A METHODOLOGY FOR IMAGE SEGMENTATION USING SUPERPIXELS AND DEPTH INFORMATION

Isail Salazar Acosta

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2017

A METHODOLOGY FOR IMAGE SEGMENTATION USING SUPERPIXELS AND DEPTH INFORMATION

Isail Salazar Acosta

A thesis presented in fulfillment of the requirements for the degree of Electronic Engineer

Advisor:
Fabio Martínez Carrillo
PhD in Systems and Computer Engineering

Co-Advisor:
Said David Pertuz Arroyo
PhD in Computer Science

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2017

Acknowledgements

This work would not have been possible without the efforts of my advisors Fabio Martínez and Said Pertuz. I would like to thank them for their patience and the many hours they spent providing insight, advice and direction.

Many thanks to my parents for being a source of stability and support in my life, allowing me to develop the present work in a comfortable way.

Many thanks to the School of Electric, Electronic and Telecommunications Engineering (E3T) and the Industrial University of Santander (UIS) for training me as a professional and teaching me all the bases to accomplish this work.

CONTENT

LIST OF FIGURES

RESUMEN

Título: Metodología para la Segmentación de Imágenes Utilizando Superpixeles e Información de Profundidad[1]

Autor: Isail Salazar Acosta[2]

Palabras Clave: Segmentación de Imágenes, Sobre-segmentación, Superpixeles, Cámaras RGB-D, Imágenes RGB-D, Nube de puntos 3D.

DESCRIPCIÓN:

Los algoritmos clásicos de segmentación de imágenes explotan la detección de similaridades y discontinuidades en diferentes patrones visuales con el fin de detectar y diferenciar regiones de interés en una imagen. Sin embargo, debido a la alta variabilidad e incertidumbre de los datos presentes en las mismas, se hace difícil producir resultados acertados. En este sentido, la segmentación basada solo en color a menudo no es suficiente para un gran porcentaje de imágenes naturales. Interesantemente, en los últimos años, la disponibilidad de cámaras RGB-D (color más profundidad) de bajo costo (p.ej., la Kinect de Microsoft) ha abierto nuevas posibilidades de investigación. Este trabajo presenta una metodología que permite la integración de la información de profundidad al problema de la segmentación. Específicamente, la imagen de color es sobre-segmentada en una determinada cantidad de superpixeles que luego son procesados en un enfoque de fusión de regiones tomando en cuenta la profundidad. Para este propósito, una nube de puntos 3D se genera a partir de los datos de profundidad a fin de detectar características relevantes en el espacio 3D: planos y contornos. Éstas son luego traducidas en segmentaciones incompletas que sirven de soporte al proceso de fusión de regiones. La salida es una segmentación final a partir de los superpixeles. Los experimentos fueron conducidos sobre la base de datos de imágenes NYU-Depth V2. Los resultados obtenidos reportan mejoras considerables con respecto a la segmentación clásica basada en color según medidas de desempeño comunes en el estado del arte.

---

[1] Trabajo de Grado

[2] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Fabio Martínez Carrillo, PhD.

ABSTRACT

Title: A Methodology for Image Segmentation Using Superpixels and Depth Information[1]

Author: Isail Salazar Acosta[2]

Keywords: Image Segmentation, Over-segmentation, Superpixels, RGB-D Cameras, RGB-D Images, 3D Point Cloud.

DESCRIPTION:

Classical image segmentation algorithms exploit the detection of similarities and discontinuities in different visual patterns (e.g., color, texture, brightness) to detect and differentiate multiple regions of interest in an image. However, due to the high variability and uncertainty of image data, it is a difficult task to achieve accurate results. In this way, segmentation based just in color is often not sufficient for a large percentage of natural images. Interestingly enough, in the last few years, the availability of low cost color-plus-depth (RGB-D) cameras (e.g., Microsoft's Kinect) has opened up new research possibilities. This work presents a methodology that allows the integration of depth information to the segmentation problem. Specifically, the color image is over-segmented into several superpixels to thereafter be processed by a depth-aware region merging approach. For this purpose, a 3D point cloud is reconstructed from the depth information to detect relevant 3D features: planes and contours. These features are then translated into coarse segmentations which serve as support inputs to the region merging process. The output is a final segmentation from superpixels. Experiments were conducted on the NYU-Depth V2 (NYUD2) dataset. Obtained results report considerable improvements over classic color segmentation in terms of state-of-the-art performance measures and are expected to pave the way for future research in scene understanding and RGB-D image segmentation.

---

[1] Bachelor Thesis

[2] Faculty of Physics-Mechanics Engineering. School of Electric, Electronic and Telecommunications Engineering. Advisor: Fabio Martínez Carrillo, PhD.

INTRODUCTION

Segmentation is a fundamental yet challenging problem in digital image processing and computer vision. Its goal is to partition an image into a set of regions that have coherent properties. These can represent objects or have one specific meaning to the particular application. Each of the pixels in a region share similar visual cues (e.g., color, edges, texture) or more complex features like depth [1] and focus level [2]. The wide range of important applications that rely on image segmentation, such as medical imaging [3], machine vision [4], object recognition [5] and content-based image retrieval [6], have motivated the development of an enormous quantity of techniques.

In general, there are three families of methodologies that stand out in the state-of-the-art: graph-based [7–12], mode seeking [13–16], and region merging [17–20]. However, since these classic segmentation methods are designed for processing RGB chromatic components, the diversity and ambiguity of the inferred visual cues are very high, making them unable to produce an optimal or correct segmentation to all kinds of images. Instead, these methods usually tend to generate *under-* or *over-segmentation*. The first case occurs when adjacent objects with close RGB values cannot be properly distinguished, as illustrated in Fig. 1 (a). In the second case, coherent regions are split into many segments due to the presence of different colors, getting a complex partition of the image, like in Fig. 1 (b). Shadows, poor illumination and light switches are also problems to consider.

Currently, new technologies have made possible the acquisition of additional information in an image, allowing to deal with the challenging issues of RGB data. Attractive examples are the powerful and cheap *RGB-D cameras* like the Microsoft Kinect and the Asus Xtion. These can provide color and depth information (namely *RGB-D image*) of indoor environments. This fact is due to limitations on the depth sensor technology,

Figure 1. **Under- and over-segmentation.** **(a)** Segmentation using [11], in which case the algorithm fails to differentiate the curtain from the base structure of the sink, clearly because of its high color similarity. **(b)** Segmentation using [17], where the color and light variations cause a notable over-segmentation in the wall and the bed sheet.

which does not work in outdoors. Despite this, the rich amount of indoor applications in computer vision [21] make the study of RGB-D images worth the effort.

RGB-D images can then enable the understanding of the scene's 3D geometry. In fact, the depth component can be projected into a *3D point cloud* [22], a collection of data points in the XYZ coordinates relative to the camera's viewpoint. This knowledge can greatly simplify the grouping of coherent regions, as structural relationships are more easily observable in 3D space rather than color space. For that reason, a variety of methods that allow the analysis of 3D data have been in recent development [21]. Remarkable approaches are based in surface normal estimation [23] and plane detection [24–26].

The problem of image segmentation using RGB-D images has also been addressed [1, 27–30]. These methods propose the computation of several color and 3D features that then are processed in a joint manner. However, this is quite a difficult task due to the different nature of color and depth data. Alternatively, this work tackles the segmentation problem by considering color and depth information in a separate fashion. The basic idea is to take advantage of different segmentations on both color and 3D space as follows:

- A *superpixel* segmentation, where the term superpixel refers to more compact segments which are local and better preserve object details. This is basically an over-segmentation and there are many algorithms explicitly developed for this purpose [31].
- A classical segmentation, which is the usual attempt to partition the image of interest.
- A planar segmentation, performed on the 3D point cloud by means of a plane detection process.
- A 3D-edge segmentation, obtained by using the 3D gradients proposed in [1].

Each of these segmentations can be treated as an independent evidence, which are then incorporated into a hierarchical region merging process adapted from [19]. So, starting from superpixels, a consensus segmentation is obtained by fusing evidence accumulated from the other segmentations. This approach allows a more tractable analysis for each color and depth channel and has the potential of a straightforward extension to multi-modal segmentation, independently considering information from different sources.

Experiments on the public NYUD2 dataset [27] show the effectiveness of the proposed methodology. In particular, standard benchmark metrics are considered to quantify the segmentation results, obtaining better performance with respect to classic color segmentation. In such way, the present work is intended to contribute with more robust segmentation of indoor scenes to relevant applications such as machine vision [4] and object recognition [5].

Chapter 1
FUNDAMENTALS AND PREVIOUS WORK

This chapter presents important background material regarding image segmentation, superpixels and RGB-D images. Previous work on these topics is reviewed. Selected metrics for comparing segmentation results against human ground-truth are also defined.

1.1 Image Segmentation

Image segmentation is the process of partitioning a digital image into regions, called segments, for defined purposes of further image analysis [32]. Mathematically, let Ω represent the entire region occupied by an image. Then, Ω is partitioned into a finite number of segments S_i, $i = 1, ..., n$, such that [33]:

(a) $\bigcup_{i=1}^{n} S_i = \Omega$.

(b) S_i is a connected set, $i = 1, 2, ..., n$.

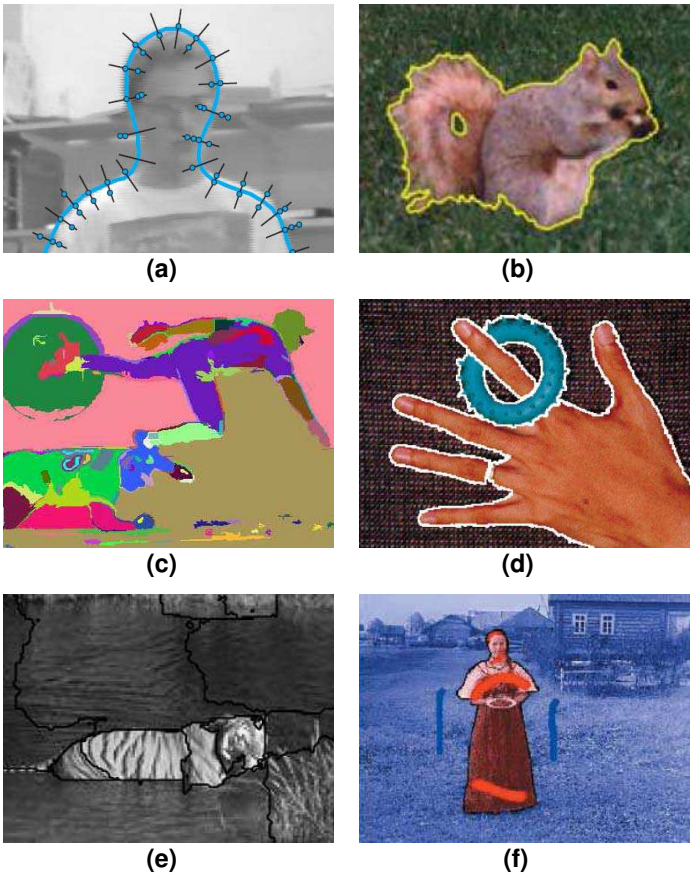
(c) $S_i \cap S_j = \emptyset$ for all i and j, $i \neq j$.

(d) $Q(S_i) =$ TRUE for $i = 1, 2, ..., n$.

(e) $Q(S_i \cup S_j) =$ FALSE for any adjacent regions S_i and S_j.

Here, $Q(S_i)$ is a logical predicate aimed at measuring the pixel consistency in the set S_i. Condition (a) indicates that the segmentation must be complete; that is, every pixel must be in a segment. (b) requires points in a region to be connected in some predefined sense. (c) indicates that the segments must be disjoint. (d) deals with the properties that must be satisfied by the pixels in a segmented region, therefore, all pixels share

Figure 2. **Some popular image segmentation techniques.** **(a)** Active contours [34]. **(b)** Level sets [35]. **(c)** Graph-based grouping [8]. **(d)** Mean shift [13]. **(e)** Normalized cuts [9]. **(f)** Binary MRF solved using graph cuts [36].



Taken from [37].

same characteristics (e.g., color). Finally, (e) indicates that two adjacent regions must be different in the sense of Q.

Fig. 2 shows some examples of segmentation techniques applied to different images. The segments are determined by analyzing similarities and discontinuities in different visual cues such as image edges, lines, color and texture. Other approaches also exploit additional features like focus level [2] and depth [1]. Thus, by identifying segments is then possible to describe the contents of an image, e.g., a segment for background, segments for objects or specific regions and even segments that represent people. The level of detail or number of segments depends on the problem being solved. That is,

segmentation should stop when the objects or regions of interest in an application have been detected [33].

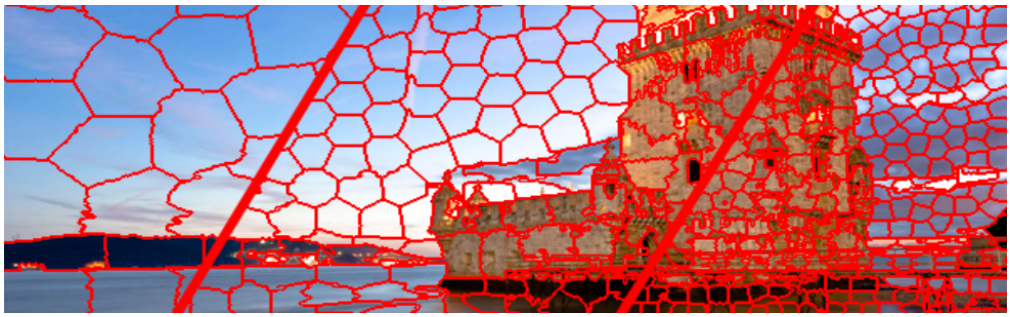
A rich amount of literature on image segmentation has been published over the past few decades. Many methods have achieved extraordinary success and became popular in a wide range of applications like medical imaging [3], object recognition [5], machine vision [4], surveillance [37] and so on. Among all the content reviewed (see References section), three families or categories are found as the most relevant: graph-based, mode seeking and region merging. Other remarkable approaches are based in fitting mixture models [38,39], active contours [34], color histograms [40,41], level sets [35] and various transformations [42,43].

Graph-based methods [7–12] generally represent the problem in terms of a graph $G = (V, E)$, where each node $v_i \in V$ corresponds to a pixel in the image, and an edge $(v_i, v_j) \in E$ connects nodes v_i and v_j. A weight is associated with each edge based on some property of the pixels that it connects, e.g., color [8]. Spectral clustering [44] is often used to partition the graph into a certain number of sub-graphs, which represent the segments. However, many graph-based methods convert the image segmentation into an optimization framework, while most of them are NP-hard to solve. Researchers often try to find alternative solutions to approximate the original problems, and some of them might result in unpredictable performance [7] or break large uniform segments into several pieces.

Mode seeking methods [13–16] are applied by clustering data $\{(x, f(x)), x \in \Omega\}$, where $x \in \Omega$ are the image pixels and $f(x)$ their feature coordinates, e.g., color and position [14]. In general, this task starts from the detection of local density maxima (or modes) in the feature space, followed by their partition into several clusters (segments). Each mode corresponds to a cluster centroid. However, these methods tend to produce superpixels (over-segmentation) and are often used as a pre-processing step for other segmentation schemes [19].

Region merging methods [17–20] are able to produce a hierarchy of segmentation, which can be typically represented as a tree with each level corresponding to a specific segmentation. For doing so, an over-segmentation process is performed. Then, adjacent regions are merged based on their relative boundary strength or inter-similarity mea-

Figure 3. **Superpixel segmentation**. Example of superpixels with multiple-scaled sizes.



Taken from [46].

sures. The final segmentation with a certain number of segments can be obtained by specifying a level (or scale) for the segmentation tree [19].

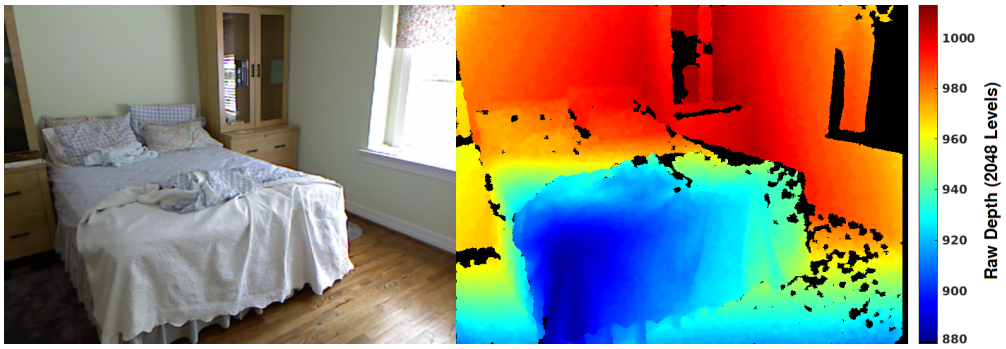
1.2 Superpixels

The term superpixel was introduced by Ren and Malik in 2003 [45] and has received increasing attention in the last years. In general, a superpixel refers to a compact region that is produced in an over-segmentation, where object details, boundaries and pixel-wise relationships are expected to be well preserved [31]. Fig. 3 illustrates an example of superpixel segmentation at different sizes.

Superpixels allow to represent an image with a set of tiny segments instead of a large number of pixels. For this reason, superpixels have been used to reduce computational complexity in several computer vision tasks. While many algorithms such as [31, 47–50] are explicitly designed to generate superpixel segmentations, others that were initially intended for classical segmentation often fall into this category due to the over-segmentation problem that affects them. Some graph-based [8, 9] and mode seeking [13, 14] methods are found to be examples of this, so, in the present work, these will be treated as superpixel methods.

Over-segmenting into superpixels is usually not preferred for a coherent segmentation, hence, superpixels need to be post-processed in order to achieve a final segmentation [31]. In such way, researchers have taken advantage of many superpixel algorithms as

Figure 4. **RGB-D image.** ***Left:*** Color image. ***Right:*** Raw depth map. Missing depth values are represented in black color.



a pre-processing step for their segmentation frameworks [2, 11, 12, 19, 45]. The present work also relies on a superpixel stage.

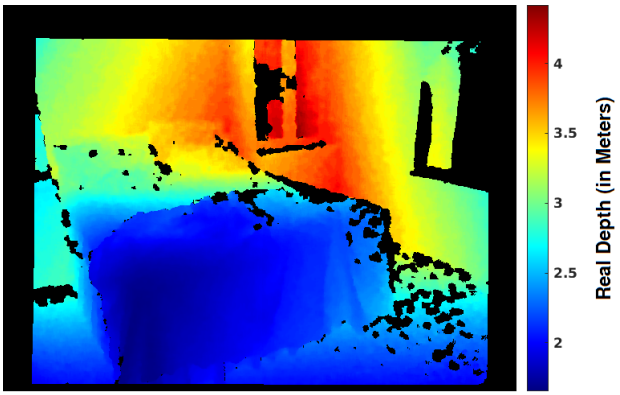
1.3 RGB-D Images

Recent advances on depth sensing technologies make the acquisition of depth information (depth map) much easier than before. New devices like the Microsoft Kinect[1], the Asus Xtion[2]and the Intel RealSense[3]are able to provide color image plus depth map, commonly referred to as RGB-D images. Naturally, it's more convenient to create RGB-D images to give a more comprehensive description of the captured scene. This interesting feature is becoming more and more popular in computer vision, as it is being succesfully used in several applications: object tracking and recognition [51,52], human activity and gesture analysis [53,54], indoor 3D mapping [55], scene understanding [56], and many others [21]. Most of these tasks imply some kind of segmentation, which is the aim of this work.

Fig. 4 shows a sample RGB-D image from the NYUD2 dataset [27] (obtained through the Microsoft Kinect). The depth map can be viewed as an image $\{x_i, y_i, d_i\}$, where x_i, y_i are the pixel coordinates of a point i in the scene, and d_i is a level that quantifies the distance of that point to the sensor. For the Kinect case, there are a total of 2048 levels of sensitivity, i.e., an 11-bit depth map [57].

[1] developer.microsoft.com/en-us/windows/kinect
[2] www.asus.com/3D-Sensor/Xtion_PRO/
[3] software.intel.com/en-us/realsense/

Figure 5. **Aligned depth map.** Due to the geometric transformations that take place in the registration process, many border pixels result without the presence of depth measures.



However, many points in a scene may have no depth due to multiple reflections, transparent objects, scattering in particular surfaces or occlusions. Furthermore, a pixel in the RGB image refers to a different point of the same pixel in the depth map. This is because of the internal device composition, which is actually two separate cameras: one for color and one for depth. Then, the slight difference of camera positions (e.g., 2.5cm for the Kinect [58]) causes that both images are not exactly in the same viewpoint. Thus, to spatially align them, it is necessary to perform a registration process. Additionally, the obtained raw depth values must be converted to real depth units (e.g., meters). Each manufacturer provides its own software to do these pre-processing operations, and optionally, some open-source tools like openNI[4]and libfreenect[5]are available.

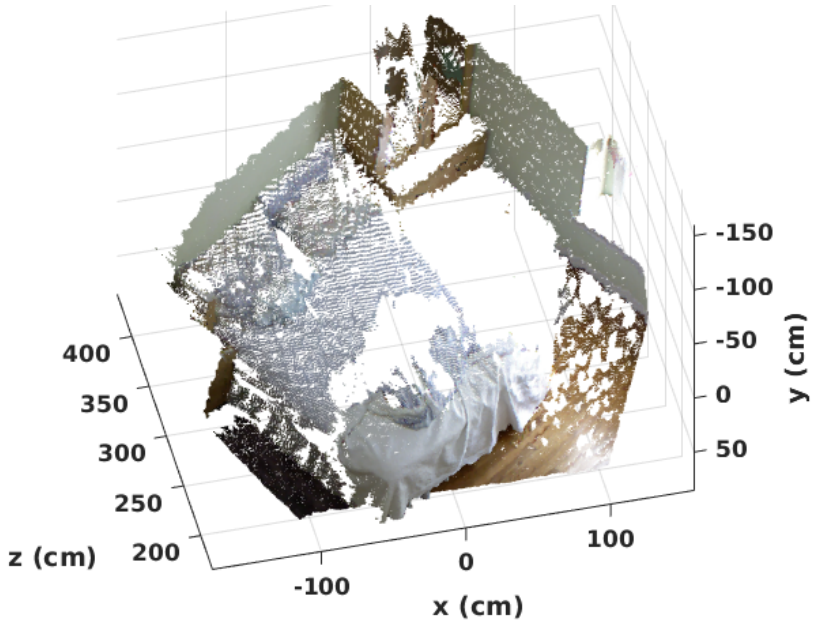
Once the depth map is pre-processed, it looks like in Fig. 5. Here, many edges and contours are consistent with the color image, allowing that depth patterns and discontinuities can be fully exploited to differentiate image regions and objects. With that in mind, it turns out that a 3D model of the captured scene can also be generated, as described below.

1.3.1 3D Geometry Reconstruction.

Given the intrinsic parameters of the depth camera, the 3D geometry of the captured scene can be reconstructed from the depth map [58]. Denoted by (x, y, d) the pixel locations and their corresponding

[4] wiki.ros.org/openni_launch/Tutorials
[5] github.com/OpenKinect/libfreenect

Figure 6. **3D point cloud.** A collection of points defined in the XYZ coordinates intended to represent the 3D geometry of the captured scene.



depth measures, the 3D pixel coordinates (X, Y, Z) (relative to the camera's view-point) are obtained as [30]:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & 0 & x_0 \\ 0 & f_y & 0 & y_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix}, \tag{1.1}$$

where f_x and f_y are the focal lengths of the x and y axes respectively, γ is the skew ratio between these two axes, and (x_0, y_0) are the principal point coordinates, as defined in the *pinhole* camera model [59].

The set of all points $\{X_i, Y_i, Z_i\}$ are called 3D point cloud. Fig. 6 illustrates the point cloud associated to the depth map in Fig. 5. This step is fundamental in many applications that use RGB-D images.

Regarding image segmentation, new methods that use RGB-D images have incorporated this 3D information to their frameworks. Hence, in the next subsection, representative RGB-D segmentation approaches will be reviewed.

1.3.2 RGB-D Image Segmentation.

Depth information provided by RGB-D cameras is very attractive to perform segmentation more accurately. Many ambiguities in the mere RGB data can be greatly reduced by knowing the 3D position of scene entities. In particular, segmentation of indoor scenes (due to the depth range limitations) using RGB-D images is a topic that has been recently studied in the state-of-the-art. For instance, Silberman *et al.* [27] modify the algorithm of [60] to use depth information for over-segmenting the image and then merge superpixels based on similarity levels, which are obtained by learned classifiers over RGB, depth, and inferred structure data. They also consider the task of *semantic segmentation*, i.e., to assign a category label (limited) to each region for a semantic interpretation of the scene. And, additionally, they provide a dataset of 1449 RGB-D images capturing 464 diverse indoor scenes (this is the NYUD2 dataset). Ren *et al.* [29] use *kernel descriptors* to capture a variety of color and depth features on different over-segmentations, followed by a Markov Random Field (MRF) context model. Gupta *et al.* [1] generalize the hierarchical segmentation approach of [18] by combining color, texture and 3D gradients in different scales. They also perform semantic segmentation by classifying regions into 40 dominant object categories of the NYUD2 dataset. Other schemes like [28, 30] propose strategies for RGB-D data clustering and then a globally optimal segmentation is achieved using graph theory.

1.4 Performance Measures

Taking as reference the works presented in [11, 18, 19], four standard metrics have been selected to quantitatively evaluate the segmentation results against human ground-truth: Segmentation Covering [61], Rand Index [62], Variation of Information [63], and Boundary Displacement Error [64].

1.4.1 Segmentation Covering.

The *overlap* between two regions R and R', defined as:

$$\mathcal{O}(R, R') = \frac{|R \cap R'|}{|R \cup R'|}, \tag{1.2}$$

has been widely used for comparing the similarity of segmented regions with respect to the ground-truth labels.

As defined in [61], the *covering* of a test segmentation S by a ground-truth segmentation

G is:

$$\mathcal{C}(S,G) = \frac{1}{N}\sum_{R\in S} |R| \cdot \max_{R'\in G} \mathcal{O}(R,R'), \tag{1.3}$$

where N denotes the total number of pixels in the image.
A value of 1 indicates perfect covering, therefore, a segmentation is viewed as better as $\mathcal{C}$ approaches 1.

1.4.2 Rand Index. Consider the two segmentations S and G of N pixels $\{x_1, x_2, ..., x_N\}$ that assign labels $\{l_i\}$ and $\{l'_i\}$, respectively, to a pixel x_i. The Rand Index RI can be computed as the ratio of the number of pixel pairs having the same label relationship in S and G [62], i.e.,

$$RI(S,G) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i\neq j}} \left[\mathbb{I}\big(l_i = l_j \wedge l'_i = l'_j\big) + \mathbb{I}\big(l_i \neq l_j \wedge l'_i \neq l'_j\big) \right], \tag{1.4}$$

where $\mathbb{I}$ is the identity function and $\binom{N}{2}$ is the number of possible unique pairs among N pixels.
This gives a measure that quantifies the fraction of pixel pairs whose labels are consistent between S and G. And, as in the case of (1.3), its maximun value is 1, when the two segmentations are actually the same.

1.4.3 Variation of Information. The Variation of Information measures the distance between two clusterings of data in terms of the information difference between them. Formally, it is defined as [63]:

$$VI(S,G) = H(S) + H(G) - 2I(G,S), \tag{1.5}$$

where H and I represent respectively the entropies and mutual information between the two clusterings, which in our case are the test segmentation S and the ground-truth segmentation G.
For this metric, unlike (1.3) and (1.4), values close to zero indicate greater similarity, evidencing high quality of segmentation.

1.4.4 Boundary Displacement Error. The Boundary Displacement Error (BDE) is intended to evaluate segmentation quality in terms of the precision of the extracted region boundaries. To do this, it defines two quantities [64]: the distance

of a boundary pixel p_i in a test segmentation S to the closest boundary pixel in the ground-truth segmentation G, and, conversely, the distance in a boundary pixel p_j in G to the closest boundary pixel in S, respectively denoted by:

$$d(p_i, G) = \min_{\substack{p_i \in S \\ p \in G}} ||p_i - p|| \tag{1.6}$$

$$d(p_j, S) = \min_{\substack{p_j \in G \\ p \in S}} ||p_j - p|| \tag{1.7}$$

Hence, the BDE is defined as follows:

$$BDE(S, G) = \frac{1}{2}\left(\frac{1}{N_1}\sum_{i}^{N_1} d(p_i, G) \; + \; \frac{1}{N_2}\sum_{j}^{N_2} d(p_j, S)\right), \tag{1.8}$$

where N_1 and N_2 are the total number of points in the boundary sets of S and G.
To the extent that the BDE is smaller, a segmentation shows greater similarity with respect to the ground-truth.

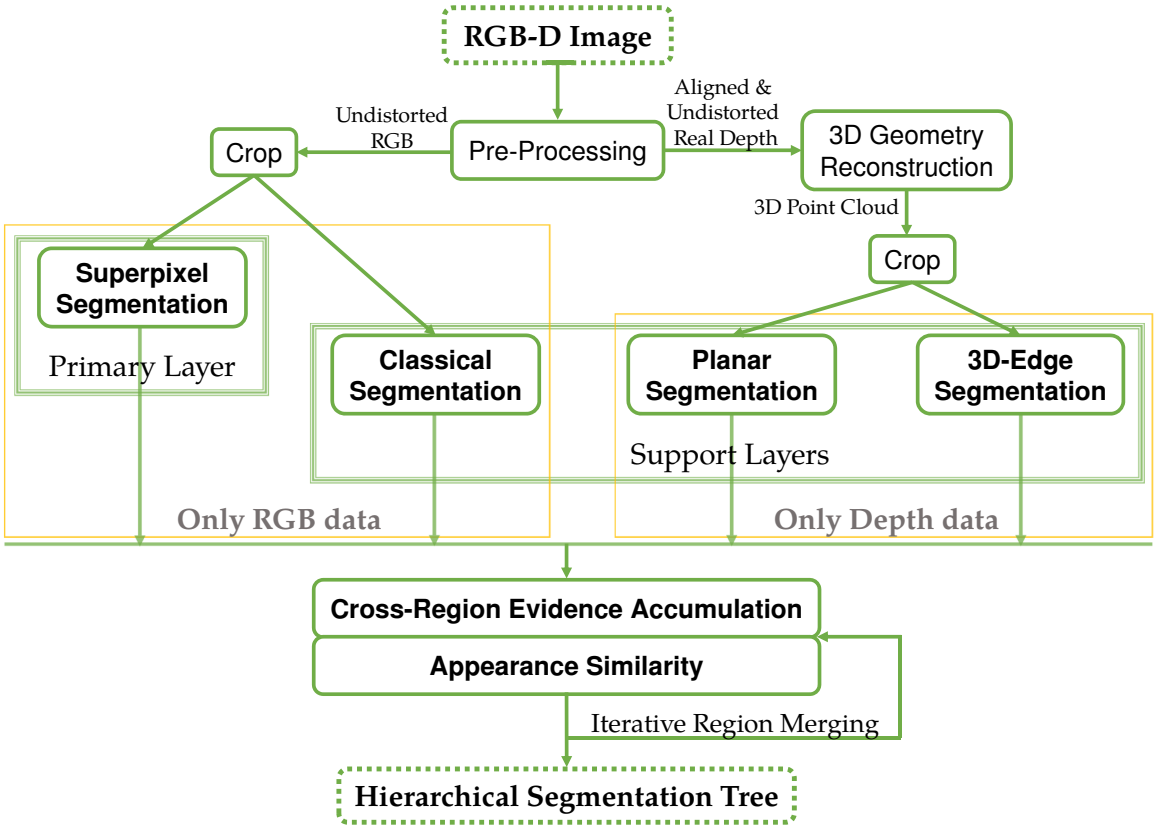
Chapter 2
METHODOLOGY

In this work, a methodology for RGB-D image segmentation is proposed. The principal goal is to improve difficulties of classic techniques that only use RGB information. A key difference with respect to state-of-the-art approaches is the analysis of color and depth data in an independent manner. For doing so, similar to consensus clustering algorithms [65] which aim to combine a set of different clusterings to find a more robust and better one, different segmentations performed on both color and 3D space are exploited to obtain a final segmentation. A high-level overview of the entire process is shown in Fig. 7. Here, the key aspect is the generation of the next four segmentations:

- **Using RGB data:**
    1. Superpixel segmentation
    2. Classical segmentation
- **Using Depth data:**
    3. Planar segmentation
    4. 3D-edge segmentation

The superpixel segmentation is treated as the *primary layer*, upon which a region merging process will be carried out. The other segmentations are the corresponding *support layers*, which will provide different evidences to calculate similarity measures between adjacent superpixels. In such way, the proposed methodology defines three main stages to obtain an RGB-D segmentation: RGB-D Image Pre-Processing, Segmentation Layers Generation, and finally, Hierarchical Region Merging.

Figure 7. **Methodology outline.** The RGB-D image is pre-processed and then passed to different segmentations schemes on both color and 3D space. The information in these set of segmentations, together with appearance cues, are incorporated in a hierarchical region merging process to construct a segmentation tree. The final segmentation can be obtained by choosing a scale (or level) in the tree.



2.1 RGB-D Image Pre-Processing

RGB-D images obtained through RGB-D cameras normally cannot be directly fed into computer vision algorithms, since the RGB and depth channels are not properly aligned. Besides that, the obtained depth measures are not in length units. Therefore, in order to correct this behavior, it is necessary to do a few pre-processing steps using camera calibration parameters [58]. Lens distortion is also considered to give more accurate results. Subsequently, for a richer understanding of the captured scene, its 3D geometry is reconstructed as indicated in section 1.3.1. As last step, color image and 3D point cloud are cropped to remove border pixels in which no depth measures are present, as highlighted in Fig. 5.

2.2 Segmentation Layers Generation

Details about the methods and tools that allow the generation of each segmentation layer will be discussed below.

2.2.1 Superpixel Segmentation. Superpixels capture image redundancy and provide a more convenient representation from which to compute image features [31]. Generally, over-segmenting into superpixels is an easier task than obtaining a "good" segmentation. For this reason, they have been exploited as an initial step in several segmentation methods [11, 12, 31, 45]. Inspired by such approaches, this work makes use of a superpixel segmentation as the methodology starting point. This is its role as the primary layer, from which a final segmentation will be obtained.

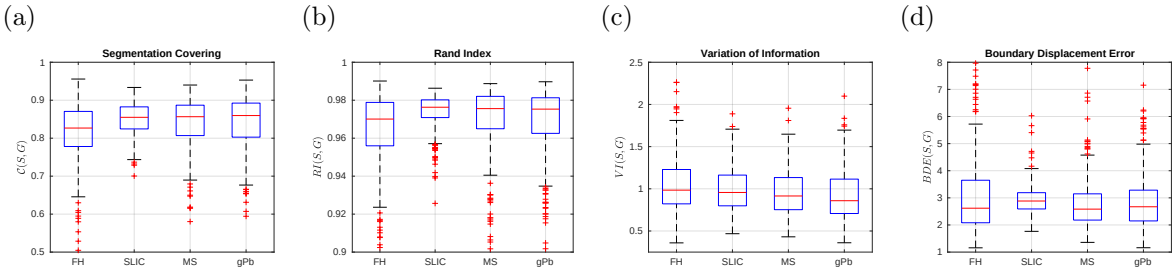
Many good superpixel algorithms exist, however, four of them are found to be the most widely used in the state-of-the-art: graph-based superpixels by Felzenszwalb and Huttenlocher [8], SLIC superpixels by Achanta *et al.* [31], mean-shift superpixels by Comaniciu and Meer [13], and *gPb*-OWT-UCM superpixels by Arbeláez *et al.* [18]. These are termed FH, SLIC, MS, and gPb, respectively. For experimental purposes, one of them will be selected as the methodology primary layer based on the next comparative process.

2.2.1.1 Comparative Analysis of Superpixel Algorithms. To evaluate the performance of superpixel algorithms regarding to their qualities to guide segmentation, the following pipeline is proposed:

1. Divide the image into P superpixels $\{\mathcal{S}_p | p = 1, 2, ..., P\}$, labeled with P labels.
2. Load the ground-truth segmentation, composed by N regions $\{\mathcal{R}_n | n = 1, 2, ..., N\}$ labeled with N labels.
3. Relabel each generated superpixel with the label of its corresponding region in the ground-truth. Specifically, a superpixel $\mathcal{S}_p$ is relabeled with the $\tilde{n}$-th ground-truth label that maximizes the region intersection:

$$\tilde{n} = \underset{n}{\operatorname{argmax}}\ \mathcal{I}(n), \tag{2.1}$$

Figure 8. **Performance comparison of FH, SLIC, MS, and gPb superpixels. (a)** Segmentation covering ($\mathcal{C}$). **(b)** Rand index (RI). **(c)** Variation of information (VI). **(d)** Boundary displacement error (BDE). For $\mathcal{C}$ and RI, higher values indicate better segmentation; for VI and BDE lower values indicate better segmentation.



where,

$$\mathcal{I}(n) = |\mathcal{S}_p \cap \mathcal{R}_n| \tag{2.2}$$

Basically, this step merges superpixels to get the closest segmentation to the ground-truth.

4. Compute performance measures for the resulting segmentation.

The evaluation was performed on 290 images randomly selected from the NYUD2 dataset [27]. Parameters for each algorithm were tunned to yield approximately 400 superpixels, being this a suitable representation for the size of NYUD2 images (640x480 pixels) determined according to preliminary experiments. Box plots of obtained results are presented in Fig. 8. Visual segmentation results for three sample images are given in Fig. 9.

From Fig. 8, it can be concluded that FH superpixels yield the most dispersed results. In terms of $\mathcal{C}$, RI and VI, its performance is slightly the lowest. SLIC superpixels show the smallest IQR. It seems competitive in terms of $\mathcal{C}$ and RI, but regarding VI and BDE is less performed than MS and gPb. In summary, MS and gPb can be considered relatively better than FH and SLIC.

In Fig. 9, from left to right, the sample images are referred to as image 1, 2 and 3. Detailing the obtained segmentations in image 1 and 2, it can be observed that the gPb segmentation presents the least noisy contour lines (in green). So, it seems to be

Figure 9. **Sample images and their segmentations by FH, SLIC, MS, and gPb superpixels.** Row 1 corresponds to the original images and their ground-truths. Rows 2, 3, 4 and 5 correspond to the superpixel segmentation and the resulting segmentation by means of equation (2.1) for FH, SLIC, MS, and gPb superpixels, respectively.

the closest one to the ground-truth. For image 3, the background wall is confused with the left wall in FH and MS segmentations. This is indeed notable in the superpixel segmentations, where a single superpixel covers both regions, clearly because of their high color similarity (both in white). Generally, more cases alike were presented in the MS segmentation. Therefore, it is concluded that the gPb segmentation is slightly more accurate and visually nicer than the others.

From such analysis, the gPb algorithm is then selected to generate the methodology primary layer in all the later experiments.

2.2.2 Classical Segmentation.

Classical segmentation differs from superpixel segmentation in the sense that it is intended to achieve a final partition of the image of interest. In terms of segmented regions, it is expected that classical segmentation produces a smaller number of regions, aiming to a clearer and simpler represen-

tation of the captured scene. However, this segmentation can present some problems (as mentioned in previous sections) that the proposed methodology aims to improve by using RGB-D images.

It is worth mentioning that the previously selected gPb algorithm, besides being used to generate superpixels, is also a popular alternative for classical segmentation due to its output being a hierarchical segmentation tree [18]. From this hierarchy, multiple segmentations can be obtained by varying its single scale parameter. That is, the lowest scale is an over-segmentation (superpixels) and the highest scale in an under-segmentation. Thus, the gPb algorithm is also adopted to obtain the classical segmentation, allowing one single process to generate the two RGB layers.

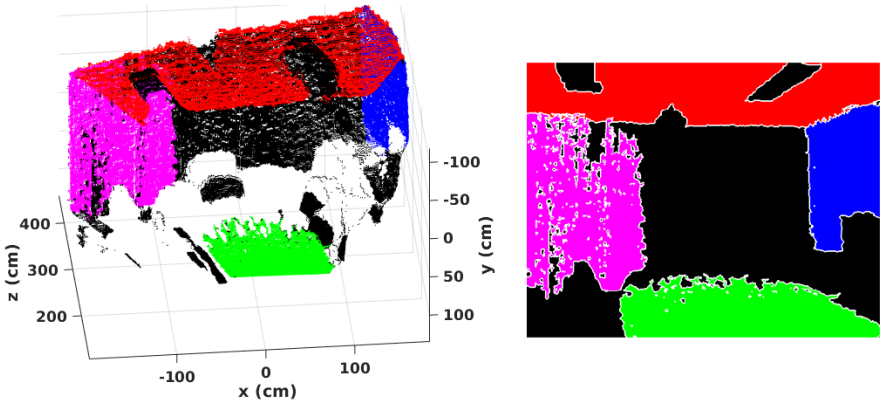
2.2.3 Planar Segmentation. Plane extraction in 3D point clouds is crucial to get relevant primitives for image analysis. For example, in the case of indoor scenes, many structures mainly consist of planar surfaces. In this work, a planar segmentation is obtained by using the plane detection method of Feng *et al.* [24], termed PAHC. This selection is due to its proven efficiency, publicly available implementation and no parameters to tune.

PAHC takes as input the 3D point cloud $\{X_i, Y_i, Z_i\}$ and returns the set of planar regions $\{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_K\}$, where each $\mathcal{P}_k$ contains the indices i of the XYZ points belonging to the k-th extracted plane. These indices are then mapped to the image for obtaining the planar segmentation. A sample output is illustrated in Fig. 10 for the left test image of Fig. 9.

In essence, to obtain each planar region $\mathcal{P}_k$, the PAHC algorithm first constructs a graph by dividing the point cloud into several regions with a uniform size. Subsequently, an agglomerative hierarchical clustering (AHC) is performed on the graph repeating the next steps:

1. Find the region that has the minimum plane fitting mean squared error (MSE).
2. Merging that region with one of its neighbors such that the merge results in the minimum plane fitting MSE.
3. Stop when the plane fitting MSE exceeds a threshold.

Figure 10. **Plane detection.** ***Left:*** Detected planes in 3D point cloud. ***Right:*** Planar segmentation. Black means no plane detected.



2.2.4 3D-Edge Segmentation.

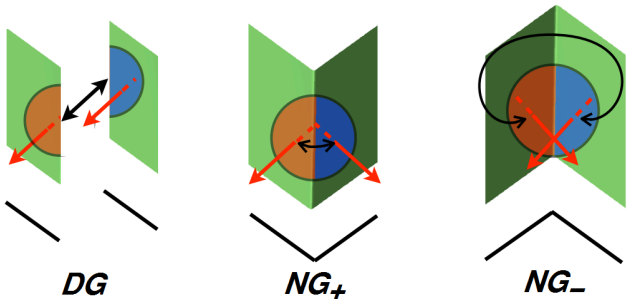
Edges are powerful image features that can mark clear distinction between different regions. In consequence, edge detection is one of the most important steps for image segmentation [66]. Edges can be detected in RGB images from its gradient responses, therefore, with the addition of depth information and the corresponding 3D point cloud, the possibility to compute 3D gradients arises.

In 3D point clouds, edges can be manifested in terms of depth discontinuities and changes in surface orientations. With this premise, Gupta *et al.* [1], as an early attempt to do 3D-edge detection, proposed the next three contour signals:

- A depth gradient DG, which identifies the presence of depth discontinuities.
- A convex normal gradient NG_+, which captures if the surface bends-out at a given point in a given direction.
- A concave normal gradient NG_-, capturing if the surface bends-in.

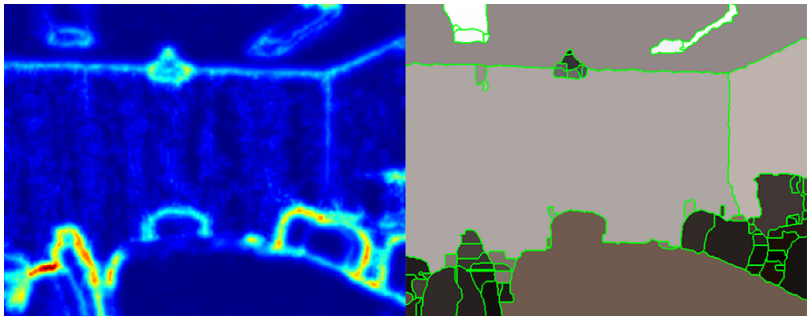
To compute them, they consider a disk centered at each XYZ point. The disk is split into two halves at a pre-defined orientation and the information in each half is compared as first suggested in [67] for the case of RGB data. Specifically, each half-disk is represented by a planar model, then, two measures are calculated: the distance between the two planes for the case of DG and the angle between the plane normals for NG_+ and NG_-. Fig. 11 presents a visual representation of these gradients.

Figure 11. **3D gradients representation.**



Adapted from [1].

Figure 12. **3D-edge segmentation.** ***Left:*** G_{3D} signal response. ***Right:*** Obtained segmentation from the G3D-OWT-UCM hierarchy by using a scale value of 0.35.



Now, in order to obtain a general contour signal, the present work takes the DG, NG_+ and NG_- gradients and adds them in one single signal G_{3D}. This is then processed using a sequence of two transformations: Oriented Watershed Transform (OWT) and Ultrametric Contour Map (UCM), a generic machinery for going from contours to a hierarchical segmentation tree, proposed by Arbeláez *et al.* in [61]. The resulting hierarchy is then termed G3D-OWT-UCM. So, for a given tree scale $K_{G3D} \in [0-1]$, a 3D-edge segmentation can be obtained. Fig. 12 shows the G_{3D} response and its respective segmentation ($K_{G3D} = 0.35$) for the same image considered in Fig. 10. From these two figures, the complementary nature of planar and 3D-edge segmentations can be appreciated.

2.3 Hierarchical Region Merging

Huang *et al.* [19] proposed a hierarchical region merging process for exploiting multiple over-segmentations in RGB data. The core of its contribution is the Cross-Region Evidence Accumulation (CREA) mechanism to fuse all the available information by means of a regional voting strategy. In this work, the approach of Huang *et al.* is adapted and implemented for the case of the previously discussed segmentations (primary and support layers). In essence, the process is based on the accumulation of evidence from the support layers by the CREA mechanism. This information, supported with measures of appearance similarity, is used to build a hierarchical segmentation tree. Then, a final segmentation is obtained by choosing an appropriate scale in the tree.

The region merging process starts from the primary layer (superpixels). Each pair of adjacent superpixels are analyzed to determine their coherency as a single entity. Formally, let's denote by L_p the primary layer and by $L_s^k = \{L_s^1, L_s^2, L_s^3\}$ the support layers. Each layer consists of a set of regions (either superpixels or common segments), that is

$$L_p = \{R_1, R_2, ..., R_{n_p}\} \tag{2.3}$$

$$L_s^k = \{R_1^k, R_2^k, ..., R_{n_k}^k\}, \tag{2.4}$$

where R_i, n_p are respectively the i-th region and the total number of regions in L_p; and R_i^k, n_k are respectively the i-th region and the total number of regions in L_s^k.

Over the regions in L_p, the region merging algorithm proceeds according to the total similarity measure between adjacent regions, denominated *joint similarity* and defined as:

$$Sj(R_i, R_j) = (1 - \lambda_a) S_{crea}(R_i, R_j) + \lambda_a S_a(R_i, R_j), \tag{2.5}$$

where $S_{crea}(R_i, R_j)$ and $S_a(R_i, R_j)$ are the similarity obtained by the CREA mechanism and the appearance comparison, respectively. The parameter $\lambda_a \in [0 - 1]$ defines the weight of the appearance similarity. Sj ranges from 0 (very different regions) to 1 (very similar regions).

The CREA mechanism works via a regional voting strategy with the information from

support layers. The appearance similarity integrates brightness and color cues from RGB data. The calculation of S_{crea} and S_a will be described in the subsections 2.3.1 and 2.3.2.

Having computed Sj between all superpixels in L_p, the region merging is carried out in an iterative manner, i.e., in each iteration, the two regions with the highest Sj, namely Sj_{MAX}, will be merged into a new and larger region. Then, Sj is updated for that region and each of its adjacent regions reusing equation (2.5). The next iteration repeats the same operation, and so on. At the beginning, Sj_{MAX} values will be high due to the presence of many regions that possibly are part of one single entity. Later, when the most similar regions become merged, its values will begin to decrease, indicating that remainder regions are not similar. In such way, the iterative merging over adjacent regions in L_p leads to a hierarchical segmentation tree, where lowest scales (high Sj_{MAX} values) are over-segmentations and the highest scales (low Sj_{MAX} values) are under-segmentations. Thus, to select an appropriate scale for coherent segmentation, the present work defines the parameter Sj_{THR} as a threshold for the Sj_{MAX} values associated to each tree level.

2.3.1 Cross-Region Evidence Accumulation.

To accumulate regional evidence among the supporting segmentations, the CREA mechanism is based on the intuition that the more frequently two regions occur in the same segment among different segmentations, the more likely it is that they belong to the same entity.

A region consists of a certain number of pixels. If the pixels of two regions all occur in the same segment of another segmentation, then this is an evidence that the two regions may belong to a coherent region. In another case, if only part of the pixels occur in the same segment, this is also an evidence of coherency, however, its strength will be different and should be related to the number of occluding pixels and the sizes of the two regions. Therefore, the relationship between two regions $R_i, R_j \in L_p$ can be viewed as the relationship between two sets of pixels. Two pixels, one from R_i and one from R_j, are called a pixel-pair across R_i and R_j. There would be totally $|R_i| \cdot |R_j|$ pixel-pairs, each one acts as an independent voter.

Let R_h^k be a region in the k-th support layer L_s^k, i.e., $R_h^k \in L_s^k$. If a pixel-pair occurs in the region R_h^k, then this voter supports the coherency of R_i and R_j w.r.t. R_h^k. So,

by considering the occluding portions between R_i, R_j and R_h^k, the ratio of voters that support the coherency of R_i and R_j w.r.t. R_h^k is:

$$\text{vote}_h^k(R_i, R_j) = \frac{|R_i| \cap |R_h^k| \cdot |R_j| \cap |R_h^k|}{|R_i| \cdot |R_j|} \tag{2.6}$$

When all pixels in R_i and R_j appear in R_h^k, $\text{vote}_h^k(R_i, R_j)$ reaches its maximum value, i.e., 1. It is possible that the voters across R_i and R_j may occur in more than one region in L_s^k. Specifically, some voters across R_i and R_j may support their coherency w.r.t. R_h^k, while others may support their coherency w.r.t. R_g^k ($g \neq h$). To obtain the ratio of voters that support the coherency of R_i and R_j w.r.t. to L_s^k, the votes are collected for R_i and R_j w.r.t. different regions in L_s^k. That is

$$\text{vote}^k(R_i, R_j) = \sum_{R_h^k \in L_s^k} \text{vote}_h^k(R_i, R_j) \tag{2.7}$$

It holds that $\text{vote}^k(R_i, R_j) \in [0 - 1]$. The ratio of voters that support the coherency of two regions w.r.t. one of the support layers can be viewed as a measure of similarity for these regions in terms of the information of that layer. Thus, the total similarity given the three support layers is computed as:

$$S_{crea}(R_i, R_j) = \frac{1}{3} \sum_{k=1}^{3} \text{vote}^k(R_i, R_j) \tag{2.8}$$

2.3.2 Appearance Similarity.

Huang *et al.* [19], besides the CREA mechanism, integrates information of brightness, color, and texture cues in their framework. In this work, brightness and color cues are considered to to see how far it can complement the information of support layers.

The RGB image is converted into the CIE-Lab color space, where the L channel corresponds to the brightness and the a,b channels correspond to the color. Over each channel, a histogram for each region in the primary layer is constructed. In the state-of-the-art, it is common to measure the dissimilarity between two normalized histograms

h_i, h_j by the chi-square distance χ^2, defined as:

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{k=1}^{N_h} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}, \tag{2.9}$$

where N_h is the number of bins of the histograms h_i, h_j (must be equal). $\chi^2(h_i, h_j)$ ranges from 0 to 1.

So, by using the χ^2 distance, the similarity between two regions R_i, R_j w.r.t each channel can be expressed as:

$$S^L(R_i, R_j) = 1 - \chi^2(h_i^L, h_j^L) \tag{2.10}$$

$$S^a(R_i, R_j) = 1 - \chi^2(h_i^a, h_j^a) \tag{2.11}$$

$$S^b(R_i, R_j) = 1 - \chi^2(h_i^b, h_j^b) \tag{2.12}$$

where h^L, h^a, h^b are the histograms in the L, a, and b channels, respectively.

Finally, the appearance similarity for R_i, R_j is computed as:

$$S_a(R_i, R_j) = \frac{1}{3} \left[S^L(R_i, R_j) + S^a(R_i, R_j) + S^b(R_i, R_j) \right] \tag{2.13}$$

Chapter 3
EXPERIMENTS AND RESULTS

In this chapter, the proposed methodology is evaluated on the NYUD2 dataset [27] and compared with representative state-of-the-art algorithms. The experiments were conducted in MATLAB R2016b 64-bits (Ubuntu Linux) on a workstation with Intel Core i7 CPU (4 cores) and 32 GB of RAM.

3.1 Dataset

The NYUD2 dataset contains 1449 RGB-D images with their corresponding ground-truths. The images show diverse indoor scenes of private apartments and commercial accommodations. In this work, in order to determine adequate values for the methodology parameters, the dataset is split into training and test sets. The training set is the 20% of the total dataset, that is, 290 randomly selected images. The other 80% is the test set, i.e., 1159 images.

3.2 Parameter Adjustment

Firstly, the K_{G3D} parameter of the G3D-OWT-UCM hierarchy (see section 2.2.4) is adjusted for producing a coherent 3D-edge segmentation. This is done by comparing the performance of obtained 3D-edge segmentations over all the training set with different K_{G3D} values. The segmentation covering is selected for this comparison due to its broad usage in the literature. Fig. 13 shows the average results for $K_{G3D} \in [0.1 - 0.55]$. It can be observed that a value of 0.35 yields the best performance. Hence, $K_{G3D} = 0.35$ is selected for subsequent experimentation.

Figure 13. **Influence of K_{G3D} parameter.** Average performance for the 3D-edge segmentation by varying K_{G3D}.

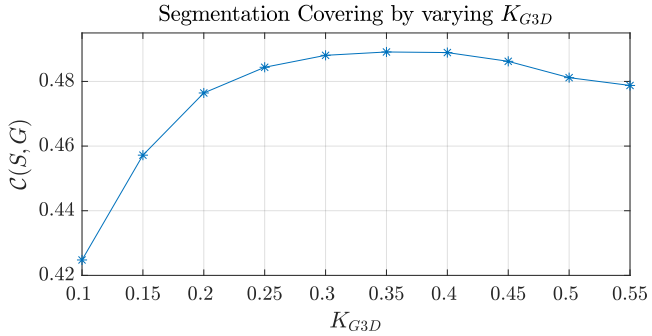
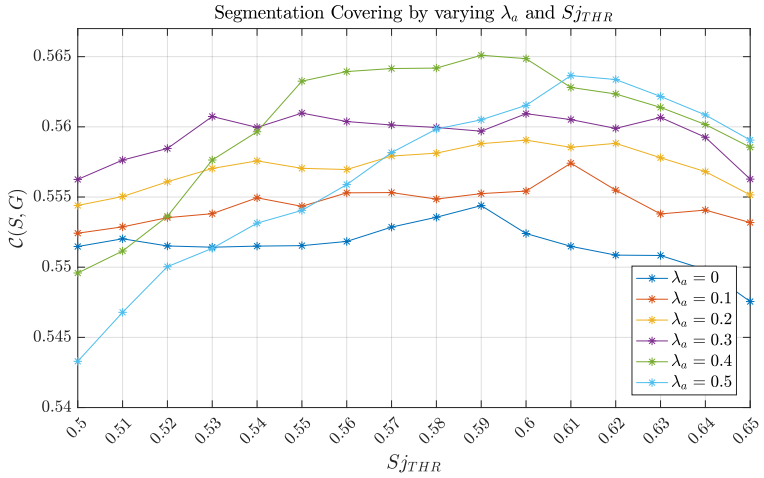
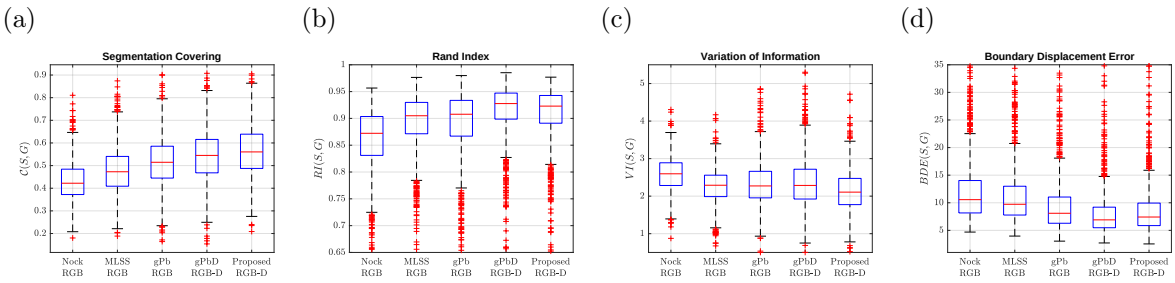


Figure 14. **Influence of λ_a and Sj_{THR} parameters.** Average performance for the proposed segmentation by varying λ_a and Sj_{THR}.



Now, the λ_a and Sj_{THR} parameters of the region merging process (see section 2.3) will be analyzed. All the methodology pipeline (see Fig. 7) is run on the training set with different combinations of λ_a and Sj_{THR}. The performance obtained in terms of segmentation covering is presented in Fig. 14. From here, it can be concluded that the appearance similarity does indeed help to get better results than only using the CREA similarity. In fact, the lowest performance is obtained when $\lambda_a = 0$. Then, as λ_a is increased, the results improve, being $\lambda_a = 0.4$ an appropriate value with the best performance for $Sj_{THR} \in [0.55 - 0.6]$. This is a reasonable result, since low values of Sj_{THR} lead to under-segmentation and high values lead to over-segmentation. Therefore, the selected values are $\lambda_a = 0.4$ and $Sj_{THR} = 0.59$.

Figure 15. **Performance comparison between state-of-the-art and proposed segmentations.** **(a)** Segmentation covering ($\mathcal{C}$). **(b)** Rand index (RI). **(c)** Variation of information (VI). **(d)** Boundary displacement error (BDE). For $\mathcal{C}$ and RI, higher values indicate better segmentation; for VI and BDE lower values indicate better segmentation.

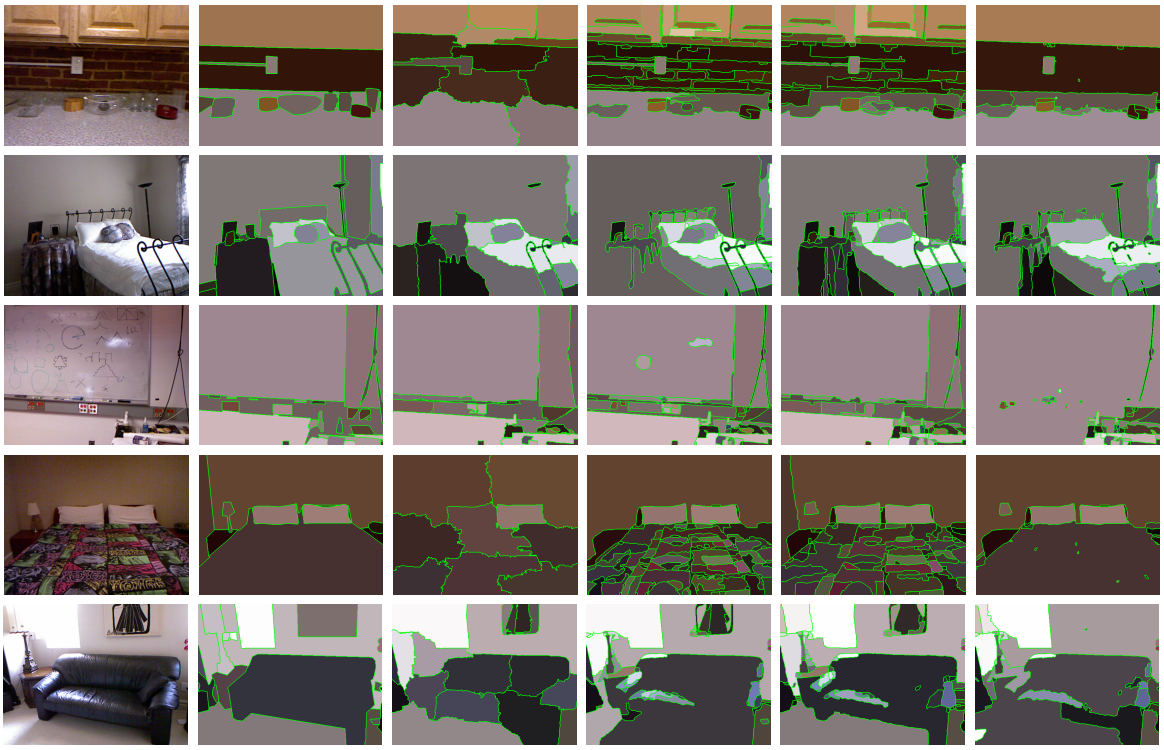


3.3 Quantitative and Qualitative Evaluation

With the previously selected parameters, the proposed methodology is run on the entire test set. Obtained segmentations are compared with four state-of-the-art segmentations: statistical region merging by Nock and Nielsen [17], full pairwise affinities for spectral segmentation by Kim *et al.* [12], *gPb*-OWT-UCM by Arbeláez *et al.* [18] and the RGB-D segmentation proposed by Gupta *et al.* [1]. These are termed Nock, MLSS, gPb, and gPbD, respectively. Box plots of performance results are presented in Fig. 15. Visual segmentation results for five sample images are given in Fig. 16. For Nock and gPb, their scale parameter is tuned based on segmentation covering scores. For MLSS the required parameter is the number of desired segments, so, the number of ground-truth segments is used. For gPbD the same parameters used by the authors in the same dataset are used.

From Fig. 15, as expected, it can be concluded that the RGB-D segmentations yield better performance than the classic RGB segmentations. In particular, in terms of $\mathcal{C}$ and VI, the proposed segmentation highlights over the other segmentations. Conversely, regarding RI and BDE, the gPbD segmentation presents the best results. Considering all four criteria, the proposed methodology seems to be competitive, proving that it is possible to achieve relatively coherent segmentation by independently considering color and depth information. However, there are some parameters to tune, being this a clear limitation in cases where there are no training images. Additionally, the performance of

Figure 16. **Sample images and their segmentations by MLSS, gPb, gPbD, and the proposed methodology.** ***From Left to Right:*** Input image, ground-truth, MLSS [12], gPb [18], gPbD [1], and proposed.



gPb segmentations which only consider RGB data seems to be quite close to the RGB-D segmentations, indicating that there is still much research to be done to effectively exploit the depth information in RGB-D images.

In Fig. 16, from top to bottom, the test images are referred to as image 1, 2, 3, 4 and 5. In images 1 and 4, the proposed segmentation presents considerably better results than the others segmentations, successfully exploiting the depth information. In image 3, however, the presence of only one depth plane causes a miss-interpretation of the captured scene and therefore obtaining a bad segmentation. In cases alike, it would be good if the proposed methodology was able to give priority to the color information. This proves that depth data does not always help to improve segmentation results. In images 2 and 5, the gPbD segmentations are the most accurate. For these cases, under-segmentation of planar regions is present in the proposed segmentations. The background walls in image 2 are confused and the floor in image 5 is merged with part

of the sofa. This indicates that the depth information was not sufficiently exploited, thus, there are many things to improve in the planar and 3D-edge segmentations.

Chapter 4
CONCLUSIONS

The proposed methodology for RGB-D image segmentation can be viewed as a framework for the integration of color and depth information, aimed to provide useful scene interpretation. It was shown that superpixels can be a good starting point to reach meaningful segmentation. Four independent color and depth segmentations were considered, however, this approach is not limited to that quantity, suggesting a potential extension to multi-modal image segmentation. Reported results give an insight of the promising features of depth and 3D data.

One major limitation is the fact that all segmentations are always incorporated to do region merging decisions, however, this can be counterproductive in some cases, making necessary the ability to discern what information is not convenient to consider based on an evaluation of its reliability. For the planar and 3D-edge segmentations, the obtained results are quite coarse since the depth information is noisy, so, to the extent that these results can be improved, it is expected to attain more features and better capture geometry information in a certain scene.

For future work, it is required to propose more robust experiments using different validation strategies over training and test images. Additionally, computational cost must be evaluated in terms of time, memory and complexity to perform a more complete comparison with state-of-the-art techniques.

In summary, RGB-D image segmentation is a topic in recent development with powerful and interesting advantages, allowing to deal with several applications that have traditionally been very hard with the sole use of RGB data, such as object detection

and machine vision. The present work is then intended to contribute to the correct perception and representation of scene entities by leveraging the complementary nature of color and depth information. A review of many key aspects of RGB-D images was presented in order to motivate its study and analysis. Obtained conclusions are expected to pave the way for future research in this topic, opening new possibilities in the field of computer vision.

REFERENCES

[1] Gupta, S., Arbelaez, P., and Malik, J. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 564–571.

[2] Pertuz, S., Garcia, M. A., and Puig, D. Focus-aided scene segmentation. *Computer Vision and Image Understanding 133* (2015), 66 – 75.

[3] Setarehdan, S., and Singh, S. *Advanced Algorithmic Approaches to Medical Image Segmentation: State-of-the-Art Applications in Cardiology, Neurology, Mammography and Pathology*. Advances in Computer Vision and Pattern Recognition. Springer London, 2012.

[4] Sonka, M., Hlavac, V., and Boyle, R. *Image Processing, Analysis, and Machine Vision*. Cengage Learning, 2014.

[5] van de Sande, K. E. A., Uijlings, J. R. R., Gevers, T., and Smeulders, A. W. M. Segmentation as selective search for object recognition. In *2011 International Conference on Computer Vision* (Nov 2011), pp. 1879–1886.

[6] Veltkamp, R., Burkhardt, H., and Kriegel, H. *State-of-the-Art in Content-Based Image and Video Retrieval*. Computational Imaging and Vision. Springer Netherlands, 2013.

[7] Peng, B., Zhang, L., and Zhang, D. A survey of graph theoretical approaches to image segmentation. *Pattern Recognition 46*, 3 (2013), 1020 – 1038.

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

[9] Shi, J., and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (Aug 2000), 888–905.

[10] Cour, T., Benezit, F., and Shi, J. Spectral segmentation with multiscale graph decomposition. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (2005), vol. 2, IEEE, pp. 1124–1131.

[11] Li, Z., Wu, X. M., and Chang, S. F. Segmentation using superpixels: A bipartite graph partitioning approach. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (June 2012), pp. 789–796.

[12] Kim, T. H., and Lee, K. M. Learning full pairwise affinities for spectral segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (June 2010), pp. 2101–2108.

[13] Comaniciu, D., and Meer, P. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (May 2002), 603–619.

[14] Vedaldi, A., and Soatto, S. Quick shift and kernel methods for mode seeking. In *European Conference on Computer Vision* (2008), Springer, pp. 705–718.

[15] Paris, S., and Durand, F. A topological approach to hierarchical segmentation using mean shift. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (June 2007), pp. 1–8.

[16] Yu, Z., Li, A., Au, O. C., and Xu, C. Bag of textons for image segmentation via soft clustering and convex shift. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (2012), IEEE, pp. 781–788.

[17] Nock, R., and Nielsen, F. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 11 (Nov 2004), 1452–1458.

[18] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 5 (May 2011), 898–916.

[19] Huang, D., Lai, J.-H., Wang, C.-D., and Yuen, P. C. Ensembling over-segmentations: From weak evidence to strong segmentation. *Neurocomputing 207* (2016), 416 – 427.

[20] Deng, Y., and Manjunath, B. S. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 8 (Aug 2001), 800–810.

[21] Han, J., Shao, L., Xu, D., and Shotton, J. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Transactions on Cybernetics 43*, 5 (Oct 2013), 1318–1334.

[22] Rusu, R. B., and Cousins, S. 3d is here: Point cloud library (pcl). In *2011 IEEE International Conference on Robotics and Automation* (May 2011), pp. 1–4.

[23] Liu, C., Yuan, D., and Zhao, H. 3d point cloud denoising and normal estimation for 3d surface reconstruction. In *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (Dec 2015), pp. 820–825.

[24] Feng, C., Taguchi, Y., and Kamat, V. R. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (May 2014), pp. 6218–6225.

[25] Hulik, R., Spanel, M., Smrz, P., and Materna, Z. Continuous plane detection in point-cloud data based on 3d hough transform. *Journal of visual communication and image representation 25*, 1 (2014), 86–97.

[26] Erdogan, C., Paluri, M., and Dellaert, F. Planar segmentation of rgbd images using fast linear fitting and markov chain monte carlo. In *Computer and Robot Vision (CRV), 2012 Ninth Conference on* (2012), IEEE, pp. 32–39.

[27] Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. Indoor segmentation and support inference from rgbd images. *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision* (2012), 746–760.

[28] Richtsfeld, A., Mörwald, T., Prankl, J., Zillich, M., and Vincze, M. Learning of perceptual grouping for object segmentation on rgb-d data. *Journal of Visual Communication and Image Representation 25*, 1 (2014), 64 – 73. Visual Understanding and Applications with RGB-D Cameras.

[29] Ren, X., Bo, L., and Fox, D. Rgb-(d) scene labeling: Features and algorithms. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (June 2012), pp. 2759–2766.

[30] Yang, J., Gan, Z., Li, K., and Hou, C. Graph-based segmentation for rgb-d data using 3-d geometry enhanced superpixels. *IEEE Transactions on Cybernetics 45*, 5 (May 2015), 927–940.

[31] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 11 (Nov 2012), 2274–2282.

[32] Klette, R. *Concise Computer Vision: An Introduction into Theory and Algorithms*. Undergraduate Topics in Computer Science. Springer London, 2014.

[33] Gonzalez, R., and Woods, R. *Digital Image Processing*. Pearson/Prentice Hall, 2008.

[34] Blake, A., and Isard, M. The condensation algorithm-conditional density propagation and applications to visual tracking. In *Advances in Neural Information Processing Systems* (1997), pp. 361–367.

[35] Cremers, D., Rousson, M., and Deriche, R. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International journal of computer vision 72*, 2 (2007), 195–215.

[36] Boykov, Y., and Funka-Lea, G. Graph cuts and efficient nd image segmentation. *International journal of computer vision 70*, 2 (2006), 109–131.

[37] Szeliski, R. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[38] Belongie, S., Carson, C., Greenspan, H., and Malik, J. Color- and texture-based image segmentation using em and its application to content-based image retrieval. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)* (Jan 1998), pp. 675–682.

[39] Yang, A. Y., Wright, J., Ma, Y., and Sastry, S. S. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding 110*, 2 (2008), 212 – 225.

[40] Shafarenko, L., Petrou, H., and Kittler, J. Histogram-based segmentation in a perceptually uniform color space. *IEEE transactions on image processing 7*, 9 (1998), 1354–1358.

[41] Tobias, O. J., and Seara, R. Image segmentation by histogram thresholding using fuzzy sets. *IEEE transactions on Image Processing 11*, 12 (2002), 1457–1465.

[42] Beucher, S., et al. The watershed transformation applied to image segmentation. *SCANNING MICROSCOPY-SUPPLEMENT-* (1992), 299–299.

[43] Lu, C. S., Chung, P. C., and Chen, C. F. Unsupervised texture segmentation via wavelet transform. *Pattern Recognition 30*, 5 (1997), 729–742.

[44] von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing 17*, 4 (2007), 395–416.

[45] Ren, X., and Malik, J. Learning a classification model for segmentation. In *Proceedings Ninth IEEE International Conference on Computer Vision* (Oct 2003), pp. 10–17 vol.1.

[46] Liu, Y., Condessa, F., Bioucas-Dias, J., Li, J., and Plaza, A. Convex formulation for hyperspectral image classification with superpixels. In *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International* (2016), IEEE, pp. 3294–3297.

[47] Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., and Siddiqi, K. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 12 (Dec 2009), 2290–2297.

[48] Veksler, O., Boykov, Y., and Mehrani, P. *Superpixels and Supervoxels in an Energy Optimization Framework*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 211–224.

[49] Liu, M. Y., Tuzel, O., Ramalingam, S., and Chellappa, R. Entropy rate superpixel segmentation. In *CVPR 2011* (June 2011), pp. 2097–2104.

[50] Van den Bergh, M., Boix, X., Roig, G., de Capitani, B., and Van Gool, L. *SEEDS: Superpixels Extracted via Energy-Driven Sampling*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 13–26.

[51] Spinello, L., and Arras, K. O. Leveraging rgb-d data: Adaptive fusion and domain adaptation for object detection. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (2012), IEEE, pp. 4469–4474.

[52] Bo, L., Ren, X., and Fox, D. Unsupervised feature learning for rgb-d based object recognition. In *Experimental Robotics* (2013), Springer, pp. 387–402.

[53] Sung, J., Ponce, C., Selman, B., and Saxena, A. Human activity detection from rgbd images. *plan, activity, and intent recognition 64* (2011).

[54] Reyes, M., Domínguez, G., and Escalera, S. Featureweighting in dynamic timewarping for gesture recognition in depth data. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on* (2011), IEEE, pp. 1182–1188.

[55] Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *In the 12th International Symposium on Experimental Robotics (ISER* (2010), Citeseer.

[56] Gupta, S., Arbeláez, P., Girshick, R., and Malik, J. Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation. *International Journal of Computer Vision 112*, 2 (2015), 133–149.

[57] Cruz, L., Lucio, D., and Velho, L. Kinect and rgbd images: Challenges and applications. In *Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2012 25th SIBGRAPI Conference on* (2012), IEEE, pp. 36–49.

[58] Smisek, J., Jancosek, M., and Pajdla, T. 3d with kinect. In *Consumer depth cameras for computer vision*. Springer, 2013, pp. 3–25.

[59] Sturm, P. *Pinhole Camera Model*. Springer US, Boston, MA, 2014, pp. 610–613.

[60] Hoiem, D., Efros, A. A., and Hebert, M. Recovering occlusion boundaries from an image. *International Journal of Computer Vision 91*, 3 (2011), 328–346.

[61] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. From contours to regions: An empirical evaluation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (June 2009), pp. 2294–2301.

[62] Unnikrishnan, R., Pantofaru, C., and Hebert, M. Toward objective evaluation of image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 6 (June 2007), 929–944.

[63] Meilă, M. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd international conference on Machine learning* (2005), ACM, pp. 577–584.

[64] Freixenet, J., Muñoz, X., Raba, D., Martí, J., and Cufí, X. *Yet Another Survey on Image Segmentation: Region and Boundary Information Integration*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 408–422.

[65] Goder, A., and Filkov, V. Consensus clustering algorithms: Comparison and refinement. In *2008 Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments (ALENEX)* (2008), SIAM, pp. 109–117.

[66] Dhankhar, P., and Sahu, N. A review and research of edge detection techniques for image segmentation. *International Journal of Computer Science and Mobile Computing (IJCSMC) 2*, 7 (2013), 86–92.

[67] Martin, D. R., Fowlkes, C. C., and Malik, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence 26*, 5 (2004), 530–549.

BIBLIOGRAPHY

ACHANTA, Radhakrishna, *et al.* SLIC superpixels compared to state-of-the-art superpixel methods. In: IEEE transactions on pattern analysis and machine intelligence 34.11 (2012): 2274-2282.

ARBELAEZ, Pablo, *et al.* Contour detection and hierarchical image segmentation. In: IEEE transactions on pattern analysis and machine intelligence 33.5 (2011): 898-916.

ARBELAEZ, Pablo, *et al.* From contours to regions: An empirical evaluation. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

BELONGIE, Serge, *et al.* Color-and texture-based image segmentation using EM and its application to content-based image retrieval. Computer Vision, 1998. Sixth International Conference on. IEEE, 1998.

COMANICIU, Dorin and MEER, Peter. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on pattern analysis and machine intelligence 24.5 (2002): 603-619.

CRUZ, Leandro; LUCIO, Djalma, and VELHO, Luiz. Kinect and rgbd images: Challenges and applications. Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2012 25th SIBGRAPI Conference on. IEEE, 2012.

FELZENSZWALB, Pedro F. and HUTTENLOCHER, Daniel P. Efficient graph-based image segmentation. International journal of computer vision 59.2 (2004): 167-181.

FENG, Chen; TAGUCHI, Yuichi and KAMAT, Vineet R. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014.

FREIXENET, Jordi, *et al.* Yet another survey on image segmentation: Region and boundary information integration. Computer Vision—ECCV 2002 (2002): 21-25.

GODER, Andrey and FILKOV, Vladimir. Consensus clustering algorithms: Comparison and refinement. 2008 Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments (ALENEX). Society for Industrial and Applied Mathematics, 2008.

GONZALEZ, Rafael C. and WOOD, Richard E. Digital image processing, 2nd Edtn. Pearson/Prentice Hall, 2008.

GUPTA, Saurabh; ARBELAEZ, Pablo and MALIK, Jitendra. Perceptual organization and recognition of indoor scenes from RGB-D images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013.

HAN, Jungong, *et al.* Enhanced computer vision with microsoft kinect sensor: A review. IEEE transactions on cybernetics 43.5 (2013): 1318-1334.

HUANG, Dong, *et al.* Ensembling over-segmentations: From weak evidence to strong segmentation. Neurocomputing 207 (2016): 416-427.

KLETTE, Reinhard. Concise computer vision: An Introduction into Theory and Algorithms. Undergraduate Topics in Computer Science. Springer, London, 2014.

LI, Zhenguo; WU, Xiao-Ming and CHANG, Shih-Fu. Segmentation using superpixels: A bipartite graph partitioning approach. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

MEILA, Marina. Comparing clusterings: an axiomatic view. Proceedings of the 22nd international conference on Machine learning. ACM, 2005.

PENG, Bo; ZHANG, Lei and ZHANG, David. A survey of graph theoretical approaches to image segmentation." Pattern Recognition 46.3 (2013): 1020-1038.

SETAREHDAN, Kamaledin and SINGH, Sameer. Advanced algorithmic approaches to medical image segmentation: state-of-the-art applications in cardiology, neurology, mammography and pathology. Springer Science & Business Media, 2012.

SHI, Jianbo and MALIK, Jitendra. Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence 22.8 (2000): 888-905.

SILBERMAN, Nathan, *et al.* Indoor segmentation and support inference from rgbd images. Computer Vision–ECCV 2012 (2012): 746-760.

SONKA, Milan; HLAVAC, Vaclav and BOYLE, Roger. Image processing, analysis, and machine vision. Cengage Learning, 2014.

SUNG, Jaeyong, *et al.* Human Activity Detection from RGBD Images. Plan, Activity, and Intent Recognition. Vol 64 (2011).

SZELISKI, Richard. Computer vision: algorithms and applications. Springer Science & Business Media, 2010.

UNNIKRISHNAN, Ranjith; PANTOFARU, Caroline and HEBERT, Martial. Toward objective evaluation of image segmentation algorithms. IEEE transactions on pattern analysis and machine intelligence 29.6 (2007): 929-944.

VAN DE SANDE, Koen E., *et al.* Segmentation as selective search for object recognition. Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

YANG, Jingyu, *et al.* Graph-based segmentation for RGB-D data using 3-D geometry enhanced superpixels. IEEE transactions on cybernetics 45.5 (2015): 927-940.