

# Joint Color-Spatial-Directional Clustering and Region Merging (JCSD-RM) for Unsupervised RGB-D Image Segmentation

Md. Abul Hasnat, Olivier Alata, and Alain Trémeau

**Abstract**—Recent advances in depth imaging sensors provide easy access to the synchronized depth with color, called RGB-D image. In this paper, we propose an unsupervised method for indoor RGB-D image segmentation and analysis. We consider a statistical image generation model based on the color and geometry of the scene. Our method consists of a joint color-spatial-directional clustering method followed by a statistical planar region merging method. We evaluate our method on the NYU depth database and compare it with existing unsupervised RGB-D segmentation methods. Results show that, it is comparable with the state of the art methods and it needs less computation time. Moreover, it opens interesting perspectives to fuse color and geometry in an unsupervised manner.

**Index Terms**—Unsupervised, clustering, RGB-D image segmentation, directional distributions, Bregman divergence, mixture model, region adjacency graph, region merging

## 1 INTRODUCTION

IN the field of image processing, *segmentation* is one of the oldest and most widely studied problems that groups perceptually similar pixels based on certain features, e.g., color, texture, etc. A variety of different techniques already exist in literature [1], which address image segmentation from different perspectives. In this paper, we address the problem of synchronized color and depth image segmentation and propose a solution that combines a clustering method [2] with a statistical region merging technique [3].

After the introduction of Microsoft Kinect camera in 2010, the availability of RGB-D images is widespread now [4], [5]. As a consequence, traditional computer vision algorithms which had been previously developed for color/intensity image, have been enhanced to incorporate depth information [5]. Recent progresses on RGB-D image segmentation [6], [7], [8], [9], [10], [11] have shown that depth as an additional feature improves accuracy of this task. Most of the techniques address the problem with supervised approaches (e.g., [6]). In contrary, unsupervised approach (e.g., [10]) remains underexplored. Moreover, it remains an important issue - what is the best way to fuse color and geometry in an unsupervised manner? This motivates us to conduct further research and contribute towards unsupervised indoor RGB-D image segmentation.

This paper proposes a scene segmentation approach which first identifies the possible image regions w.r.t. a statistical image generation model. Then it merges regions using the planar statistics and the RGB-D image gradients.

The proposed model is based on three different cues/features of the RGB-D image: color, 3D location and surface normals. It follows a generative model-based approach for these features in which they are issued independently (*naïve Bayes* [12], [13] assumption) from a finite mixture of certain probability distributions. The model considers the Gaussian distribution [2] for color and 3D features and the directional (Fisher or Watson) distribution [14], [15], [16] for surface normals. We use the directional distribution because: (a) it provides adequate statistics to explain the planar property of regions and (b) it helps us to develop a simple and effective region merging method. A common property of the Gaussian and directional (Fisher or Watson) distributions is that they belong to the Regular Exponential Family (REF) [14], [17], [18]. We exploit this property to develop an efficient clustering method based on the proposed image generation model.

Finite Mixture Models are commonly used for cluster analysis [19], [20], [21], [22]. In the context of image analysis and segmentation these models have been employed with the Gaussian distribution for clustering the color image [1], [23], [24], [25], [26]. Recently, these models have been exploited to analyze 3D images by clustering the surface normals with a mixture of directional distributions [15], [16]. These clusters are obtained by using the Expectation Maximization (EM) algorithm that performs Maximum Likelihood Estimate (MLE) of the model parameters [2], [27], [28]. This paper proposes an EM method for a combined mixture model of multiple probability distributions, where each distribution belongs to the REF. Precisely, we propose an EM method for joint clustering of independent features.

Bregman Soft Clustering (BSC) is a centroid based parametric clustering method [29]. It has been effectively employed to estimate parameters of mixture models which are based on the REF [18]. Compare to the traditional EM based algorithm, BSC provides additional benefits: (a)

• The authors are with the Laboratoire Hubert Curien, Université Jean Monnet, Saint Etienne, France.

E-mail: {mdabul.hasnat, olivier.alata, alain.tremeau}@univ-st-etienne.fr.

Manuscript received 28 Aug. 2015; revised 8 Dec. 2015; accepted 21 Dec. 2015. Date of publication 29 Dec. 2015; date of current version 10 Oct. 2016.

Recommended for acceptance by R. Yang.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TPAMI.2015.2513407

simplifies the computationally expensive M-step of traditional EM method; (b) applicable to mixed data types and (c) computational complexity is linear in the data points. This paper extends the BSC algorithm to perform efficient clustering

w.r.t. the proposed image model.

Image segmentation based on region merging is one of the oldest techniques used in computer vision [2]. Numerous methods which merge regions in a RGB image exploit color and edge information [3], [30], [31], [32]. For indoor scenes, the use of color is often unreliable due to numerous effects caused by spatially varying illumination [6] and the presence of shadows. Therefore, for indoor scenes, color based merging is not as effective as it is for outdoor scenes. On the other hand, for indoor scenes the planar surfaces are considered as important geometric primitives. They are often employed for scene decomposition [6], [8], [33], and grouping coplanar segments into extended regions [34]. This motivates us to develop a region merging algorithm by mainly exploiting planar property of regions instead of color. Recent work [16] has shown that the concentration parameter ( $\kappa$ ) of the directional distributions can be exploited for characterizing planar surfaces. We take this into account and efficiently exploit the concentration ( $\kappa$ ) of the surface normals in order to accept or reject a merging operation.

This paper proposes a novel unsupervised RGB-D segmentation method. It begins by applying a joint clustering method on RGB-D image features (color, position and normals), which generates a set of regions. Next, it applies a statistical region merging method on resulting regions to obtain the final segmentation. We evaluate the proposed method using RGB-D images of the NYU depth database (NYUD2) [8] and compare our results with the state of the art unsupervised techniques. To benchmark the segmentation task, we consider commonly used evaluation metrics such as [35], [36]: segmentation covering, probability rand index (*PR*), variation of information (*VoI*), boundary displacement error (*BDE*) and boundary based F-measure (*BFM*). Moreover, we also consider the computation time of comparable methods as a measure of evaluation.

Finally, the contributions related to the work described in this paper can be highlighted as follows:

- A statistical RGB-D image generation model (Section 3.1.1) that incorporates both color and geometric properties of the scene.
- An efficient probabilistic joint clustering method (Section 3.2) which exploits the Bregman divergence (BD) [17], [29], [37]. The method has the following properties: (a) performs clustering with respect to the proposed image model; (b) provides an intrinsic view of the indoor scene and (c) provides statistics w.r.t. the planar property of the regions.
- A statistical region merging method (Section 3.3) which satisfies certain region merging predicates. It can be incorporated independently with any other existing indoor RGB-D scene segmentation method.
- A benchmark (Section 4.1) on the NYUD2 [8] for unsupervised scene segmentation. Results from our method show that it is comparable w.r.t. the state of the art and better in terms of computation time.

This work exploits our earlier work on image analysis using directional distribution [15], [16], [38]. Moreover, it provides an extension of our recent work on RGB-D segmentation [39] by including additional details and newer contributions, such as:

- We introduce<sup>1</sup> a general framework which exploits: (a) two theoretical models based on directional statistics in 3D (Fisher and Watson distributions) and (b) information geometry (Bregman divergence).
- We propose a general methodology that can be used with unambiguous (using Fisher distribution) or ambiguous (using Watson distribution) direction.
- For the study of indoor scenes, the ambiguity<sup>2</sup> in surface normal is removed, which allows to use Fisher distribution. New experimental results are discussed through a common framework based on both Fisher and Watson distributions.
- Several additional image models are explored, discussed and corresponding results are provided.
- An enhanced discussion based on new experiments, additional illustrations and clarifications.

The outline of the rest of this paper is as follows: Section 2 discusses the background of RGB-D segmentation methods and related works. Section 3 presents the proposed method. Section 4 provides experimental results and discussion. Finally, Section 5 draws conclusions and discusses future perspectives.

## 2 BACKGROUND OF RGB-D SEGMENTATION

Color image segmentation of natural and outdoor scene is a well-studied problem due to its numerous applications in computer vision. Different methods have been already proposed in the state of the art based on different perspectives. [1, Chapter 5] provides a detail overview of them.

Many of the established image analysis methods have been either extended or directly employed to depth image in order to deal with depth features, see [40, Chapter 6] for details. In the simplest cases, the depth image is considered as a grayscale image or converted to a cloud of 3D points. However, such approaches have limitations [40]. For example, clustering using only 3D points often fails to locate the intersections among planar surfaces with different orientations such as wall, floor, ceiling, etc. This is due to the fact that the 3D points associated to the intersections are grouped into a single cluster. Therefore, better features such as surface normals are suggested to use [33], [41]. However, from a recent study [16], we observe that: (a) the use of surface normals solely is not sufficient to extract full semantics of the scene, e.g., multiple objects with nearly similar orientations may grouped into the same cluster irrespective of their 3D location and (b) it is necessary to incorporate additional features, such as color, texture, etc. to provide better interpretation of indoor environments. Such observations

1. In order to cluster surface normal, we proposed two methods: one based on the Fisher distribution in [15] and another based on the Watson distribution in [16]. In this paper, we exploit both of them within a common framework.

2. In our previous work [39], we used the toolbox of [8] which produced ambiguity [33] in the direction of the normals. In this paper, we decided to use the toolbox of [9] which removes such ambiguity.

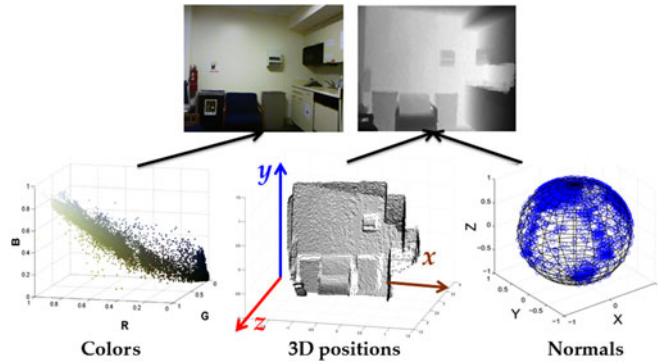


Fig. 1. Illustration of the proposed image generation model. The first row shows the color and depth image. The second row shows the model features in their respective spaces.

raise the necessity to jointly exploit depth, color and other features for the task of RGB-D image analysis.

A number of recent research activities, such as [10], [6], [9] and [8], proposed different methodologies for indoor scene understanding and analysis with promising results. Most of these researches incorporate depth as complementary information with color images. They can be categorized mainly from two aspects: (a) *feature-wise*: different types, levels and dimensions of features and (b) *method-wise*: numerous distinctions, such as supervised, unsupervised, clustering based, graph based, split-merge based, etc. Different methods emphasize on different aspects of the problem, which in general opens a number of interesting and challenging issues to focus on.

A common approach to tackle the RGB-D scene analysis problem is to extract different features, design kernels and classify pixels with learned classifiers. For example, [9] proposed contextual models in a supervised setting. Their model combines kernel descriptors with a segmentation tree or with superpixels Markov Random Field (MRF). To this aim, they extended the well-known gPb-UCM algorithm [35] to incorporate the global probability of boundaries (gPb) of depth image with gPb of RGB image. The RGB-D scene analysis method proposed by [8] first gives an over-segmentation of the scene by applying watershed on the gPb of the RGB image. Next, it aligns the over-segmentation with the 3D planes. Finally, using a trained classifier it applies a hierarchical segmentation in order to merge regions. Another interesting feature of [8] is that it provides an annotated RGB-D dataset (NYUD2) to perform scene analysis. Recently, [6] extended the gPb-UCM [35] method to a supervised setting. First, they combine geometric contour cues: convex and concave normal gradients with monocular cues: brightness, color, texture. Then, they detect pixels as contours via learned classifiers for eight different orientations. Finally, they generate a hierarchy of segmentations from all oriented detectors. All of the above-mentioned methods follow supervised approach. Let us now focus on methods developed for the unsupervised domain.

Dal Mutto et al. [10] discussed about the fusion of color with geometry based on an unsupervised setting and provide a solution using the normalized cut spectral clustering method. Their approach identifies an optimal multiplier to balance between color and depth. For this reason, they generate several segmentations with different values of the

multiplier. Each segmentation is obtained by applying spectral clustering on the fused subsampled features. Finally, they select the best segmentation based on their proposed RGB-D segmentation quality evaluation score. In practice, this method requires more computation time than others as it generates a number of different segmentations for a single image. A different method is proposed in [34] and [7], which first extracts edges from RGB image, applies Delaunay Triangulation on edges to construct triangular graph and then generates initial segmentation by applying Normalized Cut (in [34]) or agglomerative merging (in [7]). Next, it extracts planar surfaces from the segments using RANSAC [1] and finally merges the coplanar segments using a greedy merging procedure. The unsupervised method that we propose in this paper is different than the above proposals as: (a) it considers surface normals as features and (b) it employs mixture model based joint clustering rather than Normalized Cut and (c) it exploits directional statistics for region merging.

Beside these approaches, the well-known graph based segmentation [42] is extended for joint color and depth image segmentation. For example, [43] extended it by including disparity with color for the purpose of segmenting stereopsis images. [44] extended it by incorporating surface normals to segment colored 3D laser point clouds. For the purpose of comparison, we develop an extension of this method that considers color, 3D and normals.

Despite all of these researches, it remains an interesting issue about how to build an appropriate statistical model to describe RGB-D images of indoor scenes and how to exploit such model to segment the captured images. Scene-SIRFS [45] is a recently proposed model whose aim is to recover intrinsic scene properties from single RGB-D image. It considers a mixture of shapes and illuminations where the mixture components are embedded in a soft segmentation of 17 eigenvectors. These eigenvectors are obtained from the normalized Laplacian corresponding to the input RGB image. Although the concept of using mixture is similar to the method proposed in this paper, the underlying objective, model and methodologies are different. We consider a mixture of shape (via 3D and normals) and color that consists of a feature vector of length 9. In the next Section, we present our proposed scene analysis method.

### 3 METHODOLOGY

In this section, we present the proposed RGB-D segmentation method. First, we discuss the statistical image generation model and present the segmentation method w.r.t. the model. Then we briefly present the joint clustering method followed by the region merging method.

#### 3.1 Model and Method

##### 3.1.1 Image Generation Model

We propose a statistical image model that fuses color and shape (3D and surface normals) features according to the *naïve Bayes* assumption [12], [13], i.e., the features are independent of each other. Furthermore, it is based on a generative model-based approach [2], where the features are issued from a finite mixture of different probability distributions. Fig. 1 provides an illustration of the proposed image generation model. We can observe that, the color and 3D



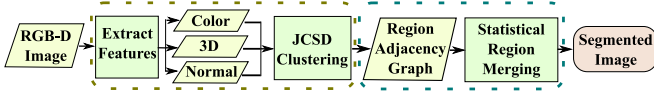


Fig. 2. Block diagram of the proposed segmentation method.

features belong to the standard euclidean space, i.e., in  $\mathbb{R}^3$  and the surface normal<sup>3</sup> belongs to the unit sphere, i.e., in  $S^2$ . Based on this observation, we consider the multivariate Gaussian [28] distribution for the color and 3D features and the directional<sup>4</sup> (Fisher or Watson) [14], [15], [16] distribution for surface normals. Mathematically, such a model with  $k$  components has the following form:

$$g(\mathbf{x}_i | \Theta_k) = \sum_{j=1}^k \pi_{j,k} f_g(\mathbf{x}_i^C | \mu_{j,k}^C, \Sigma_{j,k}^C) f_g(\mathbf{x}_i^P | \mu_{j,k}^P, \Sigma_{j,k}^P) f_{dir}(\mathbf{x}_i^N | \mu_{j,k}^N, \kappa_{j,k}^N). \quad (1)$$

Here  $\mathbf{x}_i = \{\mathbf{x}_i^C, \mathbf{x}_i^P, \mathbf{x}_i^N\}$  is the nine dimensional feature vector of the  $i$ th pixel with  $i = 1, \dots, M$ . Superscripts denote: C - color, P - 3D position and N - normal.  $\Theta_k = \{\pi_{j,k}, \mu_{j,k}^C, \Sigma_{j,k}^C, \mu_{j,k}^P, \Sigma_{j,k}^P, \mu_{j,k}^N, \kappa_{j,k}^N\}_{j=1 \dots k}$  denotes the set of model parameters where  $\pi_{j,k}$  is the prior probability,  $\mu_{j,k} = \{\mu_{j,k}^C, \mu_{j,k}^P, \mu_{j,k}^N\}$  is the mean,  $\Sigma_{j,k} = \{\Sigma_{j,k}^C, \Sigma_{j,k}^P\}$  is the variance-covariance symmetric positive-definite matrix and  $\kappa_{j,k}^N$  is the concentration of the  $j$ th component.  $f_g(\cdot)$  and  $f_{dir}(\cdot)$  are the density functions of the multivariate Gaussian distribution (Section 3.2.2) and the directional (Fisher or Watson) distribution (Sections 3.2.3 and 3.2.4) respectively.

### 3.1.2 Segmentation Method

Figs. 2 and 3 illustrate the block diagram and an example of the proposed RGB-D segmentation method. It consists of two sub-tasks: (1) clustering heterogeneous (color, 3D and Normal) data and (2) merging regions. The first task performs a joint color-spatial-directional (JCSO) clustering and generates a set of regions. The second task performs a refinement on this set with the aim to merge regions which are susceptible to be over-segmented. In the next two sub-sections we present our methods to accomplish these tasks.

## 3.2 Joint Color-Spatial-Directional Clustering

In order to cluster heterogeneous data, we develop a joint color-spatial-directional clustering method. The clustering method estimates the parameters of the mixture model (Eq. (1)) as well as clusters the image data/features. As an outcome, we obtain the groups of image pixels which form the regions in the image. However, notice that in an unsupervised setting the true number of segments are unknown. Therefore, we cluster features with the assumption of a

3. Surface normal is a 3D unit vector that describes the planar property of a pixel. This planar property is the perpendicular direction to the plane which is fitted on each pixel using chosen neighboring pixels. In the unit sphere of Fig. 1, each blue point indicates the direction of a pixel's normal w.r.t. the origin of the sphere.

4. We use the term *directional* for the *Fisher* and the *Watson* distribution. Both of them are parameterized with a mean direction  $\mu$  and concentration value  $\kappa$ . They belong to the regular exponential family, which allows us to provide a common formulation despite their different normalization function, see Section 3.2.3 and 3.2.4.

known maximum number of clusters ( $k = k_{max}$ ). Section 4.2 provides additional details on this issue. Such assumption often causes an over-segmentation of the image. In order to overcome this issue, it is necessary to merge the over-segmented regions (see Section 3.3).

The proposed joint clustering method follows the Bregman Soft Clustering method [29] and extends it to combine multiple probability distributions which belong to the REF. The extension is based on the independence assumption to combine different distributions for different features. This allows computing the divergence among two distributions based on the following combined form:

$$f_{\cup}(\mathbf{x}_i | \Theta_{j,k}) = f_g(\mathbf{x}_i^C | \mu_{j,k}^C, \Sigma_{j,k}^C) f_g(\mathbf{x}_i^P | \mu_{j,k}^P, \Sigma_{j,k}^P) f_{dir}(\mathbf{x}_i^N | \mu_{j,k}^N, \kappa_{j,k}^N), \quad (2)$$

where  $\Theta_{j,k} = \{\pi_{j,k}, \mu_{j,k}^C, \Sigma_{j,k}^C, \mu_{j,k}^P, \Sigma_{j,k}^P, \mu_{j,k}^N, \kappa_{j,k}^N\}$  denotes the  $j$ th component of parameter  $\Theta_k$ . This allows to develop a joint Bregman soft clustering method for the model in Eq. (1).

### 3.2.1 Regular Exponential Family of Distributions and Bregman Divergence

A multivariate probability density function  $f(x|\eta)$  belongs to the Regular Exponential Family [17] if it has the following (see [29, Eq. (3.7)] [18, Eq. (60)] form<sup>5</sup>:

$$f(x|\eta) = \exp(-D_G(t(x), \eta)) \exp(k(x)) \quad (3)$$

and

$$D_G(\eta_1, \eta_2) = G(\eta_1) - G(\eta_2) - \langle \eta_1 - \eta_2, \nabla G(\eta_2) \rangle \quad (4)$$

with  $G(\cdot)$  is the Legendre dual of  $F(\cdot)$ .  $F(\cdot)$  is a strictly convex log normalizing function associated with a probability distribution.  $\nabla G$  is the gradient of  $G$ .  $t(x)$  denotes the sufficient statistics and  $k(x)$  is the carrier measure. The expectation of the sufficient statistics  $t(x)$  w.r.t. the density function (Eq. (3)) is called the expectation parameter ( $\eta$ ).  $D_G$  is the Bregman Divergence [17], [29], [37] computed from expectation parameters, see Appendix A. BD can be used as a measure of dissimilarity between two distributions of the same exponential family which are defined by two expectation parameters  $\eta_1$  and  $\eta_2$ . We will define in the following Section the particular forms obtained with the Gaussian distribution and the directional (Fisher and Watson) distribution.

### 3.2.2 Multivariate Gaussian Distribution

For a  $d$  dimensional random vector  $\mathbf{x} = [x_1, \dots, x_d]^T \in \mathbb{R}^d$ , the multivariate Gaussian distribution is defined as:

$$f_g(\mathbf{x} | \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} \det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right). \quad (5)$$

Here,  $\mu \in \mathbb{R}^d$  denotes the mean and  $\Sigma$  denotes the variance-covariance symmetric positive-definite matrix. To write the

5. In order to keep our formulations concise, we use the expectation parameters  $\eta$  to define the REF distributions. However, the other form (see Appendix A) and related derivations are available in [25] (for the Gaussian distribution) and [15], [16] (for the Fisher and Watson distributions).

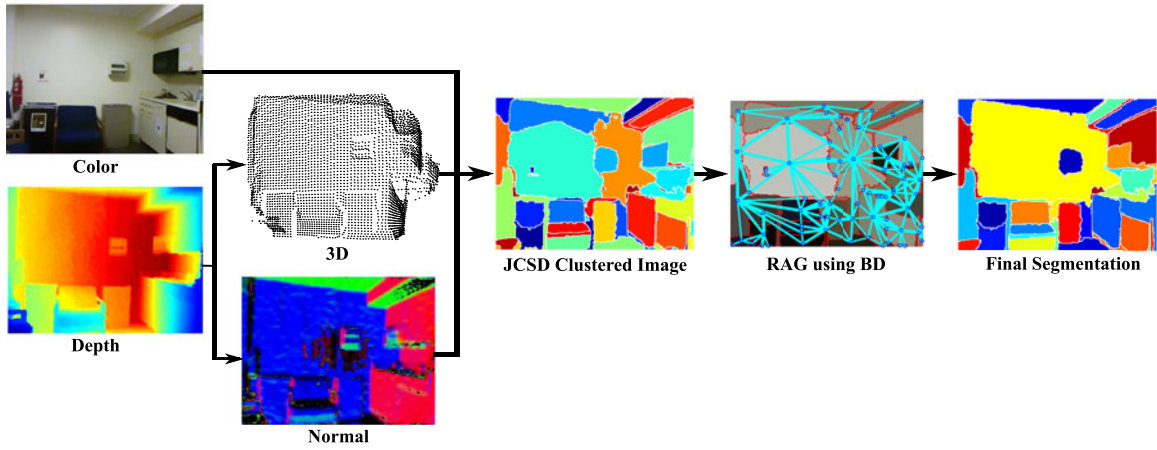


Fig. 3. Illustration with an example of the proposed RGB-D segmentation method.

multivariate Gaussian distribution in the form of Eq. (3), the elements are defined as [18]: sufficient statistics  $t(\mathbf{x}) = (\mathbf{x}, -\mathbf{x}\mathbf{x}^T)$ ; carrier measure  $k(\mathbf{x}) = 0$ ; expectation parameter  $\eta = (\phi, \Phi) = (\mu, -(\Sigma + \mu\mu^T))$  and

$$G_g(\eta) = -\frac{1}{2} \log(1 + \phi^T \Phi^{-1} \phi) - \frac{1}{2} \log(\det(\Phi)) - \frac{d}{2} \log(2\pi e). \quad (6)$$

### 3.2.3 Fisher Distribution

For a three-dimensional random unit vector  $\mathbf{x} = [x_1, x_2, x_3]^T \in S^2 \subset \mathbb{R}^3$  (i.e.,  $\|\mathbf{x}\|_2 = 1$ ), the Fisher distribution is defined as [14], [15]:

$$f_{dir}(x|\mu, \kappa) = \frac{\kappa}{\sinh(\kappa)} \exp(\kappa \mu^T x). \quad (7)$$

Here,  $\mu$  denotes the mean (with  $\|\mu\|_2 = 1$ ) and  $\kappa$  denotes the concentration parameter (with  $\kappa \geq 0$ ). The Fisher distribution is a special case of the von Mises-Fisher (vMF) [14] distribution for three dimensional observations. To write the Fisher distribution in the form of Eq. (3), the elements are defined as [15], [38]: sufficient statistics  $t(x) = x$ ; carrier measure  $k(x) = 0$ ; expectation parameter  $\eta = \|\eta\|_2 \mu$  and

$$G_{dir}(\eta) = \kappa \|\eta\|_2 - \log\left(\frac{\kappa}{\sinh(\kappa)}\right). \quad (8)$$

With the above formulation, for a set of observations  $\mathbf{X} = \{\mathbf{x}_i\}_{i=1, \dots, M}$  we estimate  $\eta = E[t(\mathbf{X})]$  and  $\kappa$  with a Newton-Raphson root finder method as [15], [38]:

$$\kappa_{l+1} = \kappa_l - \frac{a - b - \|\eta\|_2}{1 - a^2 + b^2}, \quad (9)$$

where,  $a = \tanh(\kappa)^{-1}$  and  $b = (\kappa)^{-1}$ .

### 3.2.4 Multivariate Watson Distribution

For a  $d$  dimensional unit vector  $\mathbf{x} = [x_1, \dots, x_d]^T \in S^{d-1} \subset \mathbb{R}^d$  (i.e.,  $\|\mathbf{x}\|_2 = 1$ ), the multivariate (axially symmetric, i.e.,  $f_{dir}(\mathbf{x}|\mu, \kappa) = f_{dir}(-\mathbf{x}|\mu, \kappa)$ ) Watson distribution (mWD) is defined as [14]:

$$f_{dir}(\mathbf{x}|\mu, \kappa) = M(1/2, d/2, \kappa)^{-1} \exp(\kappa(\mu^T \mathbf{x})^2). \quad (10)$$

Here,  $\mu$  is the mean direction (with  $\|\mu\|_2 = 1$ ),  $\kappa \in \mathbb{R}$  the concentration and  $M(1/2, d/2, \kappa)$  the Kummer's function [14]. To write the mWD in the form of Eq. (3), the elements are defined as [16]: sufficient statistics  $t(\mathbf{x}) = [x_1^2, \dots, x_d^2, \sqrt{2}x_1x_2, \dots, \sqrt{2}x_{d-1}x_d]^T$ ; carrier measure  $k(\mathbf{x}) = 0$ ; expectation parameter  $\eta$ :

$$\eta = \|\eta\|_2 \nu, \quad (11)$$

where  $\nu = [\mu_1^2, \dots, \mu_d^2, \sqrt{2}\mu_1\mu_2, \dots, \sqrt{2}\mu_{d-1}\mu_d]^T$  and

$$G_{dir}(\eta) = \kappa \|\eta\|_2 - \log M(1/2, d/2, \kappa). \quad (12)$$

With the above formulation, for a set of observations  $\mathbf{X} = \{\mathbf{x}_i\}_{i=1, \dots, M}$  we estimate  $\eta = E[t(\mathbf{X})]$  and  $\kappa$  with a Newton-Raphson root finder method as [16]:

$$\kappa_{l+1} = \kappa_l - \frac{q(1/2, d/2; \kappa_l) - \|\eta\|_2}{q'(1/2, d/2; \kappa_l)}, \quad (13)$$

where  $q(1/2, d/2; \cdot)$  is the Kummer-ratio,  $q'(1/2, d/2; \cdot)$  is the derivative of  $q(1/2, d/2; \cdot)$ .

### 3.2.5 Bregman Divergence for the Combined Model

Our image model (in Eq. (1)) combines different exponential family of distributions (associated to color, 3D and normals) based on independent (*naïve Bayes* [12], [13]) assumption. Therefore, Bregman Divergence [17], [29], [37] of the combined model can be defined as a linear combination of the BD of each individual distributions:

$$D_G^U(\eta_i, \eta_j) = D_{G,g}^C(\eta_i^C, \eta_j^C) + D_{G,g}^P(\eta_i^P, \eta_j^P) + D_{G,dir}^N(\eta_i^N, \eta_j^N), \quad (14)$$

where,  $D_{G,g}(\cdot, \cdot)$  denotes BD using the multivariate Gaussian distribution [25] and  $D_{G,dir}(\cdot, \cdot)$  denotes BD using the directional (Fisher or Watson) distribution [15]. Then, it is possible to define, with expectation parameter  $\eta = \{\eta^C, \eta^P, \eta^N\}$ :

$$G^U(\eta) = G_g(\eta^C) + G_g(\eta^P) + G_{dir}(\eta^N). \quad (15)$$

### 3.2.6 Bregman Soft Clustering for the Combined Model

Bregman Soft Clustering exploits Bregman Divergence in the Expectation Maximization [27] framework to compute the

Maximum Likelihood Estimate of the mixture model parameters and provides a soft clustering of the observations [29].

In order to cluster data with the combined model (Eq. (1)), it is necessary to estimate the model parameters and obtain  $\hat{\Theta}_k$  for  $g(\mathbf{X}|\Theta_k)$  such that:

$$\hat{\Theta}_k = \arg \max_{\Theta_k} g(\mathbf{X}|\Theta_k) \text{ with } g(\mathbf{X}|\Theta_k) = \prod_{i=1}^M g(\mathbf{x}_i|\Theta_k). \quad (16)$$

Here,  $\mathbf{X} = \{\mathbf{x}_i\}_{i=1,\dots,M}$  is the set of observations. Let  $\gamma_i = j$  denotes the class label of an observation  $\mathbf{x}_i$  with  $j = \{1, \dots, k\}$ .

BSC consists of an Expectation step (E-step) and a Maximization step (M-step). In the E-step of the algorithm, the posterior probability is computed as [18]:

$$p(\gamma_i = j|\mathbf{x}_i) = \frac{\pi_{j,k} \exp(G^\cup(\eta_{j,k}) + \langle t(\mathbf{x}_i) - \eta_{j,k}, \nabla G^\cup(\eta_{j,k}) \rangle)}{\sum_{l=1}^k \pi_{l,k} \exp(G^\cup(\eta_{l,k}) + \langle t(\mathbf{x}_i) - \eta_{l,k}, \nabla G^\cup(\eta_{l,k}) \rangle)}. \quad (17)$$

Here,  $\eta_{j,k}$  and  $\eta_{l,k}$  denote the expectation parameters for any cluster  $j$  and  $l$  given that the total number of components is  $k$ . The M-step updates the mixing proportion and expectation parameter for each class as:

$$\pi_{j,k} = \frac{1}{M} \sum_{i=1}^M p(\gamma_i = j|\mathbf{x}_i) \text{ and } \eta_{j,k} = \frac{\sum_{i=1}^M p(\gamma_i = j|\mathbf{x}_i) \mathbf{x}_i}{\sum_{i=1}^M p(\gamma_i = j|\mathbf{x}_i)}. \quad (18)$$

Initialization (using the EM method) is a challenging issue and has significant impact on clustering [46]. Our initialization procedure consists of setting initial values for prior class probability ( $\pi_{j,k}$ ) and the expectation parameters ( $\eta_{j,k}$ ) with  $1 \leq j \leq k$ . We obtain these initial values for the Gaussian and directional (Fisher or Watson) distributions using a combined k-means type clustering. After initialization, we iteratively apply the E-step and M-step until the convergence criteria are met. These criteria are based on maximum number of iterations (e.g., 200) and a threshold difference (e.g., 0.001) between the negative log likelihood values (see Eq. (19) and Eq. (1)) of two consecutive steps.

$$nLLH(\hat{\Theta}_k) = - \sum_{i=1}^M \log(g(\mathbf{x}_i|\hat{\Theta}_k)). \quad (19)$$

The above procedures lead to a soft clustering, which generates associated probabilities and parameters for each component of the proposed model defined by Eq. (1). Finally, for each sample we get the cluster label ( $\hat{\gamma}_i$ ) using the updated combined BD (Eq. 14) as:

$$\hat{\gamma}_i = \arg \min_{j=1,\dots,k} D_G^\cup(t(\mathbf{x}_i), \hat{\eta}_{j,k}). \quad (20)$$

Applying Eq. (20) performs hard clustering on the data. Let us call this entire clustering method the joint color-spatial-directional algorithm (Algorithm 1). It can be seen as a general method to combine different types of REF probability distributions with an independent assumption.

Applying Algorithm 1 on RGB-D image features (color, position and normals) performs a joint color-spatial-

directional clustering. This clustering method is based on the assumption of a known maximum number of components  $k = k_{max}$ . Image regions obtained by such clustering often lead to over-segmentation, see Fig. 3 for example. Therefore, it is necessary to merge the over-segmented regions. In the following section, we propose a region merging method to overcome such over-segmentation problem.

---

#### Algorithm 1. JCSD Clustering Algorithm for Joint Color-Spatial-Directional Clustering

---

**Input:**  $\mathbf{X} = \{\mathbf{x}_i \mid \mathbf{x}_i = \{\mathbf{x}_i^C, \mathbf{x}_i^P, \mathbf{x}_i^N\} \wedge 1 \leq i \leq M\}$

**Output:** Clustering of  $\mathbf{X}$  with  $k$  components.

Initialize  $\pi_{j,k}$  and  $\eta_{j,k}$  for  $1 \leq j \leq k$  using combined k-means;

**while** not converged **do**

    {Perform the E-step of EM};

**foreach**  $i$  and  $j$  **do**

        Compute  $p(\gamma_i = j|\mathbf{x}_i)$  using Eq. (17)

**end**

    {Perform the M-step of EM};

**for**  $j = 1$  to  $k$  **do**

        Update  $\pi_{j,k}$  and  $\eta_{j,k}$  using Eq. (18)

**end**

Define final values of parameters as  $\hat{\pi}_{j,k}$  and  $\hat{\eta}_{j,k}$ . Assign each observation to a cluster using Eq. (20)

---

### 3.3 Region Merging

In this step, we merge the regions which are obtained from the JCSD clustering. To this aim, first we build a Region Adjacency Graph (RAG) [30] (see Figs. 2 and 3). This graph is defined such as each region is a node and each node has edges with its adjacent nodes. In order to weight the edges, we consider a dissimilarity measure based on directional distributions and the boundary strength among two regions. Similar to the standard methods [3], [30], [31], our approach depends on the region merging predicate and merging order. As an outcome of region merging we obtain the final segmentation.

#### 3.3.1 Region Adjacency Graph

RAG provides an inherent view of the region merging strategy. From the JCSD clustered labels, we build the RAG by first applying a 3x3 median filter (removes noisy labels) and then locating the regions from the enclosed boundaries. Fig. 4 illustrates an example of the RAG.

Let  $R = \{r_i\}_{i=1,\dots,Z}$  be the set of regions,  $G = (V, E)$  be the RAG, where  $v_i \in V$  is the set of nodes corresponding to  $r_i \in R$  and  $E$  is the set of edges among adjacent nodes.

Each node  $v_i$  is characterized by the source parameters ( $\mu$  and  $\kappa$ ) of the directional distribution (Sections 3.2.3 or 3.2.4). Besides, in order to merge nodes efficiently, we compute the probability ( $\pi$ ) and the expectation parameter ( $\eta$ ) for each node. For a region  $r_i$ ,  $\pi_i$  is computed as the ratio of the number of region pixels w.r.t. total number of image pixels and  $\eta_i^N$  is computed as the mean of the normals.

Each edge  $e_{ij}$  consists of two weights:  $w_d$ , based on statistical dissimilarity and  $w_b$ , based on boundary strength. The dissimilarity based weight  $w_d$  is computed as:

$$w_d(v_i, v_j) = \min(D_{G,dir}^N(\eta_i^N, \eta_j^N), D_{G,dir}^N(\eta_j^N, \eta_i^N)), \quad (21)$$



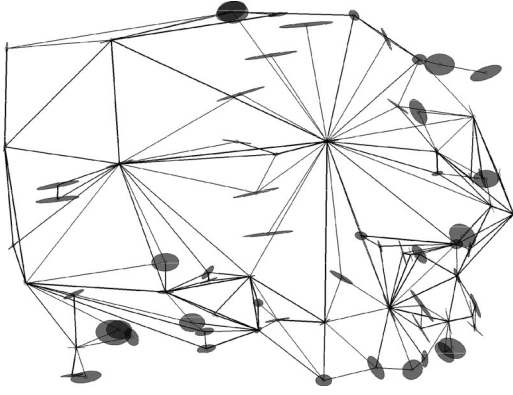


Fig. 4. Illustration of a 3d view of the Region Adjacency Graph constructed from JCSD clustered regions shown in Fig. 3. For each circle (node), its radius represents the concentration ( $\kappa$ ) and its orientation represents the mean direction ( $\mu$ ) of the image normals of the associated region. Each edge represents the weight ( $w_d$  or  $w_b$ ) among two nodes. In this picture several circles resemble ellipses because of 3D to 2D projection. The 2D view of this graph overlaid on the original image is illustrated in Fig. 3.

where,  $D_{G,dir}^N(\eta_i^N, \eta_j^N)$  is the Bregman divergence (Eq. (32)) among the directional (Fisher or Watson) distributions of  $r_i$  and  $r_j$ . The boundary based weight  $w_b$  is computed as:

$$w_b(v_i, v_j) = \frac{1}{|r_i \cap r_j|} \sum_{b \in r_i \cap r_j} I_G^{rgb,d}(b), \quad (22)$$

where,  $r_i \cap r_j$  is the set of boundary pixels among two regions,  $|\cdot|$  denotes the cardinality and  $I_G^{rgb,d}$  is the normalized magnitude of image gradient<sup>6</sup> (MoG) [1] computed from the RGB-D image.  $I_G^{rgb,d}$  is obtained by first computing MoG for each color channels ( $I_G^r, I_G^g, I_G^b$ ) and depth ( $I_G^d$ ) individually, and then taking the maximum of those MoGs at each pixel.

### 3.3.2 Merging Strategy

Our region merging strategy is defined by an iterative procedure which is based on a merging predicate in a predefined order. The predicate consists of: (a) evaluating the *candidacy* of each node; (b) evaluating the *eligibility* of merging adjacent nodes and (c) verifying the *consistency* of the merged nodes. Fig. 5 illustrates examples to understand the predicate. Fig. 6 provides an example of the merging strategy for a particular node. Once two nodes are merged, the information regarding the merged node and its edges are updated instantly. This procedure continues until no valid candidates are left to merge.

*candidacy* of a node defines whether it is a valid candidate to be merged. For each node, first we check its *candidacy*. This helps us to filter out invalid candidates and hence reduces the computation time. For each node, our *candidacy* criterion checks the planar property of the corresponding region. In indoor scenes regions are either planar (e.g., floor, walls, etc.) or non planar (e.g., coffee pot, lamps, etc.). Whatever the method used, we noticed that most of over-segmentation

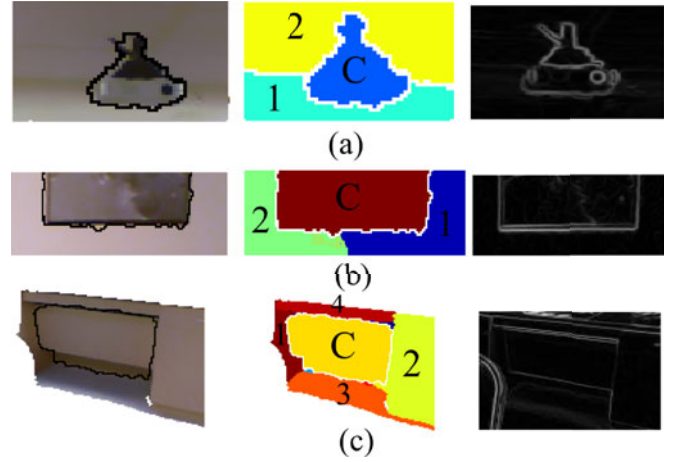


Fig. 5. Illustration of the region merging predicate with different examples. The left column shows the Rol under process (surrounded by a black boundary) in the original image. The middle column shows the Rol (labeled as **C**) and the neighboring regions (labeled with numbers). The last column shows the magnitude of RGB-D image gradient.

errors are related to planar regions rather than non-planar regions (due to shadows, non-uniformity of lightness, etc.). Therefore, we propose to use the planarity assumption as the first criterion for merging, focus on adjacent planar regions and avoid non planar regions. This planarity property can be easily investigated from the concentration parameter ( $\kappa$ ) of each node  $v_i$ . We define the *candidacy* of a node  $v_i$  as follows:

$$candidacy(v_i) = \begin{cases} \text{true}, & \text{if } \kappa_i > \kappa_p, \\ \text{false}, & \text{otherwise.} \end{cases} \quad (23)$$

Here  $\kappa_i$  is the concentration parameter of  $r_i$ .  $\kappa_p$  is the threshold that defines the planar property of a region. From our study on planar statistics, see Appendix B, we observed that the  $\kappa$  value of a region can be exploited to discriminate among planar and non-planar surfaces. See Section 4.1 for details about this threshold value, which is set as  $\kappa_p = 5$ . In Fig. 5a, the region of interest (labeled as **C**) has  $\kappa_C = 3$ , which signifies that it is not a valid candidate for merging. Conversely,  $\kappa_C = 58$  in Fig. 5b and  $\kappa_C = 11$  in Fig. 5c means that those regions are valid candidates.

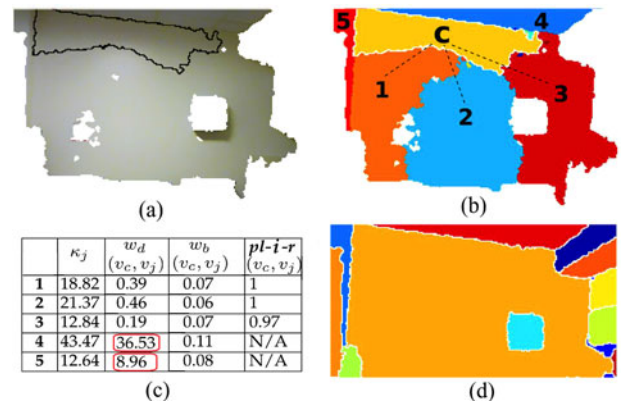


Fig. 6. Illustration of the region merging strategy for a single region/node. (a) shows the Rol under process (surrounded by a black boundary) in the original image. (b) shows the Rol (labeled as **C**) and the neighboring regions (labeled with numbers). (c) provides the computed values (N/A means not necessary to compute) w.r.t. the Rol. (d) shows the regions after merging operation is completed for all Rols.

6. To compute image gradient  $\Delta I = \left( \frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y} \right)$ , with  $\frac{\partial I(x,y)}{\partial x} \approx \frac{I(x+1,y) - I(x-1,y)}{2}$  and  $\frac{\partial I(x,y)}{\partial y} \approx \frac{I(x,y+1) - I(x,y-1)}{2}$ , we used the 'sobel' operator in MATLAB implementation.

We define the *eligibility* to merge two nodes ( $v_i$  and  $v_j$ ) using the boundary based weight  $w_b$  (Eq. 22) and the dissimilarity based weight  $w_d$  (Eq. (21)) as:

$$\text{eligibility}(v_i, v_j) = \begin{cases} \text{true}, & (a) w_b(v_i, v_j) < th_b; \text{ and} \\ & (b) w_d(v_i, v_j) < th_d; \\ \text{false}, & \text{otherwise,} \end{cases} \quad (24)$$

where,  $th_b$  and  $th_d$  are the thresholds for  $w_b$  and  $w_d$ . See Section 4.1 for the details, where we set  $th_d = 3$  and  $th_b = 0.2$ . From experiments, we observed that most pairs of selected (to be merged) regions have very low  $w_d$ . This motivates us to set a heuristic to verify the *eligibility* based on  $w_d$ . The use of the boundary based weight  $w_b$  is motivated from existing techniques such as the OWT-UCM [35].

In order to understand  $w_b$  from example, consider the regions in Fig. 5(b), labeled as C, 1 and 2. They are valid candidates, because  $\kappa_C = 58$ ,  $\kappa_1 = 81$  and  $\kappa_2 = 53$ . Boundary values are:  $w_b(v_C, v_1) = 0.8$ ,  $w_b(v_C, v_2) = 0.7$  and  $w_b(v_1, v_2) = 0.03$ , which signifies that C (region of interest) should not be merged with 1 and 2. Contrarily, the regions 1 and 2 can be merged. Indeed, from visual observation it makes more sense to merge 1 and 2 (wall surfaces), rather than merging C (picture) with others.

Now, to understand  $w_d$  from example, consider two regions of Fig. 5a, labeled as 1 and 2. They are valid candidates ( $\kappa_1 = 65$  and  $\kappa_2 = 67$ ) and eligible ( $w_b(v_1, v_2) = 0.15$ ) to merge. However, the dissimilarity value  $w_d(v_1, v_2) = 7$  is more than the threshold, which means that 1 and 2 do not merge due to the high difference among their surface orientations. This is coherent with visual observation as the back wall (1) and the ceiling (2) should not be merged.

We use the plane inlier ratio *pl-i-r* to verify the *consistency* [31] of a merged region. It is computed by dividing the total number of inliers (3D points fitted within a plane based on a minimum/threshold distance [7]) with the total number of 3D points. We employed the RANSAC [1] algorithm for plane fitting. We define the *consistency* as:

$$\text{consistency}(v_i, v_j) = \begin{cases} \text{true}, & \text{if } pl-i-r(v_i, v_j) > th_r, \\ \text{false}, & \text{otherwise,} \end{cases} \quad (25)$$

where,  $th_r$  is the threshold associated to *pl-i-r*. We set  $th_r = 0.9$  following the existing methods, such as [7].

In order to understand  $th_r$  from example, consider two regions in Fig. 5c, labeled as C and 2. They are valid candidates ( $\kappa_C = 11$  and  $\kappa_2 = 15$ ) and eligible ( $w_b(v_C, v_2) = 0.13$  and  $w_d(v_C, v_2) = 0.8$ ) to merge. However, *pl-i-r*( $v_C, v_2$ ) = 0.84 is less than  $th_r$ , means an inconsistency. This is coherent with visual observation as they belong to two different planes localized at a different distance/depth.

Finally, we define the *region merging predicate* [31]  $P_{ij}$  based on *candidacy* (Eq. (23)), *eligibility* (Eq. (24)) and *consistency* (Eq. (25)) as:

$$P_{ij} = \begin{cases} \text{true}, & \text{if } (a) \text{ candidacy}(v_j) = \text{true}; \text{ and} \\ & (b) \text{ eligibility}(v_i, v_j) = \text{true}; \text{ and} \\ & (c) \text{ consistency}(v_i, v_j) = \text{true} \\ \text{false}, & \text{otherwise.} \end{cases} \quad (26)$$

Fig. 6 illustrates the result after a merging iteration applied to all RoIs. It shows that, based on Eq. (26), several regions are merged (C, 1, 2 and 3) while others remain alone (4, 5). The weight  $w_d$  in condition-(b) is related to the statistical properties of the regions. In the absence of a boundary among two adjacent regions, one may ignore it and expect similar results, because the condition-(c) could also be used to detect the ineligible regions. However, this will significantly increase the computation time because the *eligibility* test (Eq. (24)) is significantly faster than RANSAC.

The *region update*, which we apply after merging, provides updated information of the merged region. We accomplish this by computing  $\pi_m$ ,  $\mu_m$ ,  $\kappa_m$  and  $\eta_m$  of the merged node from  $\pi$  and  $\eta$  of the individual nodes. For a pair of nodes, first we compute the probability ( $\pi_m$ ) and expectation parameter ( $\eta_m$ ) of the merged node as [15], [16]:

$$\pi_m = \pi_i + \pi_j \text{ and } \eta_m = \frac{\pi_i \eta_i + \pi_j \eta_j}{\pi_m}. \quad (27)$$

Next, we compute the mean ( $\mu_m$ ) and the concentration ( $\kappa_m$ ) of the merged node from  $\eta_m$ , see Section 3.2.3.

The *region merging order* [31] sorts the adjacent regions that should be sequentially evaluated and merged. We define it using  $w_d$  (Eq. 21) among the adjacent nodes. The node  $v_j$ , which has minimum  $w_d$ , will be evaluated first using Eq. (26). In Fig. 6c, the region 3 should be chosen as the first one. We use  $w_d$  as the constraint due to its ability to discriminate (w.r.t.  $\mu$  and  $\kappa$ ) the regions.

Algorithm 2 provides the pseudo code of the proposed region merging method. It takes a set of regions obtained from Algorithm 1 as input and provides the final segmentation result as output. In the next Section, we evaluate the results obtained from the RGB-D segmentation method detailed in this paper.

---

#### Algorithm 2. Region Merging Algorithm

---

**Input:**  $R = \{r_i\}_{i=1,\dots,Z}$ ,  $G = (V, E)$ ,  $\kappa_p$ ,  $th_b$ ,  $th_d$  and  $th_r$

**Output:** Final segmentation after region merging.

Compute *candidacy*( $v_i$ ) for  $\{v_i\}_{i=1,\dots,Z}$  using Eq. (23);

Set  $i = 1$ ;

**foreach**  $i$  **do**

**if** *candidacy*( $v_i$ ) **is true then**

**while** no adjacent node of  $v_i$  is left to check **do**

      Sort the edges  $e_{ij}$  (defined with  $w_d(v_i, v_j)$  in Eq. (21)) in an ascending order;

      Evaluate each  $v_j$  with the *merging predicate*  $P_{ij}$  (Eq. (26));

**if**  $P_{ij}$  **is true then**

        Merge two nodes  $v_i$  and  $v_j$  and update the RAG;

        Start over again from sorting the adjacents  $e_{ij}$  of the node  $v_i$ .

**else**

        Check the next node

**end**

**end**

**end**

**end**

---

## 4 RESULTS AND DISCUSSION

### 4.1 Experiments and Results

In this Section, we evaluate the proposed method on the benchmark image database NYUD2 [8] which consists of



TABLE 1  
Sensitivity of JCSD-RM with Respect to the Parameters  $\{k, \kappa_p, th_b, th_d, th_r\}$

	$\{k, 5, 0.2, 3, 0.9\}$			$\{20, \kappa_p, 0.2, 3, 0.9\}$			$\{20, 5, th_b, 3, 0.9\}$			$\{20, 5, 0.2, th_d, 0.9\}$			$\{20, 5, 0.2, 3, th_r\}$		
	15	20	25	2	5	8	0.1	0.2	0.3	2	3	4	0.85	0.9	0.95
<b>VoI</b>	<b>2.17</b>	2.20	2.28	2.21	<b>2.20</b>	2.27	2.34	<b>2.20</b>	2.21	2.22	<b>2.20</b>	<b>2.20</b>	<b>2.20</b>	<b>2.20</b>	2.21
<b>BDE</b>	9.4	<b>8.97</b>	8.99	9.65	<b>8.97</b>	9.08	9.25	<b>8.97</b>	9.38	9.11	<b>8.97</b>	9.04	9.03	<b>8.97</b>	9.03
<b>PRI</b>	0.90	<b>0.91</b>	0.90	0.90	<b>0.91</b>	0.90	0.90	<b>0.91</b>	0.90	<b>0.91</b>	<b>0.91</b>	<b>0.91</b>	<b>0.91</b>	<b>0.91</b>	<b>0.91</b>
<b>GTRC</b>	<b>0.60</b>	<b>0.60</b>	0.59	0.58	<b>0.60</b>	0.58	0.56	<b>0.60</b>	0.59	0.59	<b>0.60</b>	<b>0.60</b>	<b>0.60</b>	<b>0.60</b>	<b>0.60</b>

1,449 indoor images with RGB, depth and ground-truth information. However, we use the ground truth labels used in [6], because they are corrected for the confused regions, which are labeled as white in the original database. We convert (using MATLAB function) the RGB color information into  $L^*a^*b^*$  (CIELAB space) color because of its perceptual accuracy [24], [47]. For the depth images, we compute the 3D coordinates and surface normals using the toolbox of [9].

Our clustering method requires to set initial labels of the pixels and the number of clusters  $k$ . We initialize it using a combined k-means method with  $k = 20$ . In this k-means method, the total distance between the cluster center and each observation is computed by adding the euclidean (for normalized color and 3D positions) and Cosine (for surface normal) distances. For the region merging we empirically set the thresholds as:  $\kappa_p = 5$  to state that a region is planar,  $th_b = 0.2$  to state that there is a boundary among two regions,  $th_d = 3$  to state that there is a dissimilarity between two regions and  $th_r = 0.9$  to determine the goodness of a plane fitting.

We evaluate performance using standard benchmarks [35] which are applied to compare the test and ground truth segmentation: (1) variation of information, it measures the distance between two segmentations in terms of their average conditional entropy; (2) boundary displacement error [36], it measures the average displacement between the boundaries of two segmentations; (3) probability rand index, it measures likelihood of a pair of pixels that has same label; (4) Ground truth region covering (*GTRC*), it measures the region overlap between ground truth and test and (5) Boundary based F-measure, a boundary measure based on precision-recall framework [35]. With these criteria a segmentation is better if *VoI* and *BDE* are smaller whereas *PRI*, *GTRC* and *BFM* are larger.

In our experiments, we obtain two sets of segmentation results by using the *Fisher* and *Watson* distribution with *JCSD-RM*. In general, the *Fisher* distribution is the fundamental choice for fitting the normals (subject to unambiguity). In our previous work [39], we considered the *Watson* distribution due to the ambiguity in the normals. In this work, yet we consider the *Watson* distribution to study its performance for unambiguous directions. Interestingly, we observe that the results from both *Fisher* and *Watson* distributions are almost equivalent w.r.t. the different measures and computation time. Therefore, in order to avoid redundancy, in this section we do not explicitly present the results of *JCSD-RM* based on the *Watson* distribution. Besides, we compare the results with those obtained from [39] and observe that the unambiguous normals used in this new paper certainly improves the performance of the overall segmentation task.

We begin the experiments by studying the sensitivity of the proposed method w.r.t. the parameters ( $k, \kappa_p, th_b, th_d, th_r$ ), which are presented in Table 1. The parameter  $k$  is related to the clustering method (Section 3.2) while  $\kappa_p, th_b, th_d$  and  $th_r$  are related to the region merging method (Section 3.3). From Table 1, using the standard deviation of the normalized values of each evaluation metric, we can sort scores in a descending order as:  $PRI(0.0057) < VoI(0.0191) < GTRC(0.0198) < BDE(0.021)$ . This means that the BDE measure provides the most discriminating view w.r.t. the parameters and according to it our choice of the threshold values are justified (see Table 1 where the BDE scores show that the chosen thresholds uniquely provide best results). Moreover, such choice can also be justified using the other measures. Additional comments about these heuristics based parameters are:

- Number of clusters  $k$  is inversely related to the number of pixels in a cluster. In segmentation, a smaller  $k$  causes a loss of details in the scene, i.e., an under-segmentation, while higher  $k$  splits the scene into more regions, i.e., an over-segmentation. Besides, the computation time of *JCSD-RM* is proportional to  $k$ .
- We set  $\kappa_p$  based on the study we did on NYUD2 (see Appendix B for details) which reveals that planar surfaces can be characterized with concentration  $\kappa \geq 5$ . While, a lower  $\kappa$  value enables to merge non-planar surfaces, a higher value may decrease the probability to merge true planar surfaces.
- Following the OWT-UCM [35] method, we empirically set the value of  $th_b$ .
- We also set  $th_d$  empirically. In theory two regions which have their normals oriented in the same direction should have a negligible Bregman divergence value. However, the inaccurate computation of the shape features and the presence of noise in the acquired depth information often causes the Bregman divergence to be high. From our experience with the images of NYUD2,  $th_d$  should be within the range between 2 to 4.
- The parameter  $th_r = 0.9$  is set by following [7]. Our results in Table 1 show further justification for this value.

Next, we compare the proposed method joint color-spatial-directional clustering and region merging (*JCSD-RM*) with several unsupervised RGB-D segmentation methods such as: RGB-D extension of OWT-UCM [9] (UCM-RGBD), modified Graph Based segmentation [42] with color-depth-normal (GBS-CDN), Geometry and Color Fusion method [10] (GCF) and the Scene Parsing Method [7] (SP). For the UCM-RGBD method we obtain best score with threshold

TABLE 2  
Comparison with the State of the Art

	VoI	BDE	PRI	GTRC	BFM
UCM-RGBD	2.35	9.11	0.90	0.57	<b>0.63</b>
GBS-CDN	2.32	13.23	0.81	0.49	0.53
GCF	3.09	14.23	0.84	0.35	0.42
SP	3.15	10.74	0.85	0.44	0.50
JCSD	2.68	10.00	0.87	0.46	0.46
JCSD-RM	<b>2.2</b>	<b>8.97</b>	<b>0.91</b>	<b>0.6</b>	0.61

Methods: UCM-RGBD [9], GBS-CDN [42], GCF [10], SP [7], JCSD and JCSD-RM (proposed). Boldface indicates best results.

value 0.1. The best results from GBS-CDN method are obtained by using  $\sigma = 0.4$ . To obtain the optimal multiplier ( $\lambda$ ) in GCF [10] we exploit the range 0.5 to 2.5. For the SP method, we scaled the depth values (1/0.1 to 1/10 in meters) to use author's source code [7].

Table 2 presents (best appears as bold) the comparison w.r.t. the average score of the benchmarks. Results show that JCSD-RM performs best according to PRI, VoI, GTRC and BDE. Moreover, it is comparable according to BFM. The reason is that, BFM favors methods like UCM-RGBD which is specialized in contours detection. On the other hand, JCSD clustering method provides an approximation (see, e.g., Fig. 5) of the object boundary which is often coarse. This can be improved by developing a spatially constrained clustering method, such as [26]. A better boundary approximation will subsequently improve the performance of the RM method. Therefore, we can say that JCSD-RM could be further improved by incorporating the boundary information more efficiently.

Ground Truth Region Covering has been chosen as one of most prominent measure of evaluation for segmentation methods [6], [35]. In Tables 1 and 2, we observed that it provides discriminative score to evaluate and differentiate among the different state-of-the-art methods. Fig. 7 provides further analysis on NYUD2 [8] using histograms of GTRC scores. We observe that, while the JCSD-RM and UCM-RGBD covers quite similar regions in the histogram, others are significantly different especially in the higher GTRC region. Particularly, the JCSD-RM has lower percentage of images with low GTRC score region and higher percentage for the high GTRC score region.

In order to conduct the experiments we used a 64 bit machine with Intel Xenon CPU and 16 GB RAM. The JCSD-RM method is implemented in MATLAB, which on average takes 38 seconds, where 31 seconds for the clustering and 7 seconds for region merging. In contrast, UCM-RGBD (MATLAB and C++) takes 110 seconds. Therefore, JCSD-RM is  $\approx 3$  times faster<sup>7</sup> than UCM-RGBD. Moreover, we believe that implementing JCSD-RM in C++ will significantly reduce the computation time.

To further analyze the computation time of JCSD-RM, we run it for different image scales. Table 3 presents relevant information from which we see that the reduction rate of JCSD computation time (in sec) w.r.t. different scales is approximately equivalent to the reduction rate of the number of pixels.

7. To perform a fair comparison, we conducted this experiment with half scaled image. This is due to the fact that the computational resource did not support to run UCM-RGBD for the full scale image.

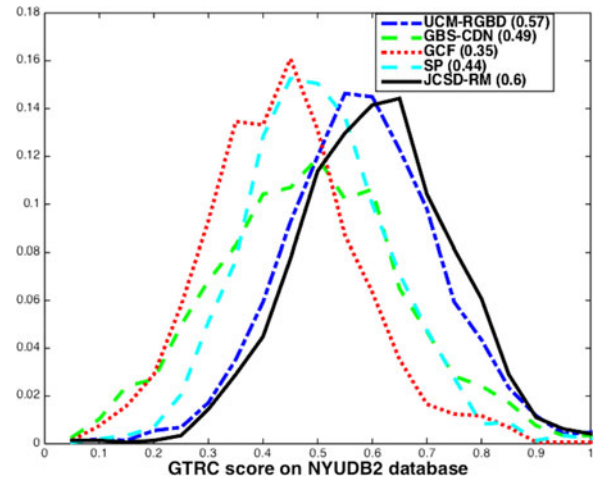


Fig. 7. Histogram of GTRC [35] scores of different methods.

## 4.2 Discussion

Several segmentation results are illustrated in Fig. 8. These examples confirm that the segmentation from JCSD-RM (our proposed) and UCM-RGBD are competitive. However, they let us note several differences: (a) JCSD-RM is better in providing the details of indoor scene structures whereas UCM-RGBD loses them sometimes (see ex. rows 3 to 5); (b) UCM-RGBD provides better estimation of the object boundaries whereas JCSD-RM gives a rough boundary and (c) UCM-RGBD shows more sensitivity on color whereas JCSD-RM is more sensitive on directions. The GBS-CDN method provides visually pleasing results, however it often tends to loose details (see ex. rows 1 to 4) of the scene structure (e.g., merges wall with ceiling). Results from the SP method seems to be severely sensitive to the varying illumination and rough changes in surfaces (see ex. row 3). The GCF method performs over-segmentation (see ex. rows 1, 3, and rows 5-7) or under-segmentation (see ex. rows 2 and 4), which is a drawback of such algorithm as it is often unable to estimate the correct number of clusters in real data. Moreover, the GCF method often fails to discriminate major surface orientations (see ex. rows 1, 2 and 4) as it does not consider the direction of surfaces (normals).

Now, in Fig. 9 let us focus and analyze some segmentation examples which have lower (less than 0.4) GTRC score. Average GTRC score of JCSD-RM is 0.6 (see Tables 1 and 2). Results show several cases for low scores:

- JCSD-RM tends to provide more details (over-segment) while the ground truth keeps minimum details, see ex. columns 1 to 3, and 5 in Fig. 9.
- JCSD-RM does not provide enough details (under-segment) while the ground truth does, see ex. columns 4 and 6 in Fig. 9. This is a very difficult case, as looking at the images we can see that the under-segmented

TABLE 3  
Computation Time of JCSD-RM w.r.t. Different Image Scales

Scale	1	1/2	1/4	1/8
Num. pixels	239k	60k	15k	4k
JCSD (req. time in sec)	132	31	8	1.5
RM (req. time in sec)	42	7	1.4	0.33



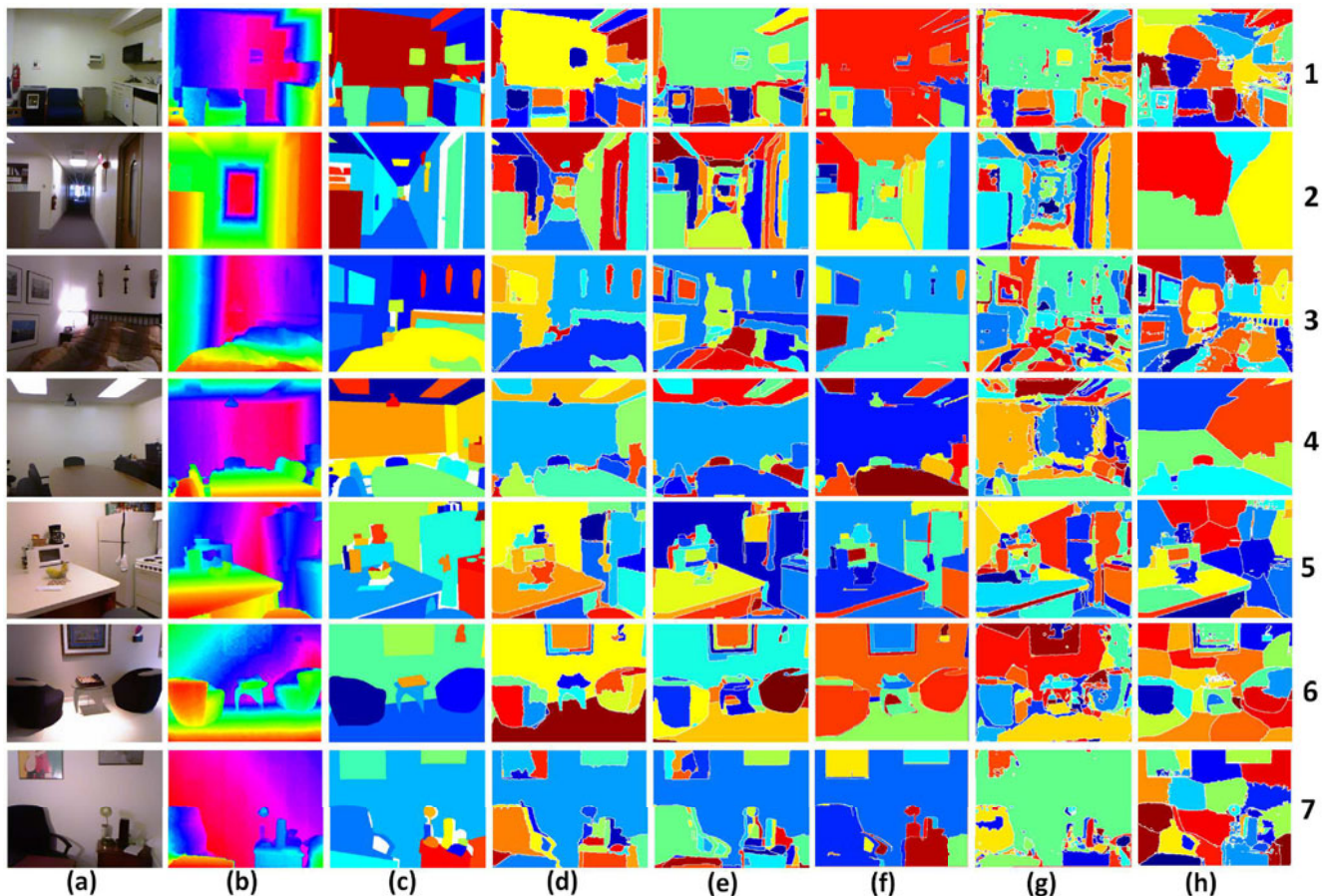


Fig. 8. Segmentation examples (from top to bottom) on NYU RGB-D database (NYUD2). (a) Input Color image (b) Input Depth image (c) Ground truth (d) JCSD-RM (*our proposed*) (e) UCM-RGBD [9] (f) GBS-CDN [42] (g) SP [7] and (h) GCF [10].

regions have similar color, depth and normal which in a general case is difficult to segment without additional knowledge.

- Example column 7 shows a characteristic example of JCSD-RM, which is to be biased on surface normals. This causes the furniture (sofa) to be segmented into several parts. Perhaps this can be improved by incorporating color based merging heuristics in our region merging method.

Now, let us focus particularly on the JCSD method, which is based on a statistical image generation model defined

from the naïve Bayes assumption [12], [13]. Computer vision or data analysis experts may question about the assumption of independence between the color, depth and normal. While we partially agree with the experts, in a first step we made this assumption because of two reasons: (a) propose a simplified method to understand the underlying grouping mechanisms of different image features in a combined fashion and (b) to empirically verify (in a second step) the relevance of this assumption in an unsupervised context.

Table 4 provides results w.r.t. several alternative models. First, we consider a unified model, called All\_GMM, that fits

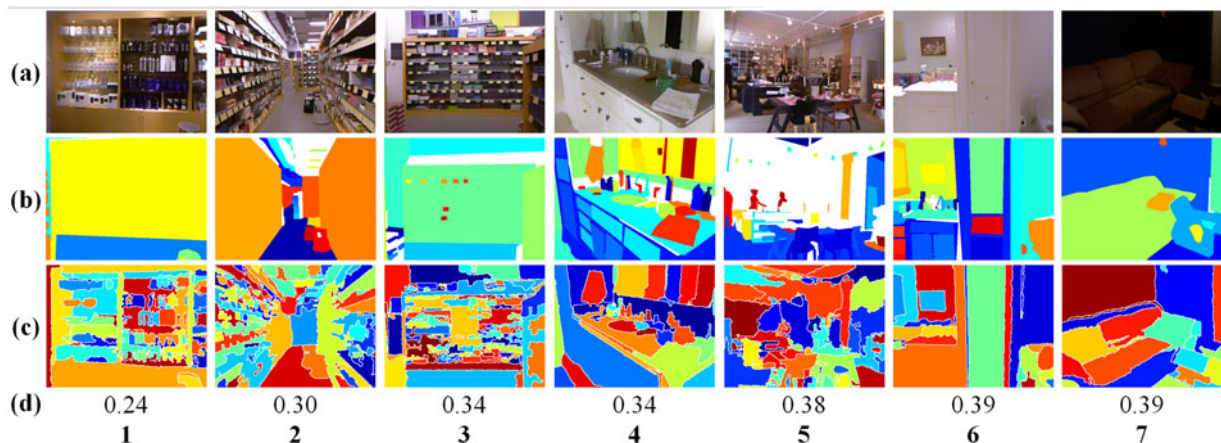


Fig. 9. Segmentation examples with lower GTRC scores (less than 0.4). (a) Input Color Image (b) Ground Truth Segmentation (c) Segmentation with the JCSD-RM method and (d) GTRC score.



TABLE 4  
Comparison among Different Image Models and the Clustering Results with/without the Region Merging Method

	VoI	BDE	PRI	GTRC
JCSD	<b>2.68</b>	<b>10.00</b>	<b>0.87</b>	<b>0.46</b>
All_GMM	3.01	11.04	<b>0.87</b>	0.43
PCA_GMM	3.94	12.01	0.85	0.34
Ind_GMM	3.22	11.04	0.86	0.41
JCSD+RM	<b>2.21</b>	<b>8.97</b>	<b>0.91</b>	<b>0.60</b>
All_GMM + RM	2.41	9.28	0.90	0.58
PCA_GMM+RM	2.85	10.25	0.88	0.50
Ind_GMM+RM	2.62	10.15	0.89	0.53

*Boldface indicates the best results among similar methods (with and without RM).*

a Gaussian Mixture Model (GMM) for all features together, i.e., no independence among features. Results show that, while JCSD is same with All\_GMM only for PRI measure, it is better w.r.t. the VoI, BDE and GTRC measures. In terms of average computing time, JCSD takes 31 sec and All\_GMM takes 45 sec, i.e., JCSD is 1.45 times faster. Moreover, in numerous images, All\_GMM fails due to the ill-conditioned covariance matrix. Based on this observation, we consider a different model, called PCA\_GMM, which reduces the features using PCA method by considering 95 percent variances of the data. On average the reduced feature dimension was 7. Results show that, while the performance decreases remarkably, there was no potential gain on computation time. From these results, we can see that our simplified image model is better w.r.t. the standard measures and computation time.

Another JCSD clustering related issue is the assumption of a known maximum number of clusters  $k_{max}$ . In the context of mixture model based clustering, numerous methods exist to automatically select the number of clusters. For example, [21] used the Bayesian Information Criteria (BIC), [22] employed the Minimum Message Length (MML), [20] proposed the Integrated Completed Likelihood (ICL) and [24] applied  $\Phi_\beta$  criterion. Besides these criteria, in our previous work [15], we proposed a modification of the slope heuristic based method. For this work, we developed and experimented (not presented in this paper) all of the above-mentioned methods and observed a large number of over-segmented and under-segmented images. We realized that, while it is difficult to improve an under-segmentation, it is easier to improve an over-segmentation, e.g., by using a region merging method. Therefore, we decided to avoid the idea of automatic number of clusters selection and use the notion of a predefined  $k_{max}$ .

Now, let us focus on a different concern related to the use of directional (Fisher or Watson) distribution for the surface normals. An expert could easily argue that the Gaussian distribution can be used in the place of directional distribution. Although, in our previous work [15] on clustering normals, we have shown that directional distribution is more appropriate than Gaussian distribution, here we provide further verification within the context of joint clustering with independent assumption. We consider JCSD type model, called Ind\_GMM, which replaces the directional distribution with the Gaussian distribution. Results show that, JCSD is better w.r.t. all evaluation measures, which further demonstrates the efficiency and relevance of our proposed image model.

Interestingly, we observe that Ind\_GMM provides slightly lower performance than All\_GMM, which reveals that the independence assumption is context dependent and does not necessarily provide better results if the unsupervised classifier (here GMM) remains same and may provide better results if we understand the heterogeneous (i.e., combination of different types of features) data and build a model to handle them appropriately.

Comparing JCSD with JCSD-RM (Tables 2 and 4), we can decompose the contributions of *clustering* and *region merging* in JCSD-RM. We see that *region merging* improves clustering output from 0.46 to 0.6 (30.43 percent) in GTRC. We believe that JCSD-RM can be improved and extended further in the following ways:

- Including a pre-processing stage, which is necessary because the shape features are often computed inaccurately due to noise and quantization [45]. Moreover, we observed significant noise in NYUD2 color images which were captured especially in low light condition. A method like Scene-SIRFS (shape, illumination and reflectance from shading) [45], which recover the intrinsic scene properties, can be used for pre-processing purpose.
- Enhancing the clustering method by adding contour information [35] efficiently. Additionally, we may consider spatially constrained model such as [26] which incorporates boundary information by adding spatially varying constraints in the clustering task.
- Enhancing the region merging method with color information. To this aim, we can exploit the estimated reflectance information (using [45]), such that the varying illumination is discounted.

## 5 CONCLUSION

We proposed an unsupervised indoor RGB-D scene segmentation method. Our method is based on a statistical image generation model, which provides a theoretical basis for fusing different cues (e.g., color and depth) of an image. In order to cluster w.r.t. the image model, we developed an efficient joint color-spatial-directional clustering method based on Bregman divergence. Additionally, we proposed a region merging method that exploits the planar statistics of the image regions. We evaluated the proposed method with a database of benchmark RGB-D images and using widely accepted evaluation metrics. Results show that our method is competitive w.r.t. the state of the art and opens interesting perspectives for fusing color and geometry. We foresee several possible extensions of our method: more complex image model and clustering with additional features, region merging with additional hypothesis based on color. Moreover, we believe that the methodology proposed in this paper is equally applicable and extendable for other complex tasks, such as joint image-speech data analysis.

## APPENDIX A

### BREGMAN DIVERGENCE—AN ALTERNATIVE FORMULATION AND RELATIONSHIP

A multivariate probability density function  $f(\mathbf{x}|\theta)$  belongs to the regular exponential family if it has the following form [17], [29]:

$$f(\mathbf{x}|\theta) = \exp(\langle t(\mathbf{x}), \theta \rangle) - F(\theta) + k(\mathbf{x}). \quad (28)$$

Here,  $t(\mathbf{x})$  is the sufficient statistics,  $\theta$  is the natural parameters,  $F(\theta)$  is the log normalizing function,  $k(\mathbf{x})$  is the carrier measure and  $\langle \cdot, \cdot \rangle$  is the inner product.

The expectation of the sufficient statistics  $t(\mathbf{x})$  is called the expectation parameter,  $\eta = E[t(\mathbf{x})]$ . There exists a one-to-one correspondence between expectation ( $\eta$ ) and natural ( $\theta$ ) parameters, which exhibits dual relationships among the parameters and functions as [29]:

$$\eta = \nabla F(\theta) \quad \text{and} \quad \theta = (\nabla F)^{-1}(\eta) \quad (29)$$

and

$$G(\eta) = \langle (\nabla F)^{-1}(\eta), \eta \rangle - F((\nabla F)^{-1}(\eta)). \quad (30)$$

Here,  $\nabla F$  is the gradient of  $F$ .  $G(\cdot)$  is the Legendre dual of the log normalizing function  $F(\cdot)$ . See details in [29, Section 3.2].

For a strictly convex function  $F(\cdot)$ , Bregman Divergence,  $D_F(\theta_1, \theta_2)$  can be formally defined as [29]:

$$D_F(\theta_1, \theta_2) = F(\theta_1) - F(\theta_2) - \langle \theta_1 - \theta_2, \nabla F(\theta_2) \rangle. \quad (31)$$

$D_F(\theta_1, \theta_2)$  measures the distance using the tangent function at  $\theta_2$  to approximate  $F$ . This can be seen as the distance between the first order Taylor approximation to  $F$  at  $\theta_2$  and the function evaluated at  $\theta_1$  [17]. The one-to-one correspondence in Eq. (29) provides the dual form of BD (of Eq. (31)) as:

$$D_G(\eta_1, \eta_2) = G(\eta_1) - G(\eta_2) - \langle \eta_1 - \eta_2, \nabla G(\eta_2) \rangle. \quad (32)$$

Due to the bijection between BD and the Exponential families, Eq. (31) and (32) can be used to measure the dissimilarity between distributions of the same family. The bijection is expressed as:  $f(\mathbf{x}|\theta) = \exp(-D_G(t(\mathbf{x}), \eta))J_G(\mathbf{x})$  where  $J_G$  is a uniquely determined function. We used this formulation in Eq. (3) of this paper. For more details, see [29, Theorem 3].

Bregman divergences generalize the squared euclidean distance, Mahalanobis distance, Kullback-Leibler divergence, Itakura-Saito divergence etc. See Table 1 of [29] and [37] for a list and corresponding  $D_F(\cdot, \cdot)$ . Besides, BD has the following interesting properties [37]:

- **Non-negativity:** The strict convexity of  $F$  implies that, for any  $\theta_1$  and  $\theta_2$ ,  $D_F(\theta_1, \theta_2) \geq 0$  and  $D_F(\theta_1, \theta_2) = 0$  if and only if  $\theta_1 = \theta_2$ .
- **Convexity:** Function  $D_F(\theta_1, \theta_2)$  is convex in its first argument  $\theta_1$  but not necessarily in the second argument  $\theta_2$ .
- **Linearity:** BD is a linear operator, i.e., for any two strictly convex functions  $F_1$  and  $F_2$  and  $\lambda \geq 0$ :

$$D_{F_1+\lambda F_2}(\theta_1, \theta_2) = D_{F_1}(\theta_1, \theta_2) + \lambda D_{F_2}(\theta_1, \theta_2).$$

## APPENDIX B

### STUDY OF PLANAR STATISTICS

For this study we applied clustering (with Fisher Mixture Model [15]) on surface normals of the images of NYUD2 [8]. Fig. 10 illustrates the histograms of  $\kappa$  (concentration of normals) values for planar and non-planar surfaces. These histograms have been obtained from analyzing four category of surfaces: (1) planar; (2) non-planar ; (3) planar + non-

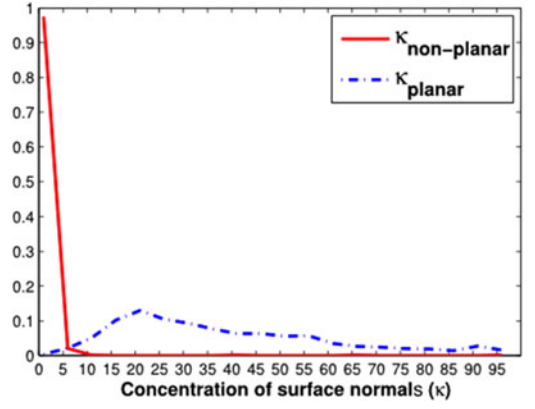


Fig. 10. Histogram of  $\kappa$  for planar and non-planar surfaces.

planar and (d) unknown (category not sure). A total of 5,410 RoIs were analyzed, among them 2,559 represented planar surface meanwhile 793 represented non-planar surfaces. Then we computed the histogram of  $\kappa$  values for all these RoIs. We observed that 99.88 percent of planar surfaces has  $\kappa > 5$  and 99.5 percent of non-planar surfaces has  $\kappa < 5$ . This heuristically shows that the assumption about planar statistics used in Eq. (23), based on  $\kappa$  values, is appropriate for region merging.

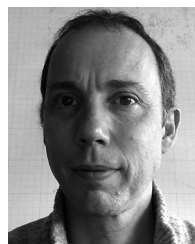
## REFERENCES

- [1] R. Szeliski, *Computer Vision: Algorithms and Applications*. New York, NY, USA: Springer, 2011.
- [2] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
- [3] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1452–1458, Nov. 2004.
- [4] Z. Zhang, "Microsoft Kinect sensor and its effect," *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, Apr.-Jun. 2012.
- [5] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Trans. Cybernetics*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.
- [6] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 564–571.
- [7] C. J. Taylor and A. Cowley, "Parsing indoor scenes using RGB-D imagery," in *Proc. Robot.: Sci. Syst. VIII*, 2013, pp. 401–408.
- [8] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.
- [9] X. Ren, L. Bo, and D. Fox, "RGB-D scene labeling: Features and algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 2759–2766.
- [10] C. Dal Muto, P. Zanuttigh, and G. M. Cortelazzo, "Fusion of geometry and color information for scene segmentation," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 5, pp. 505–521, Sep. 2012.
- [11] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3D point clouds for indoor scenes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, vol. 1, no. 2, p. 4.
- [12] H. Zhang, "The optimality of Naive Bayes," *AA*, vol. 1, no. 2, p. 3, 2004.
- [13] D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," in *Proc. 10th Eur. Conf. Mach. Learn.*, 1998, pp. 4–15.
- [14] K. V. Mardia and P. E. Jupp, *Directional Statistics*, vol. 494. Hoboken, NJ, USA: Wiley, 2009.
- [15] M. A. Hasnat, O. Alata, and A. Trémeau, "Model-based hierarchical clustering with Bregman divergence and Fisher mixture model: Application to depth image analysis," *Statist. Comput.*, pp. 1–20, 2015.
- [16] M. A. Hasnat, O. Alata, and A. Trémeau, "Unsupervised clustering of depth images using Watson mixture model," in *Proc. 22nd Int. Conf. Pattern Recog.*, 2014, pp. 214–219.

- [17] M. Liu, B. C. Vemuri, S.-I. Amari, and F. Nielsen, "Shape retrieval using hierarchical total Bregman soft clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2407–2419, Dec. 2012.
- [18] F. Nielsen and V. Garcia. (2009). Statistical exponential families: A digest with flash cards. CoRR, vol. abs/0911.4863 [Online]. Available: <http://arxiv.org/abs/0911.4863>
- [19] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *Comput. J.*, vol. 41, no. 8, pp. 578–588, 1998.
- [20] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 7, pp. 719–725, Jul. 2000.
- [21] C. Fraley and A. E. Raftery, "Model-based methods of classification: Using the mclust software in chemometrics," *J. Statistical Softw.*, vol. 18, no. 6, pp. 1–13, 2007.
- [22] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 381–396, Mar. 2002.
- [23] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1546–1562, Sep. 2007.
- [24] O. Alata and L. Quintard, "Is there a best color space for color image characterization or representation based on multivariate Gaussian mixture model?" *Comput. Vis. Image Understanding*, vol. 113, no. 8, pp. 867–877, 2009.
- [25] V. Garcia and F. Nielsen, "Simplification and hierarchical representations of mixtures of exponential families," *Signal Process.*, vol. 90, no. 12, pp. 3197–3212, 2010.
- [26] T. M. Nguyen and Q. Wu, "Fast and robust spatially constrained Gaussian mixture model for image segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 4, pp. 621–635, Apr. 2013.
- [27] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, 2nd ed., series Wiley series in probability and statistics. Hoboken, NJ, USA: Wiley, 2008.
- [28] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 1. New York, NY, USA: Springer, 2006.
- [29] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, 2005.
- [30] A. Trémeau and P. Colantoni, "Regions adjacency graph applied to color image segmentation," *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 735–744, Apr. 2000.
- [31] B. Peng and D. Zhang, "Automatic image segmentation by dynamic region merging," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3592–3605, Dec. 2011.
- [32] A. Martínez-Usó, F. Pla, and P. García-Sevilla, "Unsupervised colour image segmentation by low-level perceptual grouping," *Pattern Anal. Appl.*, vol. 16, no. 4, pp. 581–594, 2013.
- [33] R. B. Rusu, *Semantic 3D Object Maps for Everyday Robot Manipulation*. New York, NY, USA: Springer, 2013.
- [34] C. J. Taylor and A. Cowley, "Segmentation and analysis of RGB-D data," in *Proc. RSS Workshop RGB-D Cameras*, 2011.
- [35] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
- [36] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration," in *Proc. 7th Eur. Conf. Comput. Vis.*, 2002, pp. 408–422.
- [37] J.-D. Boissonnat, F. Nielsen, and R. Nock, "Bregman Voronoi diagrams," *Discr. Comput. Geometry*, vol. 44, no. 2, pp. 281–307, 2010.
- [38] M. A. Hasnat, O. Alata, and A. Trémeau, "Hierarchical 3-d von Mises-Fisher mixture model," in *Proc. 1st Workshop Divergences Divergence Learn.*, 2013.
- [39] M. A. Hasnat, O. Alata, and A. Trémeau, "Unsupervised RGB-D image segmentation using joint clustering and region merging," in *Proc. Brit. Mach. Vis. Conf.*, 2014.
- [40] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, *Time-of-Flight Cameras and Microsoft Kinect*. New York, NY, USA: Springer, 2012.
- [41] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using RGB-D cameras," in *Proc. Robot Soccer World Cup XV*, 2012, pp. 306–317.
- [42] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.
- [43] Y. Niu, Y. Geng, X. Li, and F. Liu, "Leveraging stereopsis for saliency analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 454–461.
- [44] J. Strom, A. Richardson, and E. Olson, "Graph-based segmentation for colored 3D laser point clouds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 2131–2136.
- [45] J. T. Barron and J. Malik, "Intrinsic scene properties from a single RGB-D image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 17–24.
- [46] C. Biernacki, G. Celeux, and G. Govaert, "Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models," *Comput. Statist. Data Anal.*, vol. 41, no. 3, pp. 561–575, 2003.
- [47] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 409–416.



**Md. Abul Hasnat** received the BSc degree from BRAC University, Bangladesh, in 2006, the MSc degree from Erasmus Mundus CIMET in 2011, and the PhD degree in "image, vision and signal" from Jean Monnet University, France, in 2014. He is a postdoctoral researcher at the École Centrale de Lyon. He worked on temporal/evolutionary data clustering, RGB-D image segmentation, spectral image reconstruction, and document image analysis. His research activities are focused in the area of computer vision, image processing, machine learning, and data mining.



**Olivier Alata** received the ENSERB (National School of Electronics and Radio-Electricity of Bordeaux) engineering degree in 1992 and the PhD degree from the University of Bordeaux in January 1998. From 1998 to 2011, he was an assistant professor at the University of Poitiers. Since 2011, he has been a professor at Telecom Saint-Etienne, Jean Monnet University, where he conducts his research at the Hubert Curien Laboratory, UMR CNRS 5516, in the Computer Science Department. His works mainly concern stochastic modeling and development of methods for the analysis and processing of signals and images (texture, segmentation, and reconstruction).



**Alain Trémeau** is a professor in computer vision at the Jean Monnet University, Saint-Etienne, France. He conducts research at the Hubert Curien Laboratory (UMR CNRS 5516) in the Computer Science Department. His research activities are mainly focused on mathematical imaging and color science with reference to human vision and perception. He wrote numerous scientific papers and book chapters, especially in the fields of color imaging and processing and in computer vision. His research activities cover a large set of applications, with a recent focus on color in cultural heritage applications.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).