 4

[4] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011. 8

[5] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(3):259–279, 1986. 4, 5

[6] Sylvain Boltz, Frank Nielsen, and Stefano Soatto. Earth mover distance on superpixels. In *2010 IEEE International Conference on Image Processing*, pages 4597–4600. IEEE, 2010. 1

[7] Chad Carson, Megan Thomas, Serge Belongie, Joseph M Hellerstein, and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. In *International conference on advances in visual information systems*, pages 509–517. Springer, 1999. 2, 4, 5

[8] Jason Chang and John W Fisher. Parallel sampling of DP mixture models using sub-cluster splits. In *NIPS*, 2013. 2, 3, 4, 5, 6

[9] Jason Chang, Donglai Wei, and John W Fisher. A video representation using temporal superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2058, 2013. 2, 3, 4, 5, 6, 7, 8

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *TPAMI*, 40(4):834–848, 2017. 8

[11] Or Dinari, Yu Angel, Oren Freifeld, and John W Fisher III. Distributed MCMC inference in Dirichlet process mixture models using Julia. In *International Symposium on Cluster, Cloud and Grid Computing (CCGRID) Workshop on High Performance Machine Learning Workshop*, 2019. 4

[12] Liuyun Duan and Florent Lafarge. Image partitioning into convex polygons. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3119–3127, 2015. 3

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 8

[14] Amal Farag, Le Lu, Holger R Roth, Jiamin Liu, Evrim Turkbey, and Ronald M Summers. A bottom-up approach for pancreas segmentation using cascaded superpixels and (deep) image patch labeling. *IEEE Transactions on Image Processing*, 26(1):386–399, 2017. 1

[15] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004. 2

[16] Thomas S Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1973. 4

[17] Oren Freifeld, Yixin Li, and John W Fisher. A fast method for inferring high-quality simply-connected superpixels. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 2184–2188. IEEE, 2015. 2, 4, 5, 6, 7

[18] Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto. Class segmentation and object localization with superpixel neighborhoods. In *2009 IEEE 12th international conference on computer vision*, pages 670–677. IEEE, 2009. 1

[19] Raghudeep Gadde, Varun Jampani, Martin Kiefel, Daniel Kappler, and Peter V Gehler. Superpixel convolutional networks using bilateral inceptions. In *ECCV*, pages 597–613. Springer, 2016. 8

[20] Andrew Gelman, Hal S Stern, John B Carlin, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013. 3, 4, 5

[21] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In *Readings in computer vision*. IEEE, 1984. 4

[22] Soumya Ghosh and Erik B Sudderth. Nonparametric learning for layered segmentation of natural images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2272–2279. IEEE, 2012. 1

[23] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a scene into geometric and semantically consistent regions. In *2009 IEEE 12th international conference on computer vision*, pages 1–8. IEEE, 2009. 8

[24] Xuming He, Richard S Zemel, and Debajyoti Ray. Learning and incorporating top-down cues in image segmentation. In *European conference on computer vision*, pages 338–351. Springer, 2006. 1

[25] Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 654–661. IEEE, 2005. 1

[26] Derek Hoiem, Alexei A Efros, and Martial Hebert. Recovering occlusion boundaries from an image. *International Journal of Computer Vision*, 91(3):328–346, 2011. 2

[27] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, 38(4):814–830, 2016. 1

[28] Alex Levinshtein, Adrian Stere, Kiriakos N Kutulakos, David J Fleet, Sven J Dickinson, and Kaleem Siddiqi. Turbopixels: Fast superpixels using geometric flows. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2290–2297, 2009. 2

[29] Li Li, Jian Yao, Jinge Tu, Xiaohu Lu, Kai Li, and Yahui Liu. Edge-based split-and-merge superpixel segmentation. In *2015 IEEE International Conference on Information and Automation*, pages 970–975. IEEE, 2015. 3

[30] Zhenguo Li, Xiao-Ming Wu, and Shih-Fu Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 789–796. IEEE, 2012. 1

[31] Imanol Luengo, Mark Basham, and Andrew P French. Smurfs: superpixels from multi-scale refinement of superregions. 2016. 3

[32] Alastair P Moore, Simon JD Prince, Jonathan Warrell, Umar Mohammed, and Graham Jones. Superpixel lattices. 2008. 2

[33] Shaul Oron, Aharon Bar-Hillel, Dan Levi, and Shai Avidan. Locally orderless tracking. *International Journal of Computer Vision*, 111(2):213–228, 2015. 1

[34] Caroline Pantofaru, Cordelia Schmid, and Martial Hebert. Object recognition by integrating multiple image segmentations. In *European conference on computer vision*, pages 481–494. Springer, 2008. 1

[35] Tanu Priya, Saurabh Prasad, and Hao Wu. Superpixels for spatially reinforced bayesian classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 12(5):1071–1075, 2015. 1

[36] Carl Yuheng Ren and Ian Reid. gslic: a real-time implementation of slic superpixel segmentation. *University of Oxford, Department of Engineering, Technical Report*, 2011. 2, 8

[37] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *null*, page 10. IEEE, 2003. 1

[38] Zhile Ren and Gregory Shakhnarovich. Image segmentation by cascaded region agglomeration. In *CVPR*, pages 2011–2018, 2013. 2

[39] James Scanlon and Narsingh Deo. Graph-theoretic algorithms for image segmentation. In *ISCAS'99. Proceedings of the 1999 IEEE International Symposium on Circuits and Systems VLSI (Cat. No. 99CH36349)*, volume 6, pages 141–144. IEEE, 1999. 2

[40] Jianbing Shen, Xiaopeng Hao, Zhiyuan Liang, Yu Liu, Wenguan Wang, and Ling Shao. Real-time superpixel segmentation by dbscan clustering algorithm. *IEEE Transactions on Image Processing*, 25(12):5933–5942, 2016. 2

[41] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000. 2

[42] Samuel David Silvey. *Statistical inference*. Routledge, 1970. 3

[43] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixel segmentation using depth information. *RWTH Aachen University, Aachen, Germany*, 4, 2014. 2, 6, 7

[44] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018. 2, 7, 8

[45] Erik Blaine Sudderth. *Graphical models for visual object recognition and tracking*. PhD thesis, Massachusetts Institute of Technology, 2006. 1

[46] Joseph Tighe and Svetlana Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *European conference on computer vision*, pages 352–365. Springer, 2010. 1

[47] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *European conference on computer vision*, pages 13–26. Springer, 2012. 2

[48] Olga Veksler, Yuri Boykov, and Paria Mehrani. Superpixels and supervoxels in an energy optimization framework. In *European conference on Computer vision*, pages 211–224. Springer, 2010. 2

[49] Murong Wang, Xiabi Liu, Yixuan Gao, Xiao Ma, and Nouman Q Soomro. Superpixel segmentation: a benchmark. *Signal Processing: Image Communication*, 56:28–39, 2017. 2

[50] Gerhard Winkler. *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*, volume 27. Springer Science & Business Media, 2012. 4

[51] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[52] Jian Yao, Marko Boben, Sanja Fidler, and Raquel Urtasun. Real-time coarse-to-fine topologically preserving segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2947–2955, 2015. 2, 6, 7

[53] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 8

[54] Xuewen Zhang, Selene E Chew, Zhenlin Xu, and Nathan D Cahill. Slic superpixels for efficient graph-based dimensionality reduction of hyperspectral imagery. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI*, volume 9472, page 947209. International Society for Optics and Photonics, 2015. 1