SPATIAL 3D LOCAL DESCRIPTORS FOR OBJECT RECOGNITION IN RGB-D IMAGES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

K. BERKER LOĞOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
INFORMATION SYSTEMS

JANUARY 2016

Approval of the thesis:

SPATIAL 3D LOCAL DESCRIPTORS FOR OBJECT RECOGNITION IN RGB-D IMAGES

submitted by **K. BERKER LOĞOĞLU** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Information Systems Department, Middle East Technical University** by,

Prof. Dr. Nazife Baykal _____
Dean, Graduate School of **Informatics Institute**

Prof. Dr. Yasemin Yardımcı Çetin _____
Head of Department, **Information Systems**

Assoc. Prof. Dr. Alptekin Temizel _____
Supervisor, **Modeling and Simulation, METU**

Assist. Prof. Dr. Sinan Kalkan _____
Co-supervisor, **Computer Engineering, METU**

Examining Committee Members:

Prof. Dr. Yasemin Yardımcı Çetin
Information Systems, METU _____

Assoc. Prof. Dr. Alptekin Temizel
Modeling and Simulation, METU _____

Assist. Prof. Dr. Aykut Erdem
Computer Engineering, Hacettepe University _____

Assist. Prof. Dr. Erhan Eren
Information Systems, METU _____

Assoc. Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: K. BERKER LOĞOĞLU

Signature :

ABSTRACT

SPATIAL 3D LOCAL DESCRIPTORS FOR OBJECT RECOGNITION IN RGB-D IMAGES

Loğoğlu, K. Berker
Ph.D., Department of Information Systems
Supervisor : Assoc. Prof. Dr. Alptekin Temizel
Co-Supervisor : Assist. Prof. Dr. Sinan Kalkan

January 2016, 103 pages

Introduction of the affordable but relatively high resolution color and depth synchronized RGB-D sensors, along with the efforts on open-source point-cloud processing tools boosted research in both computer vision and robotics. One of the key areas which have drawn particular attention is object recognition since it is one of the crucial steps for various applications. In this thesis, two spatially enhanced local 3D descriptors are proposed for object recognition tasks: *Histograms of Spatial Concentric Surflet-Pairs* (SPAIR) and *Colored SPAIR* (CoSPAIR). The proposed descriptors are compared against the state-of-the-art local 3D descriptors that are available in Point Cloud Library (PCL) and their object recognition performances are evaluated on several publicly available datasets. The experiments demonstrate that the proposed CoSPAIR descriptor outperforms the state-of-the-art descriptors in both category-level and instance-level recognition tasks. The performance gains are observed to be up to 9.9 percentage points for category-level recognition and 16.49 percentage points for instance-level recognition over the second-best performing descriptor.

Keywords: Point Clouds, RGB-D, 3D Descriptors

ÖZ

RGB-D İMGELERDE NESNE TANIMA İÇİN ÜÇ BOYUTLU UZAMSAL YEREL TANIMLAYICILAR

Loğoğlu, K. Berker
Doktora, Bilişim Sistemleri Bölümü
Tez Yöneticisi : Doç. Dr. Alptekin Temizel
Ortak Tez Yöneticisi : Yrd. Doç. Dr. Sinan Kalkan

Ocak 2016 , 103 sayfa

Ucuz ve göreceli olarak yüksek çözünürlüklü sayılabilecek, renk ve derinlik bilgilerini eş zamanlı kaydedebilen RGB-D algılayıcıların yaygınlaşması ile birlikte, açık kaynak kodlu nokta bulutu işleme yazılımları üzerine çalışmaların da artması robotik ve üç boyutlu görü alanlarındaki çalışmaları önemli ölçüde arttırmıştır. Bu alanlardaki birçok uygulamanın önemli adımlarından biri olması nedeni ile, özellikle ilgi çeken konuların en başında nesne tanıma gelmektedir. Bu tezde, özellikle nesne tanıma alanında kullanılmak üzere iki adet, üç boyutlu, uzamsal nokta bulutu tanımlayıcı önerilmiştir; *Uzamsal Eşmerkezli Yönlü Yüzey Nokta Çiftleri Histogramı* (SPAIR) ve *Renkli Uzamsal Eşmerkezli Yönlü Yüzey Nokta Çiftleri Histogramı* (CoSPAIR). Önerilen tanımlayıcılar, birçok halka açık veri kümesi üzerinde, açık kaynak kodlu "Nokta Bulutu İşleme Kütüphanesi" (Point Cloud Library - PCL) içinde bulunan en gelişkin tekniklerle karşılaştırılmıştır. Gerçekleştirilen bu deneyler göstermiştir ki, önerilen CoSPAIR tanımlayıcısı, en gelişkin yöntemlerden hem kategori hem de örnek seviyesinde önemli miktarda üstündür. Elde edilen başarım artışının kategori seviyesinde 9.9, örnek seviyesinde ise 16.49 yüzdelik puana kadar çıkabildiği gözlemlenmiştir.

Anahtar Kelimeler: Nokta Bulutu, RGB-D, 3B Tanımlayıcılar

TO MY BELOVED FAMILY

ACKNOWLEDGMENTS

First and foremost I would like to thank my advisors Dr. Alptekin Temizel and Dr. Sinan Kalkan for their guidance and supervision.

I would also thank my thesis committee for their valuable comments and suggestions.

I would like to thank my beloved family for their unconditional support and motivation.

Last but not least, I would like to thank my wife Eda, without whose love, encouragement and support, I would not have finished this thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

3DSC	3D Shape Context
ANN	Approximate Nearest Neighbor
CoSPAIR	Color - Histograms of Spatial Concentric Surflet-Pairs
CSHOT	Color - Signature of Histograms of Orientations
CVFH	Clustered Viewpoint Feature Histogram
ESF	Ensemble of Shape Functions
FPFH	Fast Point Feature Histograms
IR	Infra-Red
ISS	Intrinsic Shape Signatures
K-D TREE	K-Dimensional Tree
NN	Nearest Neighbor
PCL	Point Cloud Library
PFH	Point Feature Histograms
RGB-D	Red, Green, Blue and Depth Channels
SHOT	Signature of Histograms of Orientations
SLAM	Simultaneous Localization and Mapping
SPAIR	Histograms of Spatial Concentric Surflet-Pairs
SPFH	Simplified Point Feature Histogram
VFH	Viewpoint Feature Histogram
VGA	Video Graphics Array

CHAPTER 1

INTRODUCTION

Object recognition is one of the major and crucial research areas in computer vision with applications in surveillance, robotics, medical image analysis, remote sensing and autonomous driving. It is a challenging task by its nature because of variations in scale, pose, illumination, viewpoint, imaging conditions, visual clutter, occlusions and deformation.

Research on object recognition can be analyzed in mainly two categories; 2D methods which deal with 2D images and videos, and 3D methods which deal with 3D scans (i.e. point clouds and meshes). 2D object recognition has been a more active research area in the past few decades thus can be considered rather mature [9, 10]. However, the trend is changing due to new technologies which make acquisition of 3D data simpler and cheaper.

Recently, with the introduction of affordable but relatively high resolution color and depth synchronized (RGB-D) cameras, such as Kinect, a new era has begun in robotics and 3D computer vision. Correspondingly, efforts on point cloud processing increased significantly. These advancements boosted research in 3D computer vision thus 3D object recognition.

1.1 Kinect® and 3D Sensors

The game-changer 3D sensor, Kinect® was introduced in 2010. The first version included an infrared (IR) projector and sensor along with a color camera with a VGA

resolution. It used *structured light* technique to sense depth. In 2013, it has evolved significantly and its core technology has changed to *time-of-flight*. The details about the versions are given in Table 1.1.

Table 1.1: Specifications of the Kinect versions.

	Kinect v1	**Kinect v2**
Technology	Infrared Structured Light	Infrared Time-of-Flight
Color Camera Resolution	640×480	1920×1080
Depth Camera Resolution	320×240	512×424
Depth Range	0.8 - 4.0 m	0.5 - 4.5 m
Field of View	57° h. & 43° v.	70° h. & 60° v.

The images obtained by Kinect (and similar sensors) are called RGB-D where "RGB" represents the three primary color (red, green and blue) channels captured by the RGB camera and "D" for the depth data. The color camera captures images at 640×480 pixels with 8-bit per channel whereas the depth data is obtained by *structured-light* technique that is shown in Figure 1.1. Kinect has an IR projector and an IR sensor. The IR projector projects a unique IR pattern (the exact pattern used by Kinect is the one used in Figure 1.1). The pattern is deformed by the shape of the object/scene which is then captured by the IR camera. The depth information is extracted by calculating the disparity from the original projected pattern.

The popularity of Kinect has led many companies to produce similar products such as Intel's RealSense embedded sensor [11] (Figure 1.2b) that is targeted for mobile as well as desktop computers. There is even ongoing work for embedding such sensors on mobile devices such as smartphones and tablets, e.g., Google's Project Tango [12] tablets which use Infineon's embedded Real3 time-of-flight sensor (Figure 1.2d).

It is important to note that, besides these aforementioned relatively cheap 3D sensors, there are (depth-only) ones that are targeted for more demanding applications such as 360° field-of-view and very high data rate LIDARs (Figure 1.2e) that are used in "self-driving" cars or laser scanners (Figure 1.2f) for very high resolution scanning applications.

The aforementioned advancements on sensor technology boosted the developments in many computer vision and robotics research areas including object detection, object

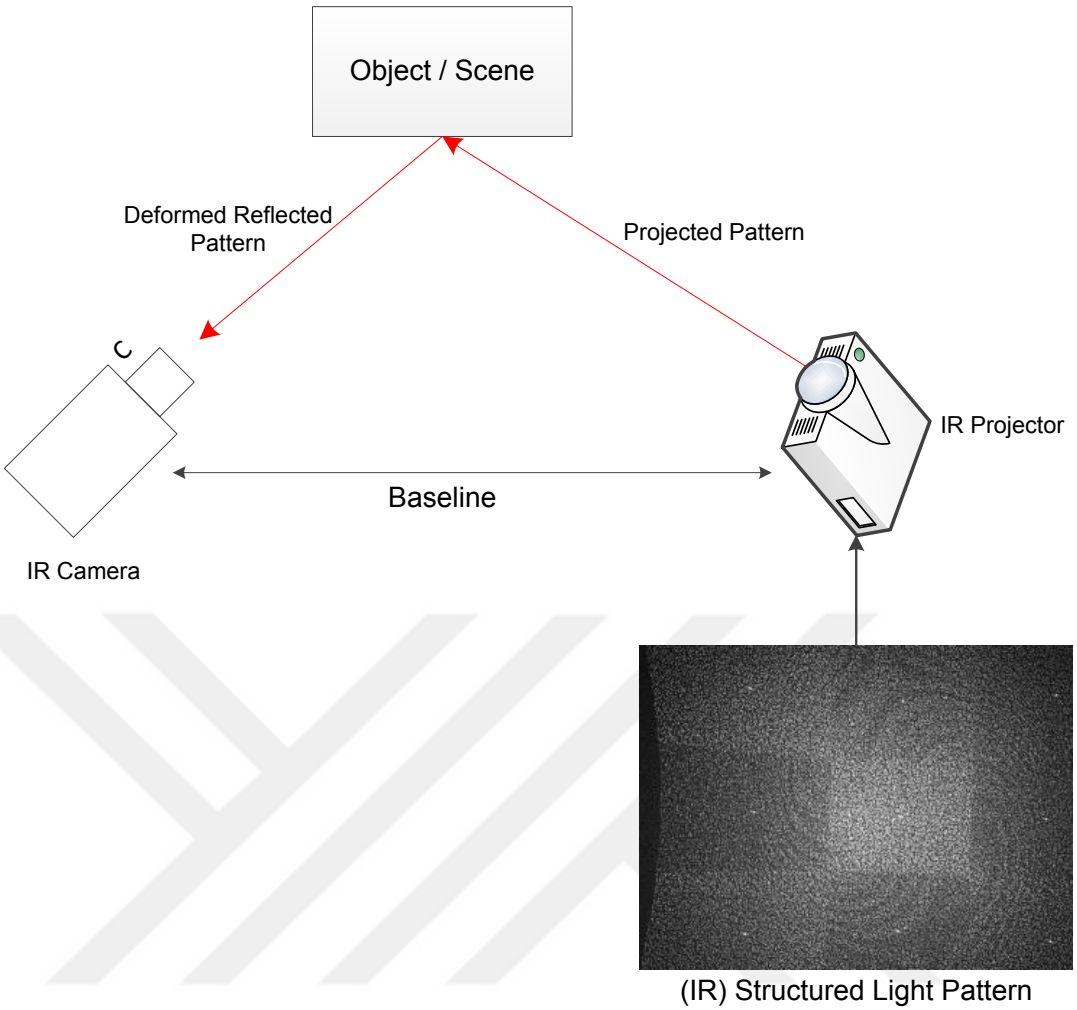
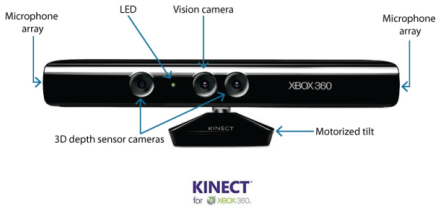
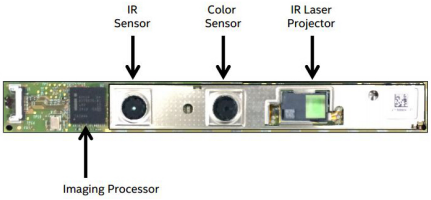


Figure 1.1: Structured light imaging system.

recognition, object tracking, human activity analysis, gesture analysis and "simultaneous localization and mapping" (SLAM). Among these, object recognition is one of the most important topics for robotics since it is indispensable for the proper interaction of robots with their surrounding.



(a) Microsoft Kinect v1

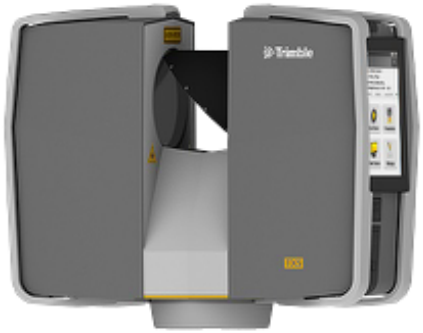


(b) Intel Realsense sensor [11]

(c) Microsoft Kinect v2

(d) Infinion Real3 Sensor [13]

(e) Velodyne HDL-64e [14]



(f) Trimble TX5 laser scanner [15]

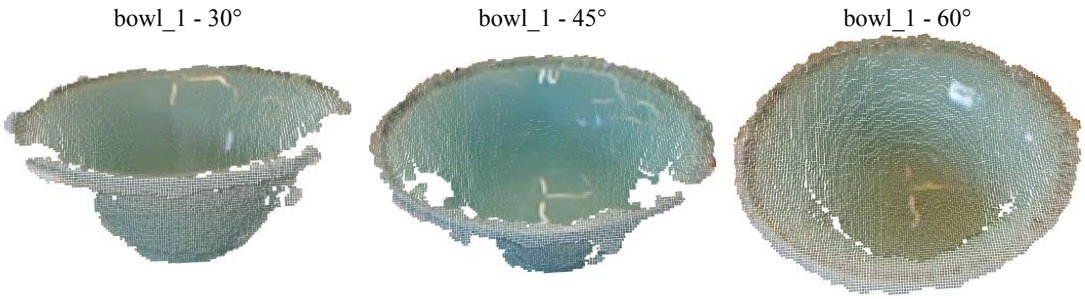
Figure 1.2: Various 3D sensors.

1.2 Problem Definition

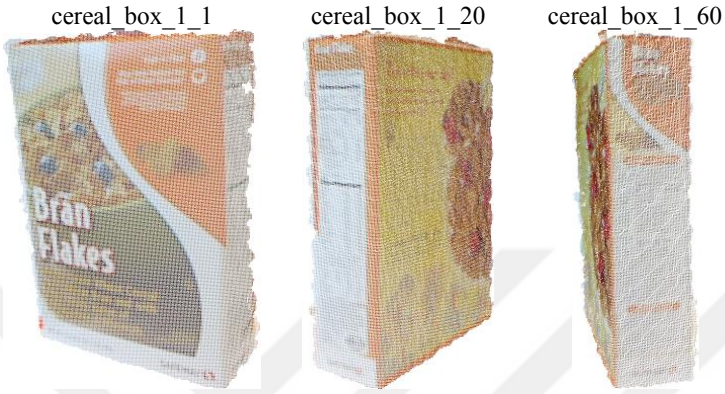
Object recognition can be performed at two different levels: *category-level* or *instance-level*. In *category-level* object recognition, an object is classified into pre-defined categories such as *cereal box* or *soda can*, whereas in *instance-level* recognition, specific instances of the objects such as "Cheerios" or "Pepsi can" are recognized. While promising results have been reported for category-level object recognition, instance-level recognition remains a more challenging problem [5, 16, 7]. The success of both object recognition tasks is directly related to the descriptors used thus there have been tremendous effort in developing 3D descriptors. Among these descriptors, only a few utilize shape and texture/color information together to take advantage of the color and depth synchronized data obtained from the aforementioned RGB-D sensors. It has been shown that such hybrid descriptors perform especially well for instance-level recognition [16] although there is much room for improvement.

Object recognition has many challenges; scale, pose, illumination, viewpoint, imaging conditions, visual clutter, occlusions and deformation. Although by using certain type of 3D sensors (ones that use IR or laser technology) some of these challenges, such as illumination, can be overcome, most of the challenges still persists. Additionally, *instance-level* recognition adds further challenges due to similarities between object instances.

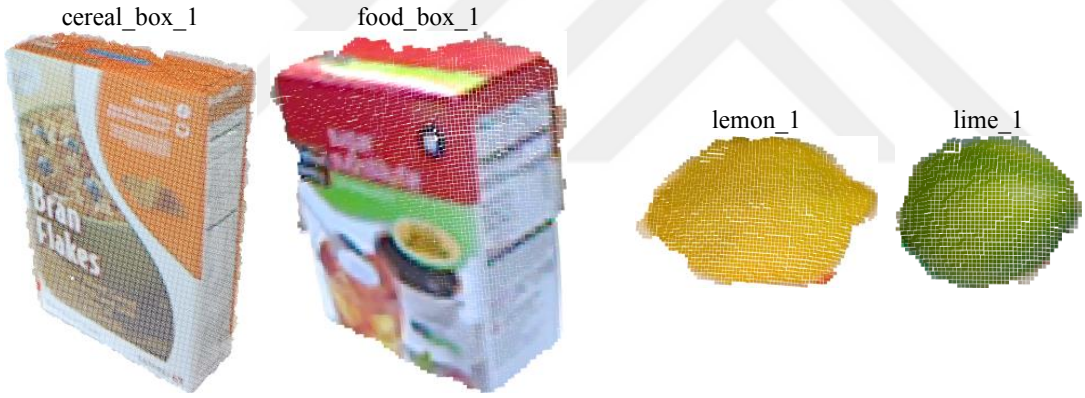
Figure 1.3 shows some of the challenging situations with captured images from the datasets used in this thesis. One of the primary challenge in object recognition is that there are infinite number of viewpoints where an object can be observed from. Consequently, the observation changes significantly depending on the viewpoint. To demonstrate, Figure 1.3a and 1.3b show how the location of the sensor, and how the viewpoint around an object drastically affects the observations, respectively. While for a certain type of object the challenges are many, it can also be challenging to differentiate certain object types that are similar in shape as shown in Figure 1.3c. Furthermore, as mentioned before *instance-level* recognition is even further challenging since the instances of a certain type of object can be extremely similar as shown in Figure 1.3d.



(a) Scans from different sensor locations / heights.



(b) Scans from different viewpoints.



(c) Similarity between different object categories.

(d) Similarity between different instances.

Figure 1.3: Challenges of 3D object recognition.

Additionally, there are challenges that are specific to certain type of sensors used for capturing the object. Some type of sensors fail to capture depth from reflective surfaces such as metals as shown in Figure 1.4a and transparent surfaces as shown in Figure 1.4b which makes recognizing these objects extremely difficult due to lack of data.

(a) Sensor fail to capture metallic surfaces.

(b) Sensor fail to capture transparent surfaces.

Figure 1.4: Challenges due to sensor incapability.

1.3 Contributions

In this thesis, we propose a novel 3D descriptor which utilizes shape and color information simultaneously - particularly targeting the instance-level object recognition problem. Along with this descriptor, a shape-only one that can be used with sensors that lack color data is also proposed.

The proposed descriptors are compared against the state-of-the-art local 3D descriptors that are available in Point Cloud Library (PCL) [17, 18] and their object recognition performances are evaluated on several publicly available datasets. The experiments demonstrate that the proposed shape+color descriptor outperforms the state-of-the-art descriptors in both category-level and instance-level object recogni-

tion tasks.

The proposed descriptors are planned to be shared with robot / computer vision community as open-source software through the Point Cloud Library.

Additionally, the work presented in this thesis has led to the following publication in Robotics and Autonomous Systems Journal's special issue on 3D robot perception with the Point Cloud Library:

- K. Berker Logoglu, Sinan Kalkan, Alptekin Temizel, "CoSPAIR: Colored Histograms of Spatial Concentric Surflet-Pairs for 3D object recognition", Robotics and Autonomous Systems, Volume 75, Part B, January 2016, Pages 558-570, ISSN 0921-8890, http://dx.doi.org/10.1016/j.robot.2015.09.027.

1.4 Outline of the Thesis

The thesis is organized as follows; firstly, in Chapter 2, the taxonomy for 3D descriptors is presented and the work on each category is discussed. The descriptors that are popular in literature as well as the ones that are available in the highly popular Point Cloud Library [19] are further detailed.

Next, the proposed descriptors *Histograms of Spatial Concentric Surflet-Pairs* and *Colored Histograms of Spatial Concentric Surflet-Pairs* are detailed in Chapter 3. A brief matching performance comparison (visual) with the state-of-the-art descriptors is also provided.

In Chapter 4 the common steps in the extraction flow of the proposed and compared descriptors are detailed. In Section 4.1 spatial decomposition of 3D space with k-d trees along with nearest neighbor search in *k-d tree*s are detailed. In Section 4.2, estimation of surface normals which provides the basis for the extraction of proposed features is explained. In Section 4.3 keypoint selection that are used in the evaluation of the descriptors are detailed.

Next, in Chapter 5, the proposed descriptors are compared to the state-of-the-art 3D descriptors and their both *category-level* and *instance-level* object recognition per-

formances are evaluated on publicly available RGB-D datasets. In Section 5.1, the method and metrics that are used in evaluating the proposed and compared descriptors are explained. In Section 5.2, the datasets that the experiments are conducted on are detailed. In Section 5.3, the effects of some design parameters specific to our proposed descriptors are investigated. In Section 5.4, the effects of various keypoint selection methods on performance are investigated. In Sections 5.5, 5.6, 5.7 and 5.8 the performance of the proposed descriptors on the chosen datasets is investigated and compared to state-of-the-art. In Section 5.9, the extraction and matching times of the descriptors are investigated. In Section 5.10, the effects of the size of the objects on the recognition performance are investigated.

Finally, the conclusions and future work are stated in Chapter 6.

CHAPTER 2

3D DESCRIPTORS

The performance of object recognition is directly related to the descriptors used, and there have been tremendous effort in developing 3D descriptors. The descriptors can be categorized mainly into three as point-wise, local and global, based on the size of the support with respect to the point to be described. In the literature, the taxonomy is further detailed by Akgul et al. [20] for global 3D descriptors, by Salti and Tombari [3, 21] for local 3D descriptors as given in Figure 2.1 and Table 2.1.

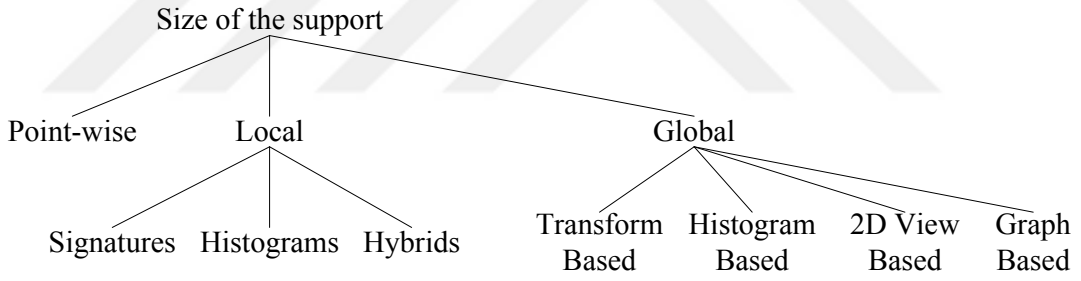


Figure 2.1: Taxonomy of 3D descriptors.

The point-wise descriptors are computed directly only on the point to be described (keypoint). They are simple and efficient however they lack robustness and descriptive power of local/global descriptors. The local descriptors embed characteristics of the neighboring points of the keypoint within a support (usually a spherical region with a support radius r). As a result, local descriptors are more descriptive and robust to clutter and occlusion. On the other hand, global descriptors are extracted from the entire object and have the ability to characterize the global shape of the object with a single vector, thus being compact and efficient. However, they fail to capture the

specific details and are not robust to occlusion and clutter.

In the following sections, local and global 3D descriptors are detailed with emphasis on the ones that are used on experiments (PFH, PFHRGB, FPFH, SHOT, CSHOT) detailed in Chapter 5.

Table 2.1: Taxonomy of 3D descriptors

Descriptor	Category	Color
SPIN [1, 22]	Local - Histogram	No
3DSC [2]	Local - Histogram	No
FPFH [23]	Local - Histogram	No
PFH [24]	Local - Histogram	No
PFHRGB [19]	Local - Histogram	Yes
ISS [25]	Local - Histogram	No
KPQ [26]	Local - Signature	No
3D SURF [27]	Local - Signature	No
MeshHoG [28]	Local - Hybrid	Yes
SHOT [21]	Local - Hybrid	No
CSHOT [29]	Local - Hybrid	Yes
VFH [4]	Global - Histogram Based	No
CVFH [30]	Global - Histogram Based	No
Shape Distributions [31]	Global - Histogram Based	No
ESF [32]	Global - Histogram Based	No
Spherical Harmonics [33]	Global - Transform Based	No
3D Radon Transform [34]	Global - Transform Based	No
LightField Descriptor [35]	Global - 2D View Based	No
Reeb Graphs [36]	Global - Gaph Based	No

2.1 Point-wise Descriptors

This category of descriptors are computed directly on a single point and based on one or more characteristics. Examples include normals, triangles and shape indexes. They are simple to compute and efficient, however they lack descriptiveness and robustness to noise.

2.2 Global Descriptors

By Akgul et al. the 3D global descriptors are classified into four; histogram based, transform based, 2D view based and graph based [20]. Transform based methods uses signal processing transforms such as Fourier and spherical harmonics. They have the advantage of being compact. Some of the transform based global descriptors are 3D Radon [34] and Rotation Invariant Spherical Harmonics (RISH) [33].

Histogram based methods share the methodology of accumulating a feature in bins defined over the feature space thus discarding all the spatial information [20]. They are easy to implement as well. Some of the widely known histogram-based global descriptors are Viewpoint Feature Histogram (VFH) [4], Clustered Viewpoint Feature Histogram (CVFH) [30], Shape Distributions [31] and Ensemble of Shape Functions (ESF) [32].

VFH is basically the global extended version of FPFH that is detailed in Section 2.3.2.3. In VFH, the statistics of the relative angles between the surface normals at each point to the surface normal at the centroid of the object (instead of query/key-points) are used with an additional viewpoint component that is computed by collecting a histogram of the angles that the viewpoint direction makes with each normal.

In [30], Aldoma et al. proposed an extension to VFH to obtain a more robust reference coordinate frame. The proposed descriptor is called Clustered Viewpoint Feature Histogram (CVFH). CVFH is in fact a semi-global descriptor; in order to obtain a more robust reference coordinate frame, first, smooth and continuous regions (C_i) are identified on the surface S of the object and only the points within C_i are used to calculate the reference frame but all the points on S are used to calculate the angular normal distribution histograms similar to VFH.

Shape Distributions is introduced for content based 3D model retrieval by Osada in 2002 [31]. The proposed descriptor is based on the distribution of distances between two randomly chosen points on the surface of a 3D mesh, called D2. In the work, D2 is compared with additional shape functions which include; the angle enclosed by two lines created from 3 randomly selected points (A3) and area of the triangle formed by three randomly selected points (D3). Wohlkinger and Vincze use these proposed

shape distributions (D2, D3 and A3) and combine with the idea of Ip et al. [37] i.e. classifying each of the computed values into three categories based on the connecting lines created by chosen points: *ON* the surface, *OFF* the surface and *MIXED* [32]. Thus, ESF is composed of 10 concatenated histograms; *ON*/*OFF*/*MIXED* A3 histograms, *ON*/*OFF*/*MIXED* D3 histograms, *ON*/*OFF*/*MIXED* D2 histograms and a final histogram which is the ratio of line distances D2 between *OFF* and *ON* parts.

In the third category, 2D view based, 3D surface is transformed into a set of 2D projections. Among are Lightfield Descriptor [35] and Ohbuchi et al.'s work [38]. In the last category, graph based, a graph is built out of the surface which is transformed into a vector-based numerical description. These methods are complex and hard to obtain. Reeb graphs [36] are among the 2D view based global descriptors.

2.3 Local Descriptors

While global descriptors are extracted from the entire object and have the ability to characterize the global shape of the object with a single vector, thus being compact and efficient, they fail to capture the specific details. On the other hand, local descriptors are extracted from multiple (key)points on the image, therefore they are more robust to occlusion and clutter. Recently, Salti and Tombari categorized local 3D descriptors into three as *histograms*, *signatures* and hybrid methods that can be categorized as both [3, 21].

2.3.1 Signatures

The descriptors in this category require an invariant Local Reference Frame (LRF) and encode the 3D neighborhood of the keypoint via geometric measurements computed on the points within the neighborhood. Even though the methods in this category are highly descriptive, they are sensitive to noise. KPQ [26] and 3D Surf [27] are among the most known descriptors that can be categorized into *signatures*.

2.3.2 Histograms

The descriptors in this category are accumulators of local topological features according to a specific domain (e.g. normal angles, curvatures) [3]. They require a LRF as *signatures* if the domain is based on coordinates, otherwise Repeatable Axis (RA). Compared to *signatures*, they are (generally) more robust to noise but less descriptive [3].

The *spin images* (SPIN) descriptor is one of the most well-known 3D descriptors in this category that is shown to be useful for object recognition tasks [1]. It was introduced by Johnson in 1997 [22]. It should be noted that, although it has been proposed for surface polygonal meshes, the adaptation to point clouds is straightforward.

In *spin images*, an *oriented point* $O = p, n$ is defined as a point p on the surface of an object with the normal n of the tangent plane in p. A unique function that is called *spin map* maps any *oriented point* x onto a 2D space (α, β):

$$SI_O(x) \rightarrow (\alpha, \beta) = \left[\sqrt{\|x - p\|^2 - (n \cdot (x - p))^2}, n \cdot (x - p) \right]. \qquad (2.1)$$

By applying the *spin map* function to all the points on an object, a *spin image* is produced. In Figure 2.2, the *spin images* calculated from various points on a duck model is shown.

Another local histogram based 3D descriptor is the *3D Shape Context* [2] (3DSC) which is proposed by Frome et al. and is directly the 3D extension of 2D shape contexts that is introduced by Belongie et al. [39]. In 3DSC, the support region is chosen as a sphere centered on the query point. The sphere is oriented such that its north pole is aligned with the surface normal of the query point. Additionally, the support region is divided equally in the azimuth and elevation dimensions whereas it is logarithmically divided along the radial dimension as shown in Figure 2.3.

3DSC lacks a repeatable local reference frame thus Tombari et al. proposed an improved Shape Context method called *Unique Shape Context* that employs a unique, unambiguous local reference frame which does not need to compute the descriptor

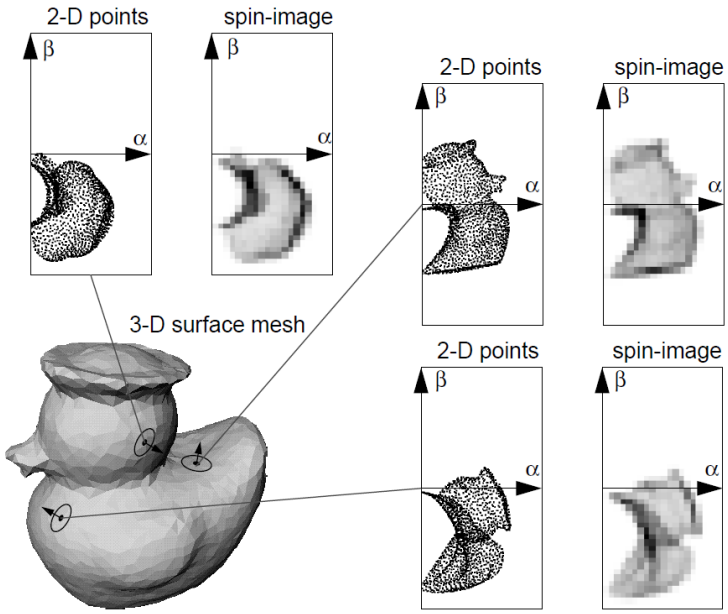


Figure 2.2: Spin images (source: [1]).

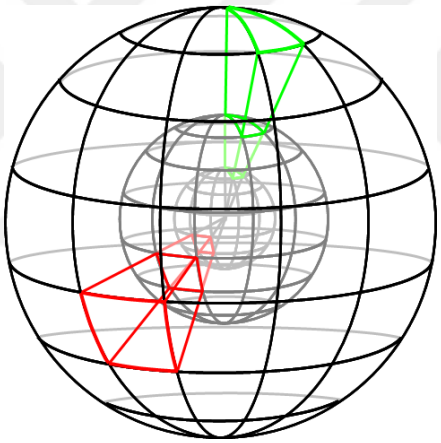


Figure 2.3: 3D shape context support radius (source: [2]).

over multiple rotations on different azimuth directions [40].

2.3.2.1 Point Feature Histograms (PFH)

Point Feature Histograms (PFH) was introduced by Rusu et al. in 2008 as a local descriptor for searching correspondences in 3D point clouds [24]. It is a pose-invariant feature based on geometrical relations of a point's nearest k-neighbors. The geometrical relations are computed from relative orientations of surface normals between point pairs. The main steps for computing a PFH descriptor are:

- For each point $\mathbf{p}$ at which a descriptor is to be extracted, the k-neighboring points within a sphere of a radius r are selected.
- For every pair of points in the sphere, 3 surflet-pair-relation features [41] are calculated (although there are 4 features defined in [41], the fourth feature, the distance between the pairs, is not used since it changes with the viewpoint).
- Histograms of the relations are calculated. Each of the 3-relations is summarized into a 5-bin histogram, and their joint-histogramming yields 5^3 bins in total.

Since PFH considers surflet-pair-relations for every pair of points inside a sphere with radius r, the computational complexity is $O(k^2)$. In other words, for dense point clouds, the time required for extracting PFH descriptors is prohibitively high for practical applications [3, 16, 23].

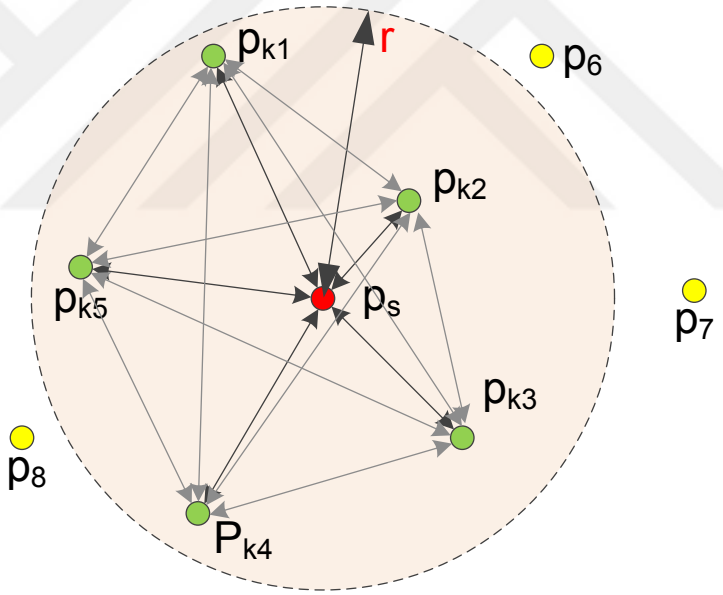


Figure 2.4: The influence region diagram for PFH.

2.3.2.2 Colored Point Feature Histogram (PFHRGB)

PFHRGB is an extension of the PFH. It includes three more histograms in addition to those in PFH. These additional histograms represent the ratio between color channels of point pairs, thus bringing the total size of the descriptor to 250 [19]. Adding color

information has been shown to increase the performance of PFH [16] but PFHRGB suffers from the same drawback as PFH, i.e., being computationally expensive.

2.3.2.3 Fast Point Feature Histograms (FPFH)

Fast Point Feature Histograms is an improvement over PFH in the sense that the computational complexity is reduced down to $O(k)$ from $O(k^2)$ [23]. This is achieved by generating the histograms from the relations between only a point and its k-neighboring points inside the support radius r, instead of analyzing relations between all pairs inside the spherical support. This is called *Simplified Point Feature Histogram* (SPFH). To re-compensate for the missing connections (compared to PFH where all the point-pairs contribute to the descriptor), the SPFHs that are extracted at the neighbors of a point $\mathbf{p}$ are weighted and summed according to their spatial distance:

$$FPFH(\mathbf{p}) = SPFH(\mathbf{p}) + \frac{1}{k}\sum_{i=1}^{k}\frac{1}{w_i}\cdot SPFH(\mathbf{p}_i), \tag{2.2}$$

where the weight w_i represents the distance between source/query point $\mathbf{p}$ and a neighbor point $\mathbf{p}_i$. It should be noted that SPFH values should be calculated for all the points on the surface to be described and the effective radius implicitly becomes $2r$ since additional point pairs outside the r radius are included as well. Although being significantly faster than PFH and PFHRGB [16], FPFH was shown to be an order of magnitude slower than its alternatives, e.g., SHOT [3]. Moreover, FPFH lacks color information.

2.3.3 Hybrids

SHOT [21], CSHOT [29] and MeshHoG [28] are among the local 3D descriptors that encode a signature of histograms thus being *hybrids*.

2.3.3.1 Signature of Histograms of Orientations (SHOT)

Signature of Histograms of Orientations (SHOT) was introduced by Tombari et al. [3, 21]. For extracting a SHOT descriptor, first, a robust, unique and repeatable 3D Local

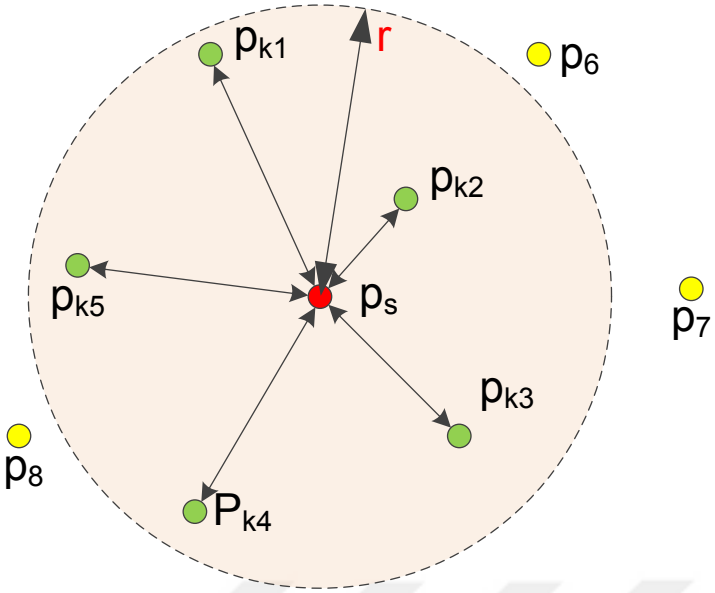


Figure 2.5: Influence region diagram for Simplified PFH.

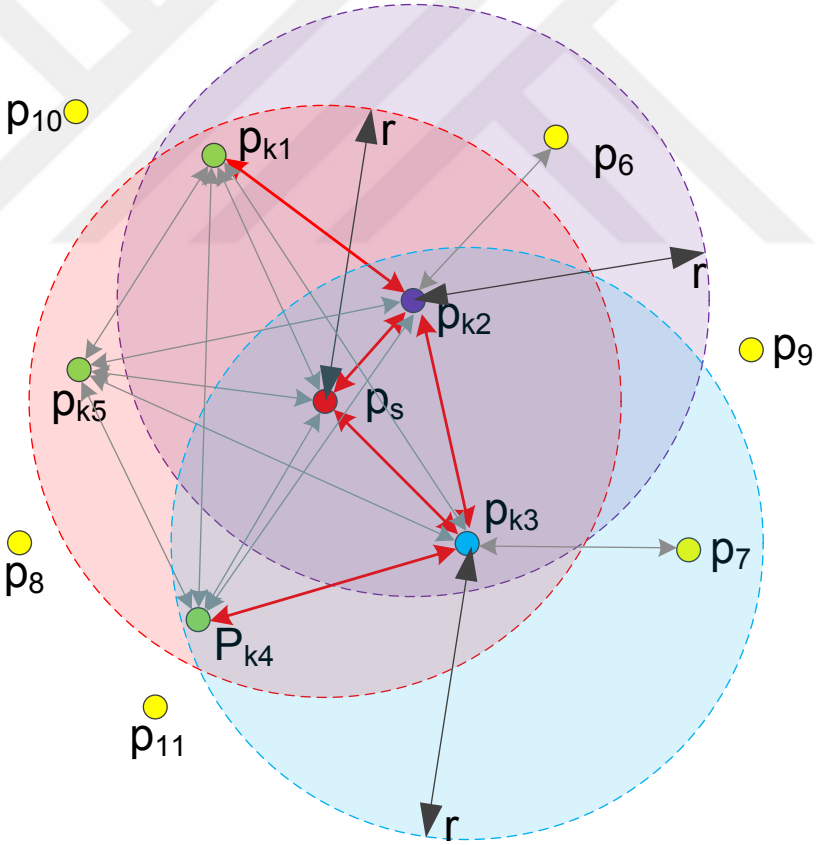


Figure 2.6: Influence region diagram for FPFH.

Reference Frame (LRF) is calculated around the source/query point. Then, a spherical grid that consists 32 volume segments (eight divisions along the azimuth, two along the elevation, and two along the radius) is centered at the point. For each of these volume segments, histogram of the angle between the normal of the source/query point and the points inside the segment is calculated. Finally, all the 32 histograms are concatenated to create the descriptor. SHOT descriptors have been shown to be rotation invariant and robust to noise [3, 21].

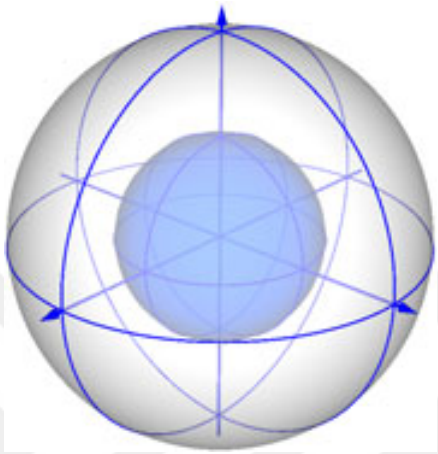


Figure 2.7: SHOT support structure (source: [3]).

2.3.3.2 Color-SHOT (CSHOT)

Color-SHOT (CSHOT) combines shape information extracted by SHOT with a texture signature [29] in order to incorporate the color information. To extract texture, the $L_1 - norm$ of the color triplets are binned into histograms. For this purpose, CIELab color space was chosen over RGB since it is perceptually more uniform. CSHOT has been reported to perform better than SHOT due to the supplementary color information [3, 16].

CHAPTER 3

PROPOSED DESCRIPTORS: SPAIR AND COSPAIR

Among the aforementioned 3D descriptors in Chapter 2, only a few utilize shape and texture/color information jointly to take advantage of the data obtained from the RGB-D sensors; the MeshHOG proposed by Zaharescu et al. [28], the colored version of the Point Feature Histograms (PFH), called PFHRGB [19, 24], and the color/texture enhanced version of Signature of Histograms of Orientations (SHOT), called CSHOT, proposed by Tombari et al. [29].

Additionally, recently, a comparative evaluation of 3D descriptors that are available in Point Cloud Library (PCL) [19] has been presented by Alexandre [16]. According to this analysis, CSHOT [29] and PFHRGB [19] which use color information in addition to shape, are the best performing descriptors, followed by the shape-only SHOT [3, 21], PFH [24] and FPFH [16]. It was also shown that PFHRGB and CSHOT are the best performing descriptors for object recognition using RGB-D data [3]. Another important point is that, in instance-level object recognition there is significant room for improvement.

Thus, to further improve recognition performance in computer/robot vision tasks, in this thesis, two new descriptors are proposed. The first one utilizes only shape information and is called *Histograms of Spatial Concentric Surflet-Pairs*, whereas the second one utilizes shape and color information jointly and is called *Colored Histograms of Spatial Concentric Surflet-Pairs*.

3.1 Histograms of Spatial Concentric Surflet-Pairs (SPAIR)

Histograms of Spatial Concentric Surflet-Pairs (SPAIR) is based on *surflet-pair-relations* similar to PFH and FPFH where a *surflet* is defined as an oriented surface point and *surflet-pair-relations* as geometric relations between two *surflets* by Wahl et al. [41].

As described in Section 2.3.2.3, Rusu et al. used a method called *Simplified Point Feature Histogram* (SPFH) that relies on the comparison of source/query point/surflet with only the direct *k*-neighbors (not all the pairs) inside a spherical support. Furthermore, in order to add spatial information, a special weighting scheme was used in FPFH as formulated in Equation 2.2.

With SPAIR, we aimed for a simpler thus faster method which requires fewer number of point-pair comparisons while adding more spatial information by encoding the geometrical properties of a point's neighborhood according to distance from the point.

As shown in Figure 3.1, in our approach, the support radius r is divided into N equal size $(r_1, r_2, ..., r_N)$ regions. The resulting 3D grid can be visualized as N concentric spheres. For each distinct spherical shell (i.e., the region between two adjacent spheres), which we name as a *level* $(L_1, L_2, ..., L_N)$, the surflet-pair-relations between the points inside a level and the source/query point (see Figure 3.2) are calculated as follows [23, 41]:

- Let $\mathbf{p}_s$ be the source/query point that SPAIR is to be extracted for, $\mathbf{p}_t$ be one of the target points inside a *level* and $\vec{n}_s$, $\vec{n}_t$ the corresponding normals.

- A fixed reference coordinate uvw frame is defined as shown in Figure 3.3, following [4]:

$$\vec{u} = \vec{n}_s, \tag{3.1}$$

$$\vec{v} = (\mathbf{p}_t - \mathbf{p}_s) \times \vec{u}, \tag{3.2}$$

$$\vec{w} = \vec{u} \times \vec{v}. \tag{3.3}$$

- Using the reference frame defined above, the angular relations between surflets

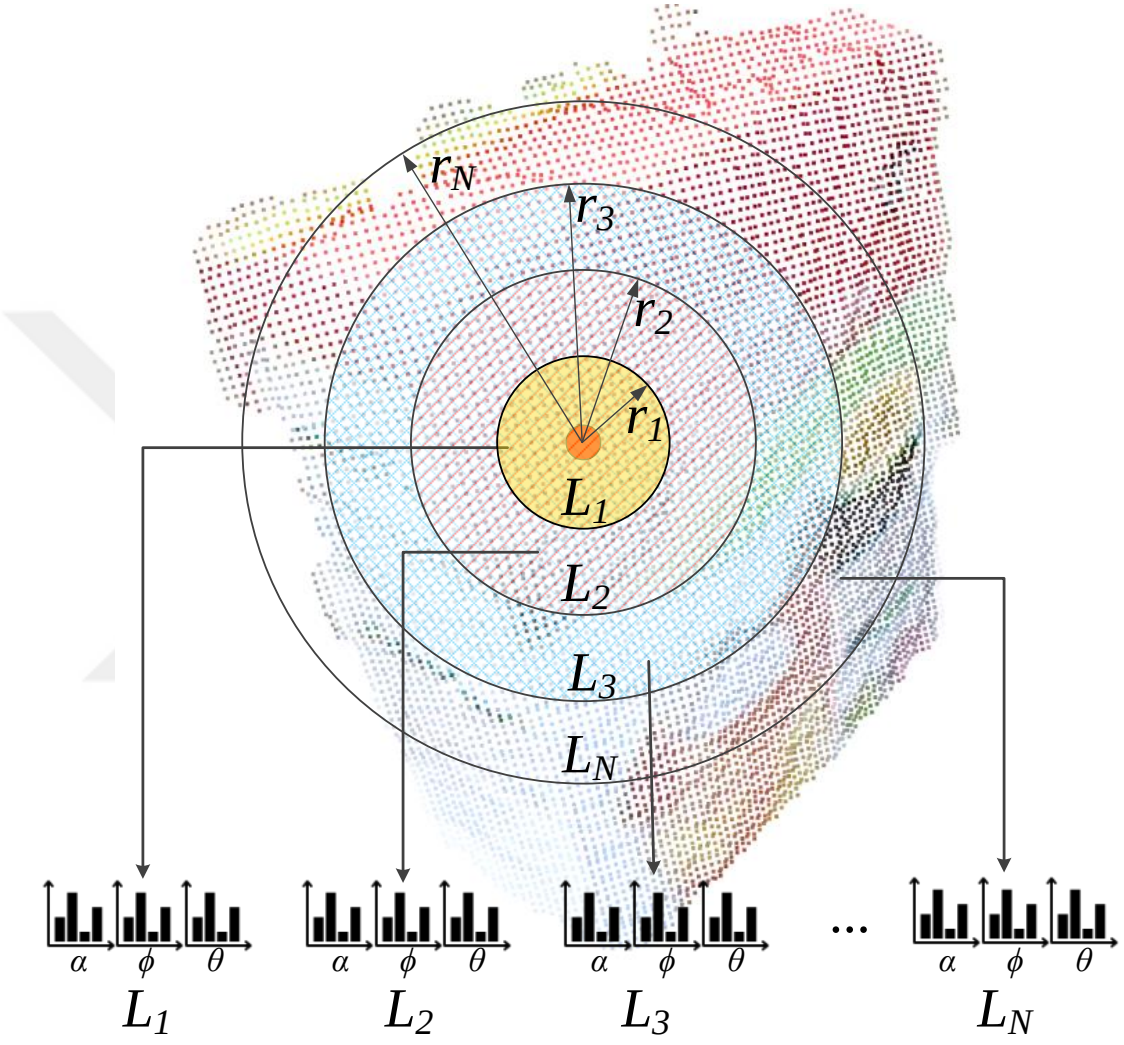


Figure 3.1: Concentric spherical regions and stitching of the histograms to construct SPAIR descriptor.

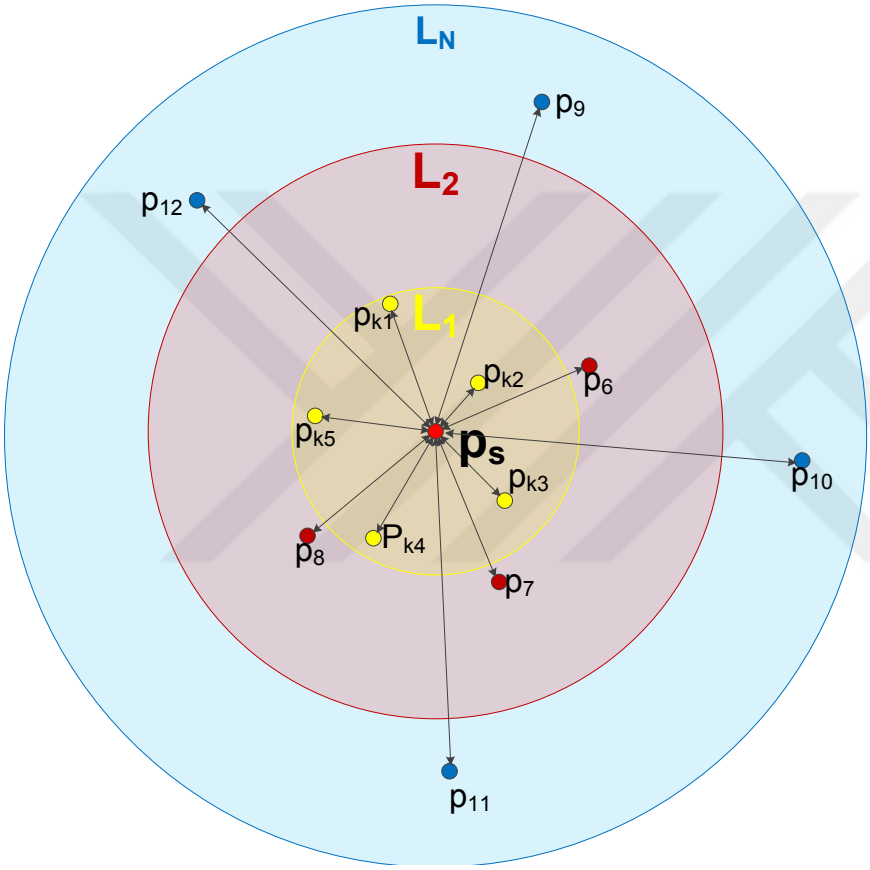


Figure 3.2: Influence region diagram for SPAIR/CoSPAIR.

are calculated as follows:

$$\alpha = \vec{v} \cdot \vec{n}_t, \tag{3.4}$$

$$\phi = \frac{\vec{u} \cdot (\mathbf{p}_t - \mathbf{p}_s)}{\|\mathbf{p}_t - \mathbf{p}_s\|}, \tag{3.5}$$

$$\theta = \arctan\left(\vec{w} \cdot \vec{n}_t, \vec{u} \cdot \vec{n}_t\right), \tag{3.6}$$

where $\alpha \in [-1, 1]$ represents $\vec{n}_t$ as the cosine of a polar angle, $\phi \in [-1, 1]$ is the direction of the translation from $\mathbf{p}_s$ to $\mathbf{p}_t$, $\theta \in [-\pi, \pi]$ corresponds to $\vec{n}_t$ as an azimuthal angle.

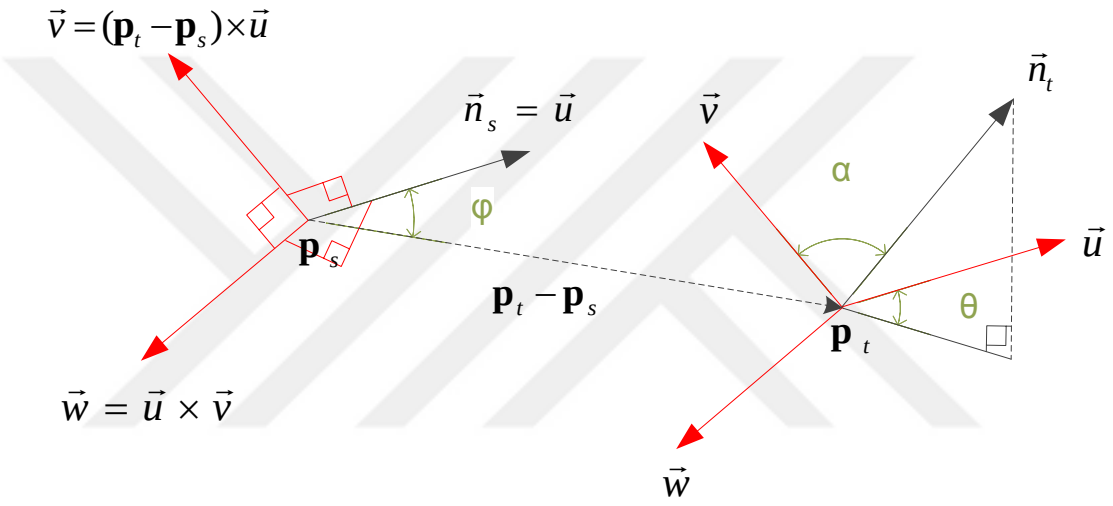


Figure 3.3: The reference coordinate uvw frame and the angular relations between surflets (adapted from [4]).

Then, the three values for the angles (α, ϕ, θ) in Equations 3.4, 3.5, 3.6 are binned into separate histograms:

$$H_\alpha^l(b) = \sum_{\mathbf{p}_t} \delta\left(\left\lfloor \frac{1}{2}\alpha(\mathbf{p}_t, \mathbf{p}_s)B \right\rfloor - b\right), \tag{3.7}$$

$$H_\phi^l(b) = \sum_{\mathbf{p}_t} \delta\left(\left\lfloor \frac{1}{2}\phi(\mathbf{p}_t, \mathbf{p}_s)B \right\rfloor - b\right), \tag{3.8}$$

$$H_\theta^l(b) = \sum_{\mathbf{p}_t} \delta\left(\left\lfloor \frac{1}{2\pi}\theta(\mathbf{p}_t, \mathbf{p}_s)B \right\rfloor - b\right), \tag{3.9}$$

where l is the level for which the histogram is being computed, $\delta()$ is the Kronecker delta function, b is the bin index of a histogram, and B is the total number of bins.

When calculations are finalized for all the defined surflet-pairs, the histograms H_α^l, H_ϕ^l and H_θ^l are normalized using the number of distinct points in each *level*:

$$\hat{H}_\alpha^l(b) = \frac{1}{C^l} H_\alpha^l(b), \tag{3.10}$$

$$\hat{H}_\phi^l(b) = \frac{1}{C^l} H_\phi^l(b), \tag{3.11}$$

$$\hat{H}_\theta^l(b) = \frac{1}{C^l} H_\theta^l(b), \tag{3.12}$$

where C^l is the number of points in *level* l.

The resulting SPAIR descriptor $\mathbf{v}_{SPAIR}$ is the concatenation of all the histograms in an order based on their distances to the center point:

$$\mathbf{v}_{SPAIR} = \hat{H}_\alpha^0 \oplus \hat{H}_\phi^0 \oplus \hat{H}_\theta^0 \oplus ... \hat{H}_\alpha^N \oplus \hat{H}_\phi^N \oplus \hat{H}_\theta^N, \tag{3.13}$$

where $\oplus$ denotes concatenation. Figure 3.1 illustrates the levels inside the concentric sphere borders and stitching of the corresponding histograms.

3.2 Colored Histograms of Spatial Concentric Surflet-Pairs (CoSPAIR)

It has been reported that adding color/texture information improves the performance of various descriptors considerably [3, 5, 16, 42]. With this motivation, we modified SPAIR such that it encodes color as well as shape, and called it *Colored Histograms of Spatial Concentric Surflet-Pairs* (CoSPAIR).

In CoSPAIR, color/texture and shape information is encoded at each level of the SPAIR descriptor as shown in Figure 3.4. In our experiments, three different color spaces; RGB, HSV and CIELab have been tested. Additionally, for each color space, two different algorithms have been evaluated: (i) Using simple color histogram of each color channel. (ii) Using histogram of $L_1 - norm$ of point pairs for each color channel. Our experiments (see Table 3.1) indicated that the best results are obtained by using simple color histograms in the CIELab color space for each channel at each level. This resulted in a descriptor that has 3 sub-features for both shape and color for

each *level*:

$$\begin{aligned}\mathbf{v}_{CoSPAIR} = \hat{H}^0_\alpha \oplus \hat{H}^0_\phi \oplus \hat{H}^0_\theta \oplus \hat{H}^0_{\mathbf{L}} \oplus \hat{H}^0_{\mathbf{a}} \oplus \hat{H}^0_{\mathbf{b}} \oplus ... \\ \hat{H}^N_\alpha \oplus \hat{H}^N_\phi \oplus \hat{H}^N_\theta \oplus \hat{H}^N_{\mathbf{L}} \oplus \hat{H}^N_{\mathbf{a}} \oplus \hat{H}^N_{\mathbf{b}}.\end{aligned} \tag{3.14}$$

where $\oplus$ denotes concatenation and $\mathbf{L}$, $\mathbf{a}$, $\mathbf{b}$ denotes the CIELab color components.

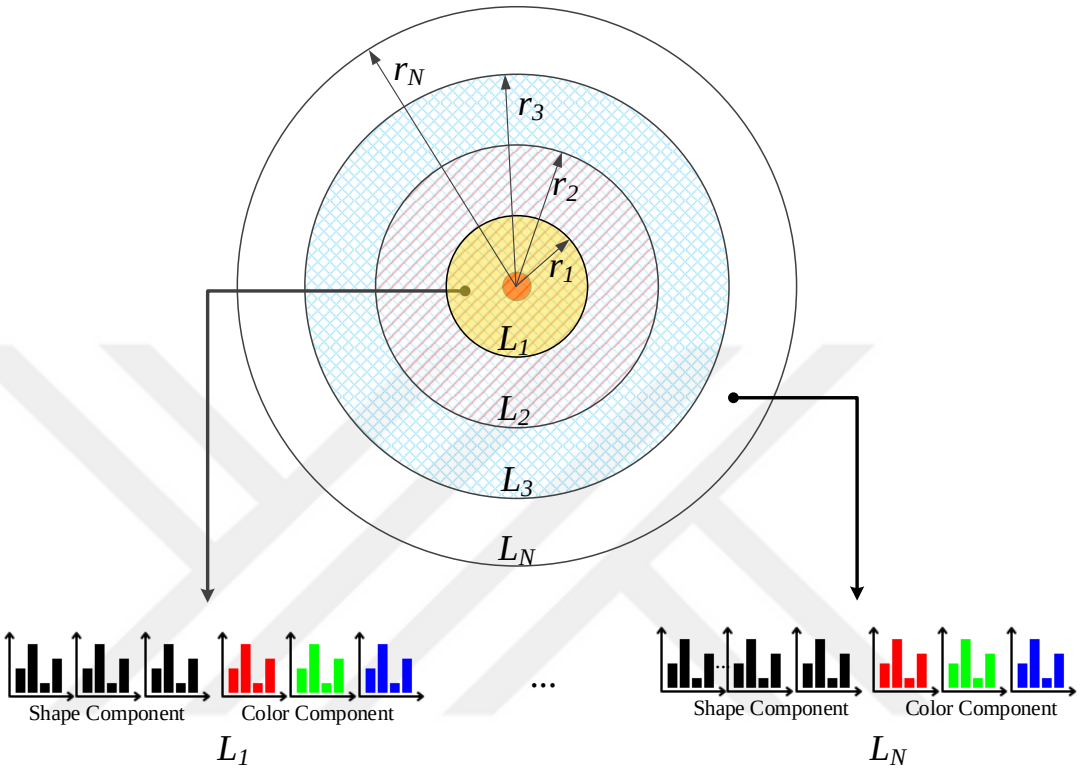
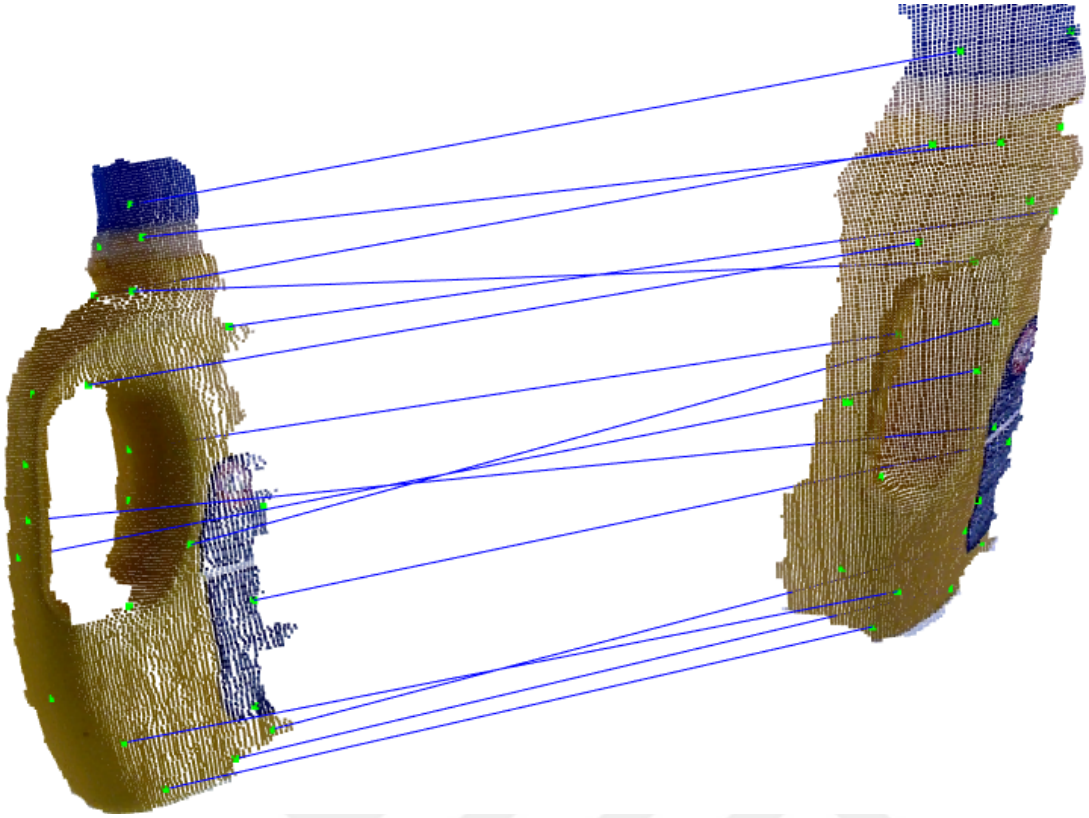
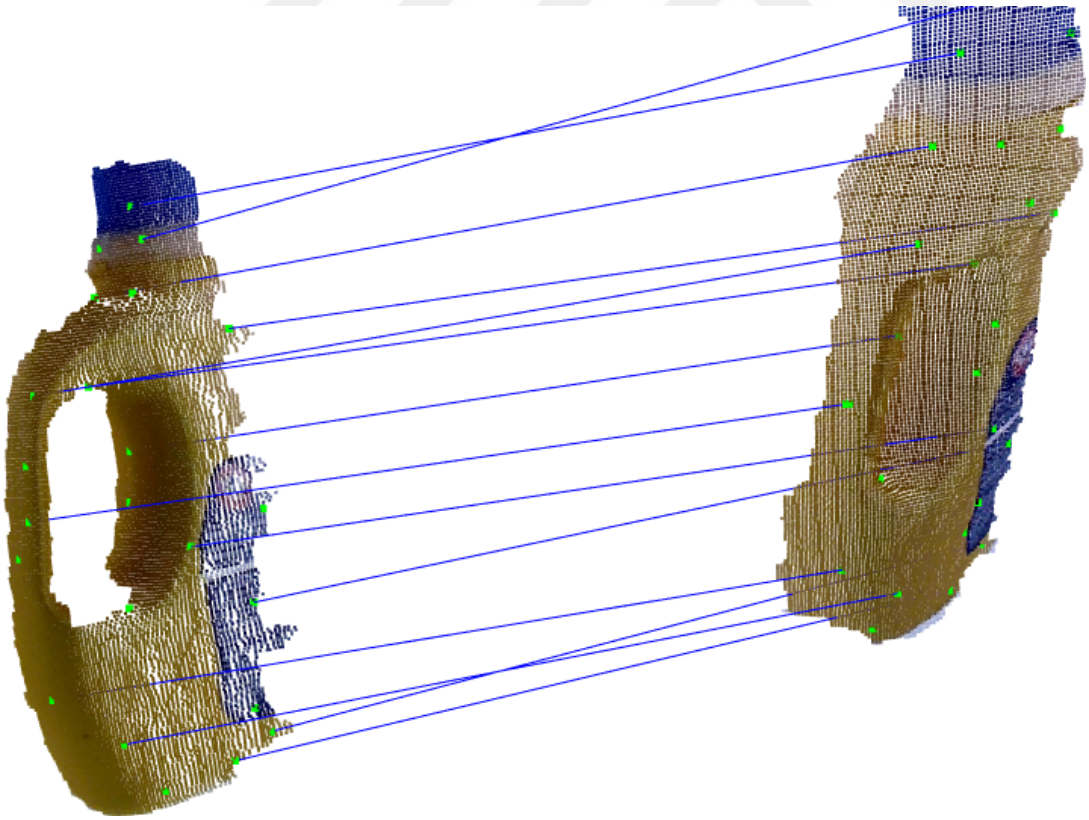


Figure 3.4: Concentric spherical regions and the stitching of shape and color histograms for the extraction of CoSPAIR.

Although the performance of the proposed descriptors will be detailed in Chapter 5, the matching success of the descriptors can be seen and compared to some of the compared descriptors (FPFH, SHOT and CSHOT) in Figure 3.5 and Figure 3.6.

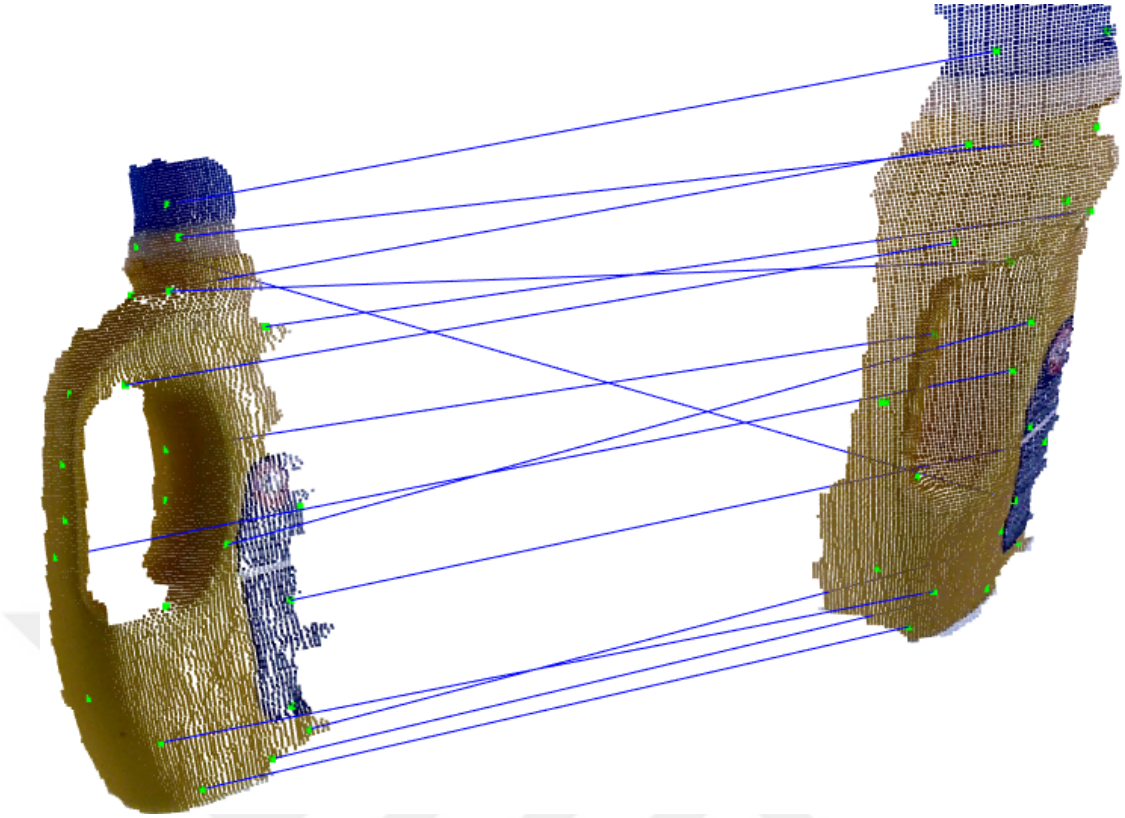


(a) CSHOT

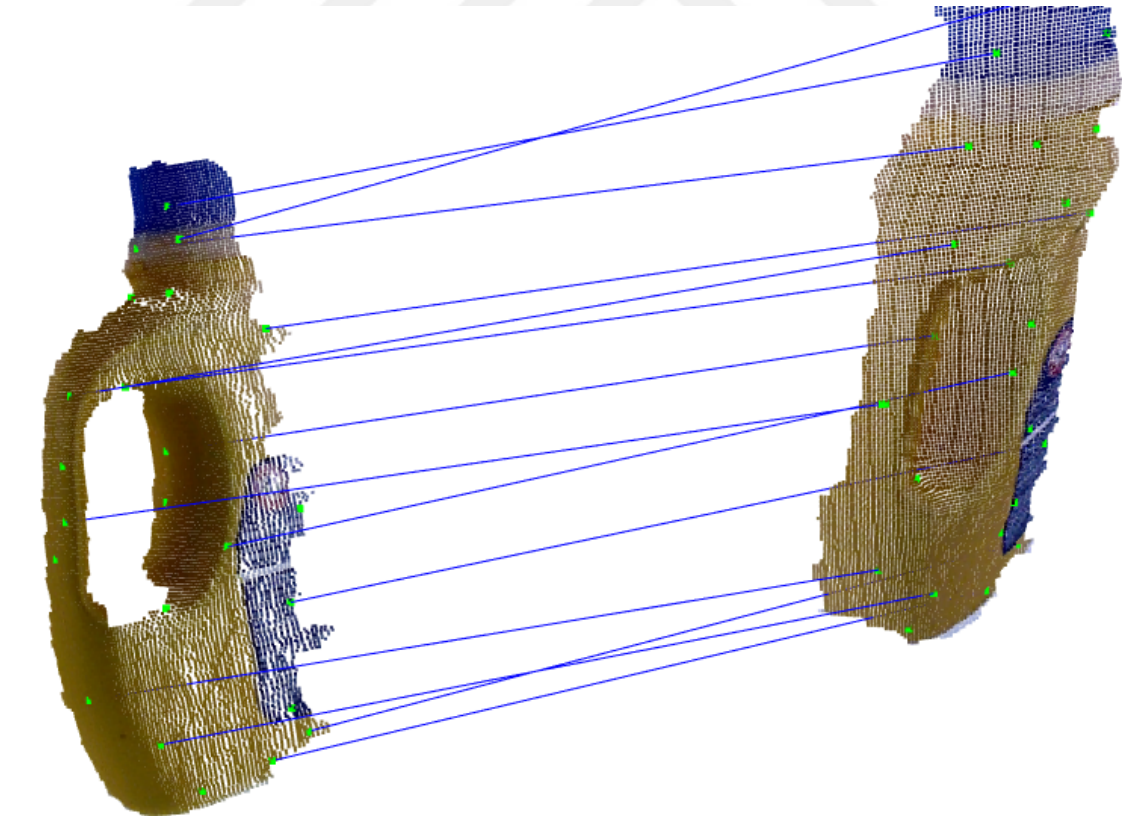


(b) CoSPAIR

Figure 3.5: Descriptor matching results - `detergent`



(c) SHOT

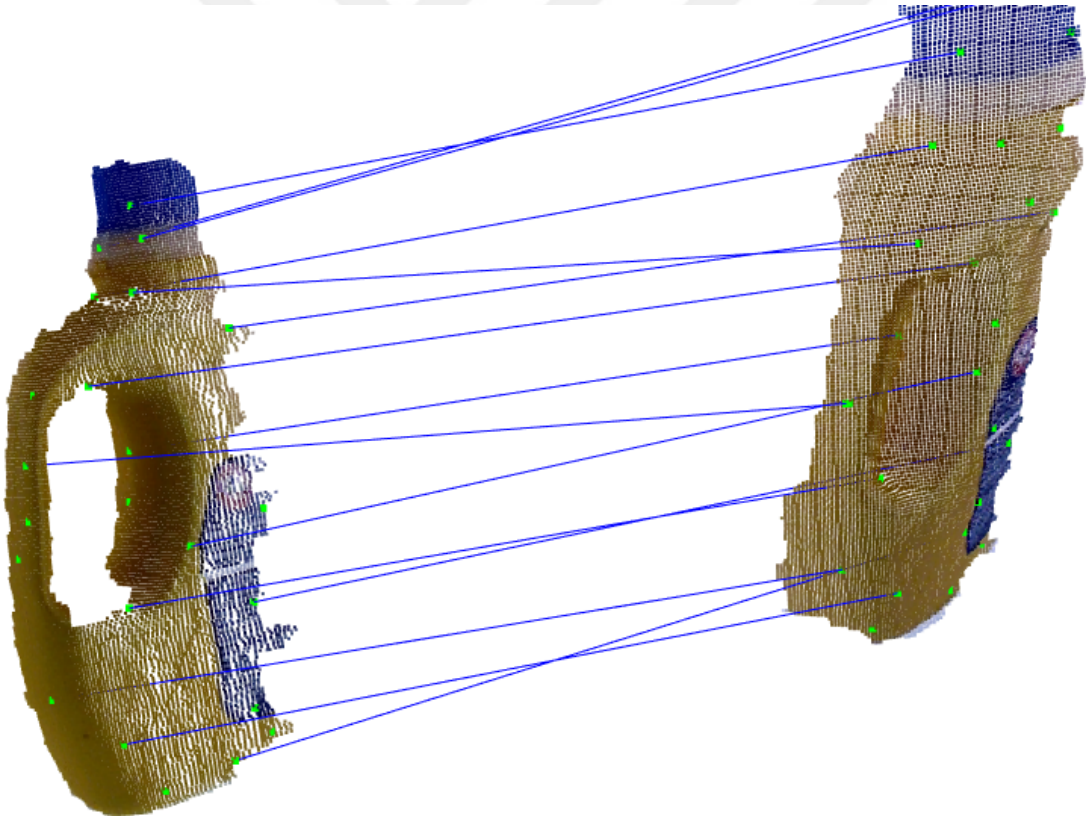


(d) SPAIR

Figure 3.5: Descriptor matching results - `detergent` (cont.)

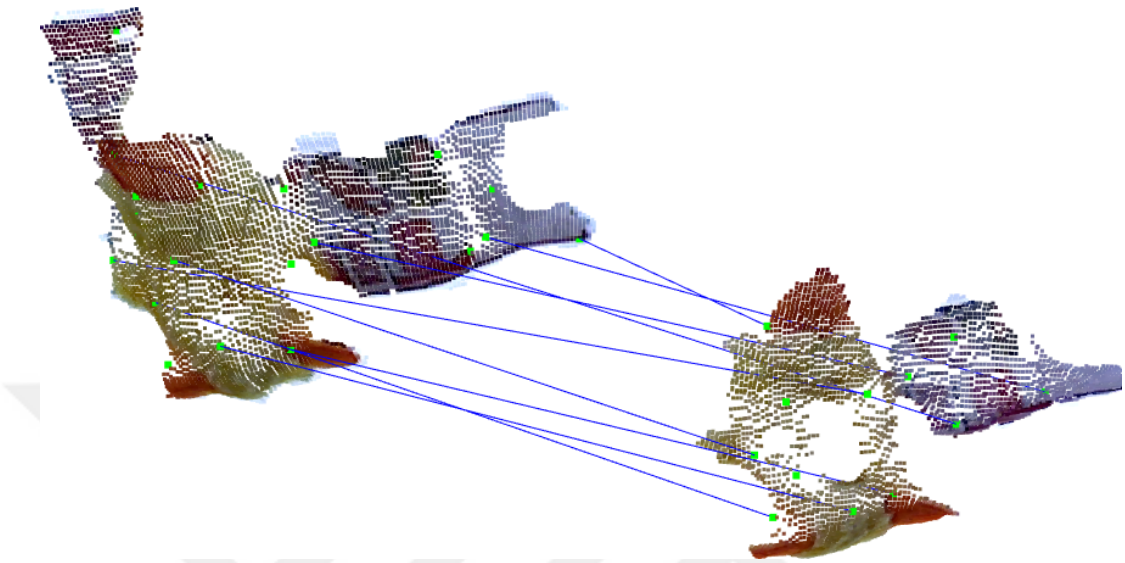
Table 3.1: Average accuracy results for different color components. The tests were conducted in Dataset 1 (see Section 5.2.1).

	Category Level	Instance Level
RGB	93.63	81.76
RGB-L_1	91.74	82.64
HSV	91.40	76.31
HSV-L_1	86.46	64.61
CIELab	**94.34**	**83.10**
CIELab-L_1	86.25	64.23

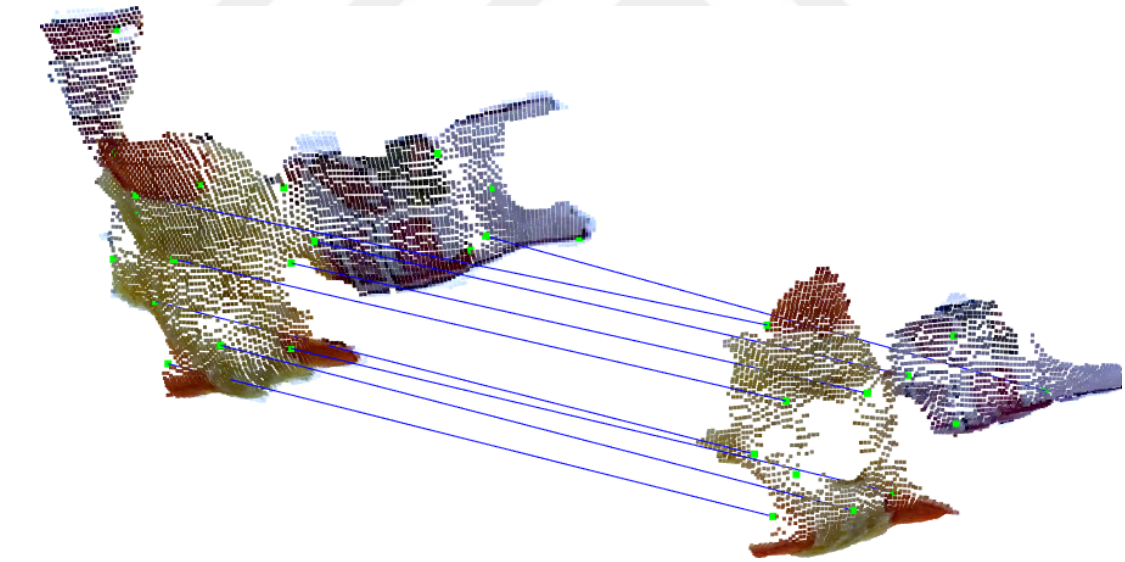


(e) FPFH

Figure 3.5: Descriptor matching results - `detergent` (cont.)

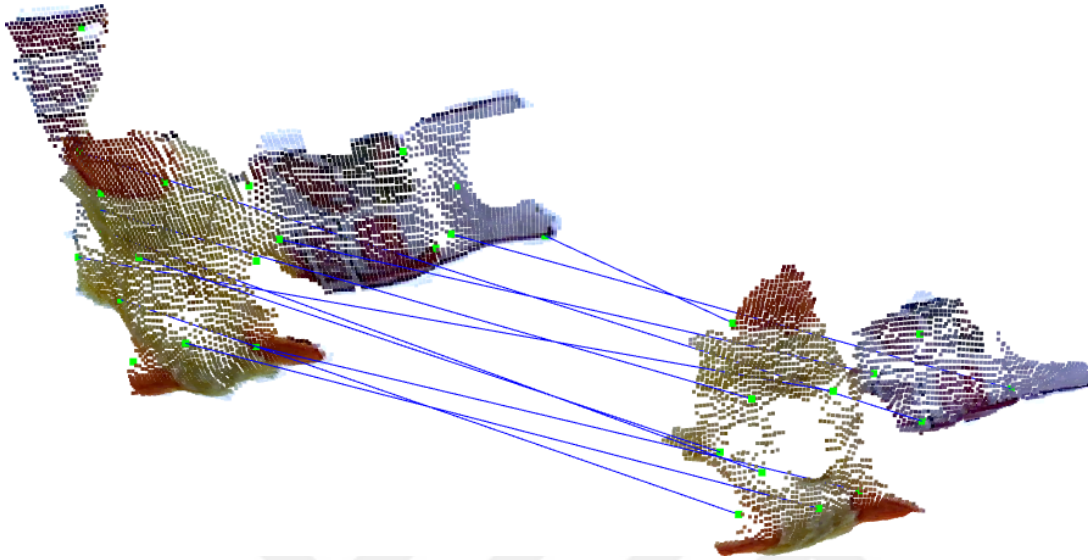


(a) CSHOT

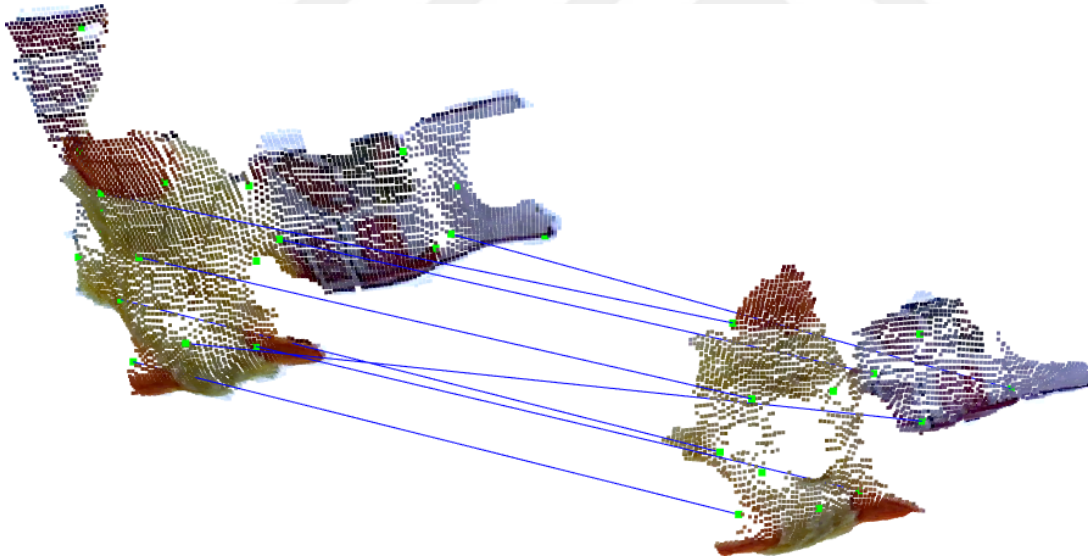


(b) CoSPAIR

Figure 3.6: Descriptor matching results - `kong duck dog toy`

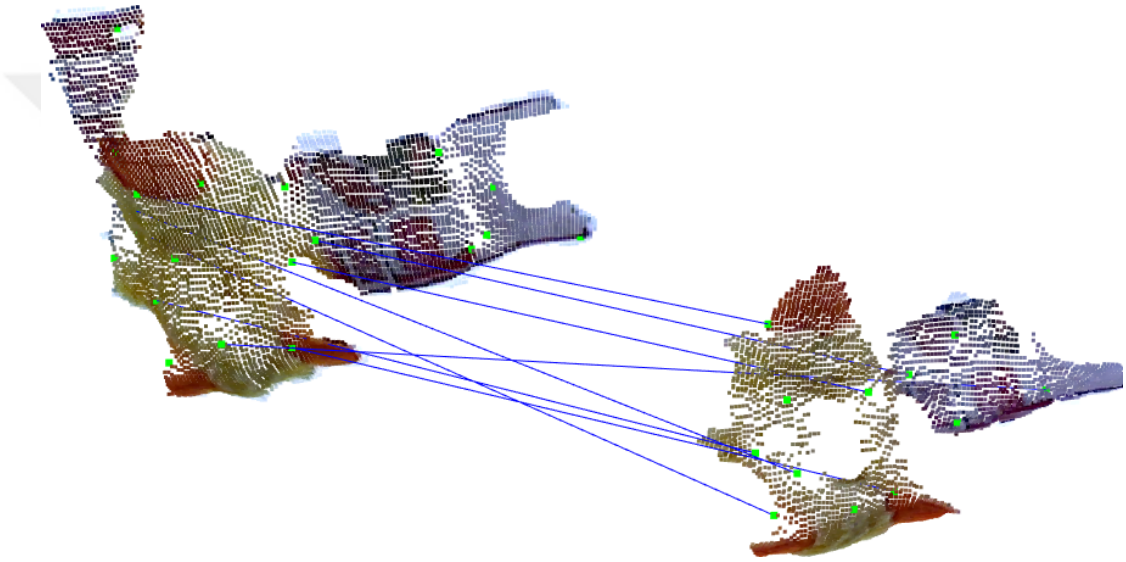


(c) SHOT



(d) SPAIR

Figure 3.6: Descriptor matching results- `kong duck dog toy` (cont.)



(e) FPFH

Figure 3.6: Descriptor matching results - `kong duck dog toy` (cont.)

CHAPTER 4

DESCRIPTOR EXTRACTION FLOW

The steps for extraction the SPAIR and CoSPAIR descriptors are same and given in Figure 4.1. The first three steps also apply to the descriptors that are compared in Chapter 5, therefore detailed below.

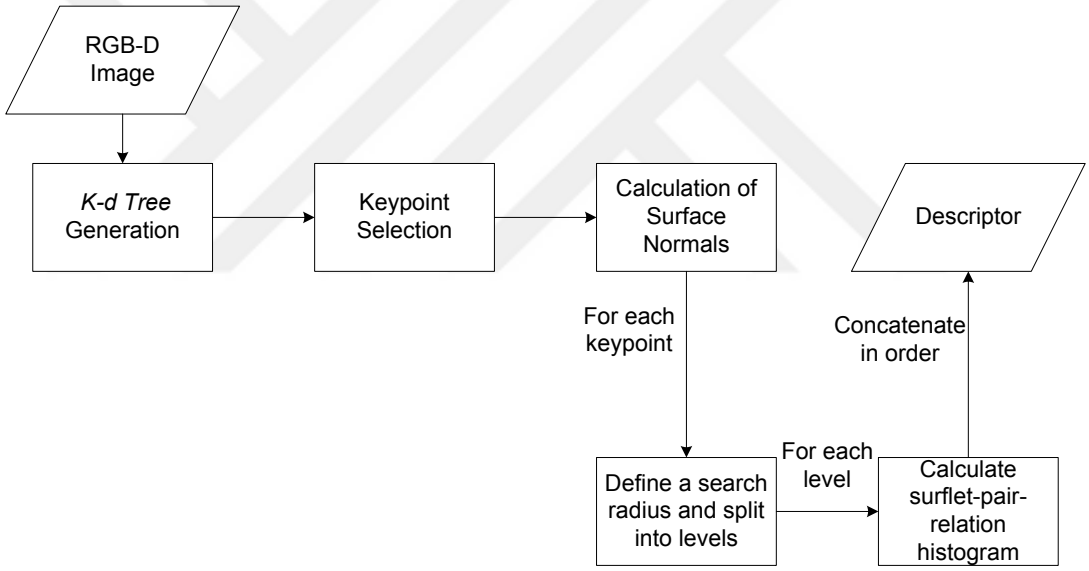


Figure 4.1: Extraction flow of SPAIR / CoSPAIR.

4.1 Spatial decomposition with K-d trees

The proposed as well as compared local 3D descriptors need to access a number of neighboring points P^k to understand and represent the geometry around a query point p_q. Thus, one needs algorithms to search P^k as fast as possible, without re-computing distances between each point every time. Spatial decomposition techniques such as

k-d tree (k-dimensional tree) [43] or *octree* [44] are solution to such problems. These techniques partition the point cloud data P into piles, such that searching and indexing of the points in P can be accomplished quickly and efficiently.

In general, there are two use cases for the determination of P^k for a query point p_q [45]:

1. Query the closest k neighbors of p_q (*k search*)
2. Query the k neighbors of the p_q within a radius r (*radius search*)

For these tasks, within the context of this thesis, *k-d tree* method is used for spatial decomposition of the point clouds.

The *k-d tree* method is introduced by Jon Bentley in 1975 [43]. Although it is a fairly old algorithm and there exist many more spatial decomposition algorithms in literature, *k-d tree* and its variants remain probably the most popular. It is in general a binary search tree (BST) that stores points in k-dimensional space. *K-d tree*s recursively and hierarchically decompose a region of space, creating a binary space partition at each level of the tree.

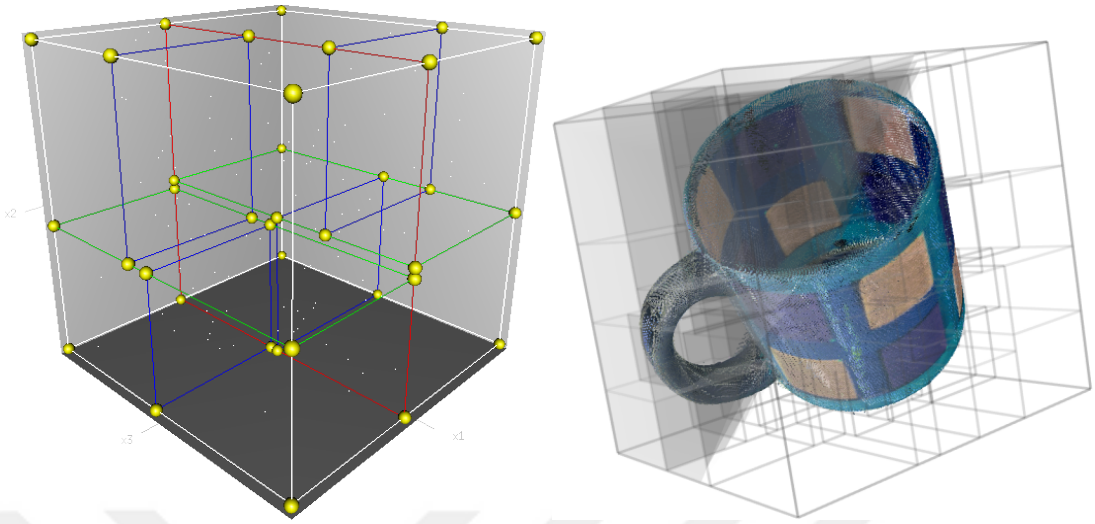
As an example, Figure 4.2 shows a 3D space partitioned by a 3D *k-d tree*.

The most known method to construct a *k-d tree* is as follow:

- The point is represented by the set of nodes in the *k-d tree*.
- Divide the points in two in half. All the points in the "right" subspace are represented by the right subtree and the points in the "left" subspace by the left subtree.
- Recursively construct *k-d tree*s for the two sets of points (cycle through the axes used to select the splitting planes in *round-robin* fashion).

In Figure 4.3 a set of points in 2D space and the related constructed *k-d tree* is shown.

Note that, *k-d tree*s are known to be inefficient as the number of dimensions increase above three [46].



(a) 3D k-d tree (image from Wikimedia Commons)

(b) 3D k-d tree on a mug (image source: [17])

Figure 4.2: Partioning of 3D space with 3D *k-d tree*s.

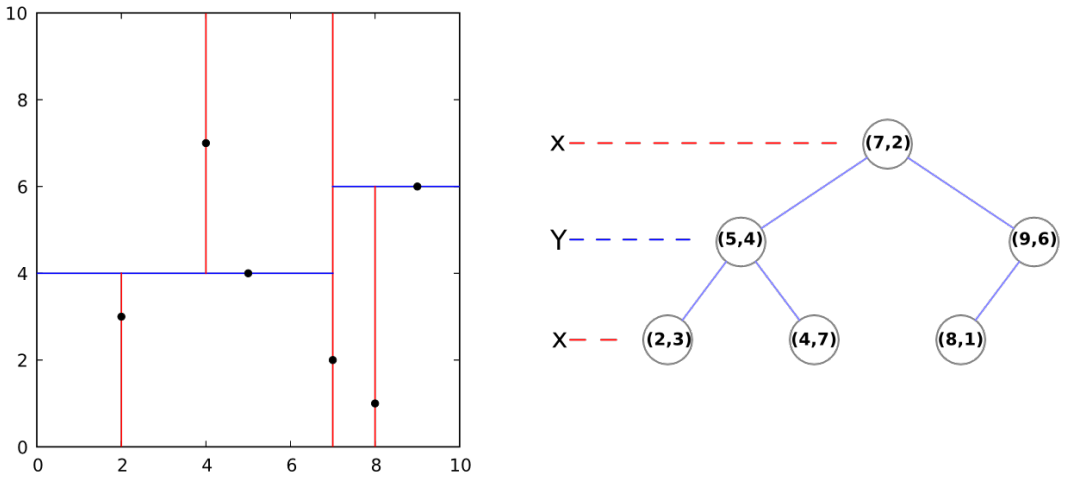


Figure 4.3: Example construction of a 2D *k-d tree* (image from Wikimedia Commons).

4.1.1 Nearest Neighbor Search in K-d Trees

To find the k points in the tree that are nearest to a given input point q, the nearest neighbor search (NN) algorithm is used. The NN search can eliminate significant chunks of the search space via the hierarchical subdivision structure of the tree yielding an efficient search. The NN search on k-d trees is performed in two stages via a backtracking, branch-and-bound search:

- In the first stage, the tree is traversed from top to bottom to find the bin (d dimensional) that contains the query point q. Then, the distances between q and the points in the bin are calculated for an initial approximation of the nearest neighbor.
- In the backtracking stage, the tree is traversed from bottom to top searching for potential points that are closer to q than current best.

For low-dimensional spaces, this process can be effective since small-number of leaf visits is usually enough. However, for higher dimensions the performance can degrade significantly. In order to reduce memory usage and increase speed in high-dimensional cases, approximate nearest neighbor (ANN) algorithms are used in practice. However, this type of algorithms don't ensure to access the exact nearest neighbor every time.

4.2 Normal Estimation

Estimating surface normals is one of the most crucial steps of many object recognition tasks as well as many computer graphics applications. There exists many methods for estimating surface normals. The existing methods are analyzed and compared by Klasing et al. for 3D point clouds [47]. In the work, the existing methods are divided into two as *optimization-based* and *averaging*. After a detailed analysis and comparison, the method that is dubbed as *PlanePCA* is stated to be superior in performance in terms of both quality and speed.

In this thesis, the surface estimation method that is implemented in the PCL library

is used. The method is developed by Rusu and is detailed in [45]. It is one of the simplest methods and is based on the first order 3D plane fitting where determining the normal to a point on the surface is approximated by estimating the normal of a plane tangent to the surface, thus leading to a least-square plane fitting estimation problem. Therefore, the solution for estimating the surface normal is reduced to analysis of the eigenvectors and eigenvalues of a covariance matrix created from the k neighborhood (P^k) of the query point p_q. For each point $p_i \epsilon P^k$, the covariance matrix C is assembled as follows:

$$C = \frac{1}{k} \sum_{i=1}^{k} \cdot (p_i - \overline{p}) \cdot (p_i - \overline{p})^T , \tag{4.1}$$

$$C \cdot \vec{v_j} = \lambda_j \cdot \vec{v_j}, j \epsilon \{0, 1, 2\} , \tag{4.2}$$

where k is the number of point neighbors considered in the neighborhood of p_i, $\overline{p}$ represents the 3D centroid of the nearest neighbors, λ_j is the j-th eigenvalue of the covariance matrix, and $\vec{v_j}$ the j-th eigenvector [45].

In general, the orientation of the normal $\vec{n}$ computed with the above method is ambiguous since there is no mathematical way to solve the sign of it. This may lead non-consistent orientation of normals over an entire point cloud dataset. However, the solution to this problem is trivial if the viewpoint V_p is known; which is the case for Kinect like 2.5D cameras that are used in this thesis. To orient all normals $\vec{n}_i$ consistently towards the viewpoint V_p, they should satisfy the equation [45]:

$$\vec{n}_i \cdot (V_p - p_i) > 0. \tag{4.3}$$

The outcome of the explained normal estimation method is given in Figure 4.4 where the estimated normals are shown as black lines.

4.3 Keypoint Selection

Due to the computational complexity to extract 3D features, to prevent excessive amount of time that is required to extract them from each point in a cloud, they should be extracted from a smaller set of points. To achieve this, algorithms which detect *keypoints* i.e. interest points that stand out are used. A proper 3D keypoint detection

(a) Red Bull (b) Nutrigrain Cherry

Figure 4.4: Estimated normals for various objects (support radius = 1 cm).

method should extract repeatable keypoints under viewpoint changes, missing parts, point density or topology variations, clutter and sensor noise [48].

Although there are many 3D keypoint detection methods exist in literature, *Intrinsic Shape Signatures* (ISS3D) has been reported to stand out for its performance, repeatability and efficiency [48, 49, 50]. Therefore, in this thesis, it is used as the primary keypoint detection method in our experiments. Additionally, to observe the effect of keypoint detectors on the performance of descriptors (Section 5.4), Harris3D and *uniform sampling* methods are used, hence explained further below.

4.3.1 Intrinsic Shape Signatures

Intrinsic Shape Signatures (ISS) is introduced by Zhong in 2009 [25]. ISS, $S_i = F_i, f_i$ at a point p_i consists of two components; the intrinsic reference frame (F_i) and the 3D shape feature vector (f_i).

ISS is based on Eigenvalue Decomposition (EVD) of the weighted scatter matrix ($cov(p_i)$) of the points within a point p's support. It possesses two significant traits:

- To include only points with large variations along each principal direction, it uses the magnitude of the smallest eigenvalue.
- To avoid detecting keypoints at points that show a similar dissemination along the principal directions, it uses the ratio between two consecutive eigenvalues.

The scatter matrix within a distance r_frame is computed as:

$$cov(p_i) = \sum_{|p_j - p_i| < r_{frame}} w_j (p_j - p_i)(p_j - p_i)^T / \sum_{|p_j - p_i| < r_{frame}} w_j, \tag{4.4}$$

where

$$w_i = 1/\|p_j : |p_j - p_i| < r_{frame}\|. \tag{4.5}$$

Then, the scatter matrix's eigenvalues $\lambda_i^1, \lambda_i^2, \lambda_i^3$ are computed in the order of decreasing magnitude together with their eigenvectors e_i^1, e_i^2, e_i^3.

During the elimination stage, points whose ratio between two consecutive eigenvalues is below a threshold are kept:

$$\frac{\lambda_2(p)}{\lambda_1(p)} < Th_{12} \wedge \frac{\lambda_3(p)}{\lambda_2(p)} < Th_{23}, \tag{4.6}$$

to avoid detecting keypoints at points that show a similar dissemination along the principal directions.

And lastly, to include only points with large variations along each principal direction, among remaining points, the saliency is determined by the magnitude of the smallest eigenvalue:

$$p_i \doteq \lambda_i^3. \tag{4.7}$$

4.3.2 Harris3D

The original Harris method that is introduced by Harris and Stephens in 1988 is a corner and edge based method [51]. The algorithm uses pixel gradients and their changes in the horizontal and vertical directions.

For 3D domain, in PCL [19], the algorithm is adjusted to work with surface normals. It replaces the pixel intensity gradients in the covariance matrix (Cov) by surface normals but uses the same response function. The keypoints response (r) measured at each point is then defined by [49, 50]:

$$r(x, y, z) = det\left(Cov\left(x, y, z\right)\right) - k\left(trace\left(Cov\left(x, y, z\right)\right)\right)^2, \tag{4.8}$$

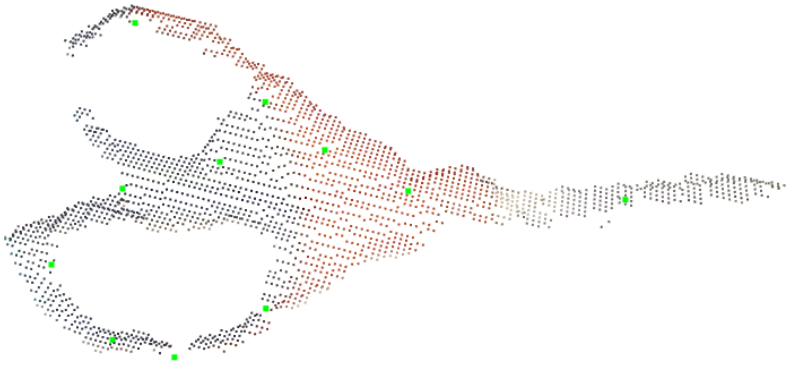
where k is a positive real valued parameter that works as a lower bound for the ratio between the magnitude of the weaker edge and the stronger edge. In addition, to prevent detecting too many keypoints that pile closely, a non-maximal suppression on the keypoints response is (usually) carried out to suppress weak keypoints around the stronger ones [50].

4.3.3 Uniform Sampling

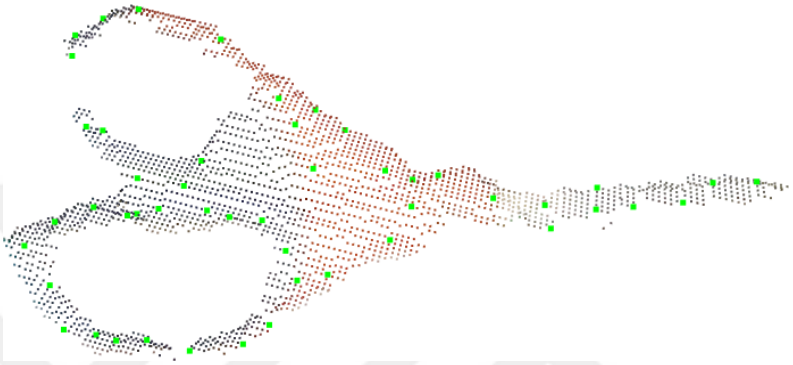
Uniform sampling is in fact not a keypoint detection method, but rather used for selecting a subset of points of a point cloud. This method is used for observing the effect of keypoint selection algorithms on the performance of descriptors in Section 5.4.

In this thesis, the *uniform sampling* algorithm that is implemented in PCL is used. The algorithm creates a 3D voxel grid over the input point cloud data and then, in each voxel all the points present are approximated (i.e., down-sampled) with their centroid.

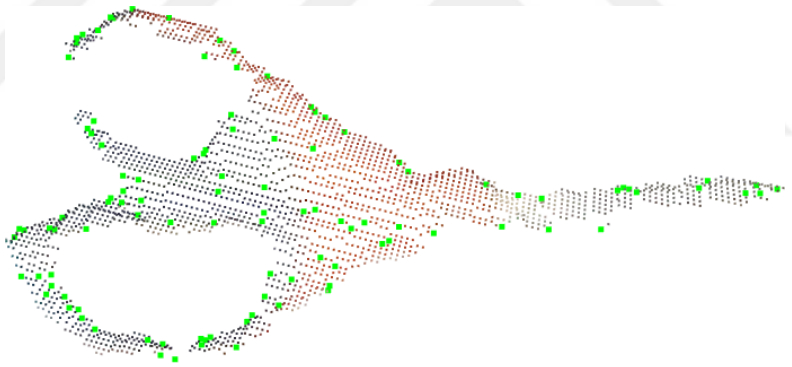
The results of the explained keypoint selection methods, ISS3D, Harris3D and *uniform sampling* for leaf size of 1 cm and 2 cm are given in Figures 4.5, 4.6 and 4.7.



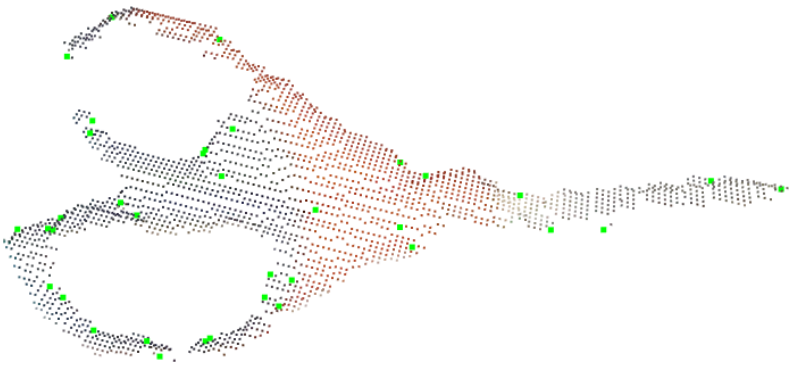
(a) Harris3D



(b) ISS

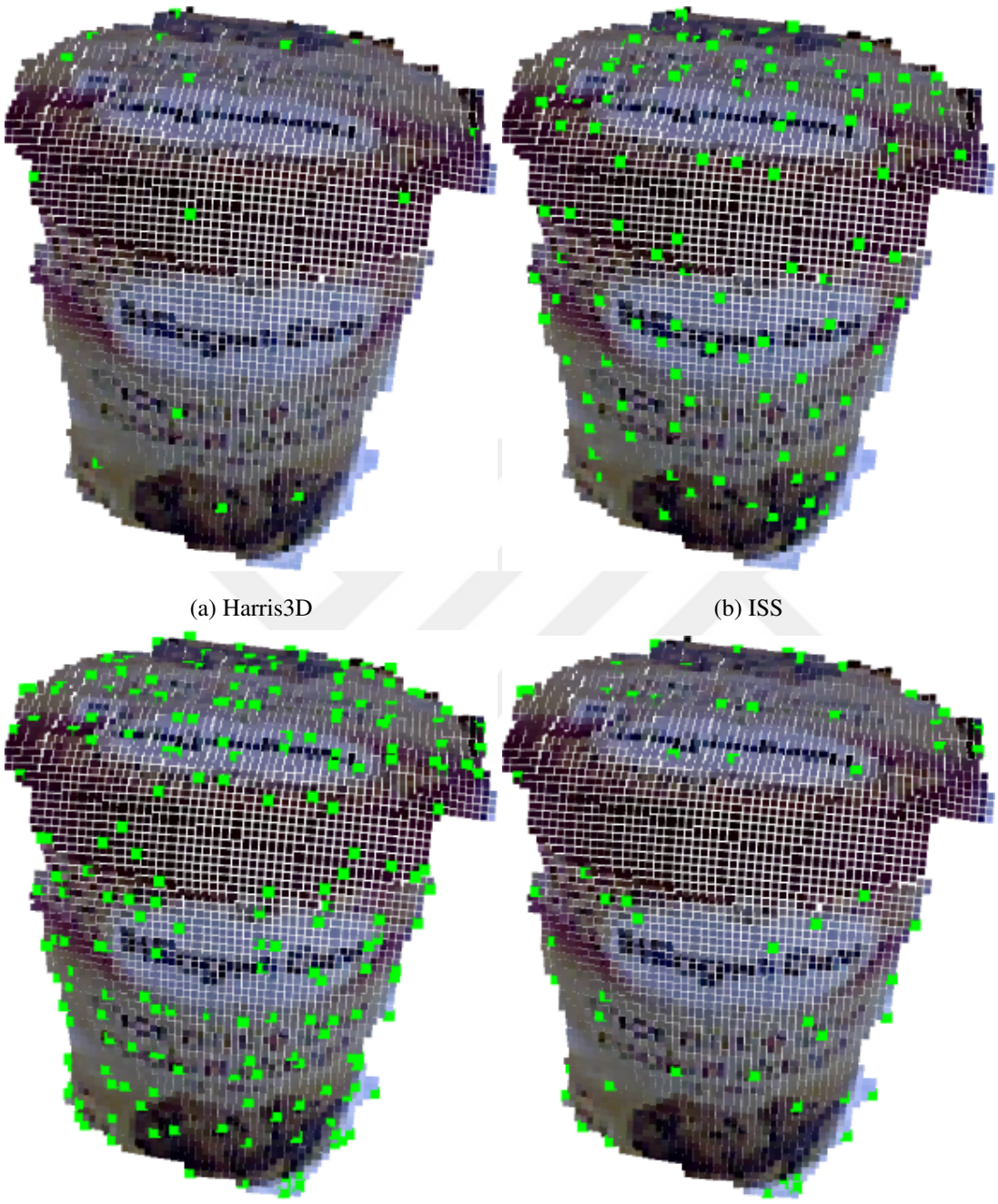


(c) Uniform sampling - 1 cm



(d) Uniform sampling - 2 cm

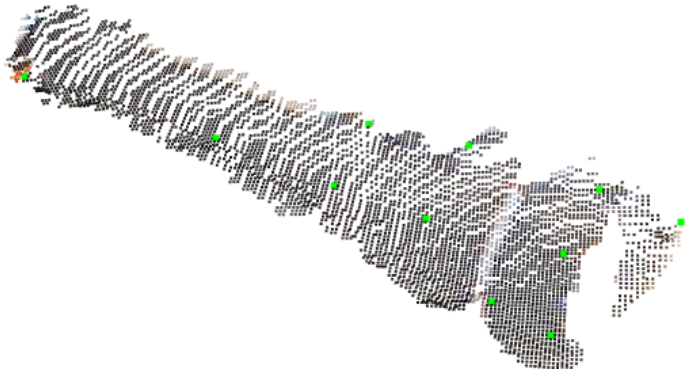
Figure 4.5: Results of various keypoint detection methods for `scissors 1`.



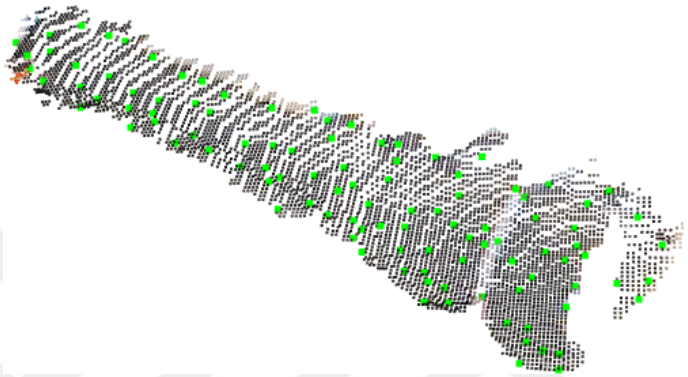
(a) Harris3D (b) ISS

(c) Uniform sampling - 1 cm (d) Uniform sampling - 2 cm

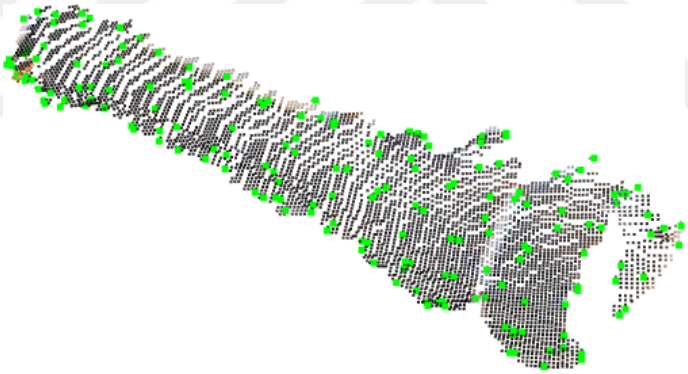
Figure 4.6: Results of various keypoint detection methods for `haagen dazs cookie dough`.



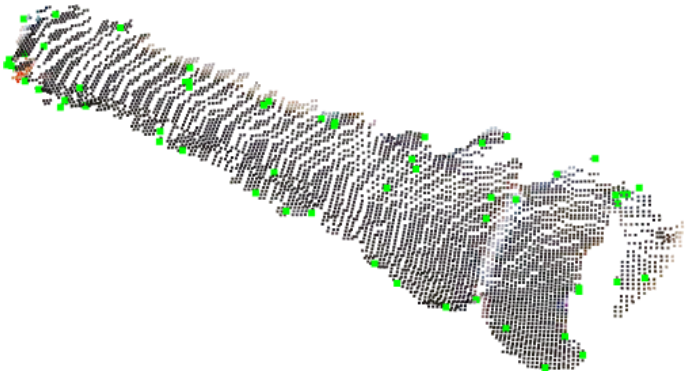
(a) Harris3D



(b) ISS



(c) Uniform sampling - 1 cm



(d) Uniform sampling - 2 cm

Figure 4.7: Results of various keypoint detection methods for `flashlight 1`.

CHAPTER 5

EXPERIMENTS AND RESULTS

In this chapter the proposed descriptors are evaluated and compared with the state-of-the-art 3D descriptors. Both *category-level* and *instance-level* object recognition performances are evaluated on publicly available RGB-D datasets. First, in Section 5.1, the evaluation method and metrics are explained. Then, in Section 5.2, the datasets are detailed. In Section 5.3, the effect of design parameters of SPAIR and CoSPAIR is investigated. In Section 5.4, the effect of various keypoint selection methods on performance is investigated. In Sections 5.5, 5.6, 5.7 and 5.8 the performance of the proposed descriptors on the chosen datasets are investigated and compared to state-of-the-art. In Section 5.9, the extraction and matching times of the descriptors are investigated. And lastly in Section 5.10, the effects of the size of the objects on recognition performance are investigated.

5.1 Evaluation Method and Metrics

We have compared the proposed descriptors against the state-of-the-art local 3D descriptors that are publicly available in the Point Cloud Library (PCL) [19]: PFH [24], PFHRGB [19], FPFH [23], SHOT [3, 21] and CSHOT [29]. The same testing procedure, which is summarized in Figure 5.1, is used for evaluating the descriptors.

For all the conducted tests/experiments, the surface normals are estimated with a search radius of 1 cm as in [16]. Then, the datasets used in the tests are split into a query set and a reference set depending on the test scenario. In this thesis, two different scenarios that are proposed in [5] are used:

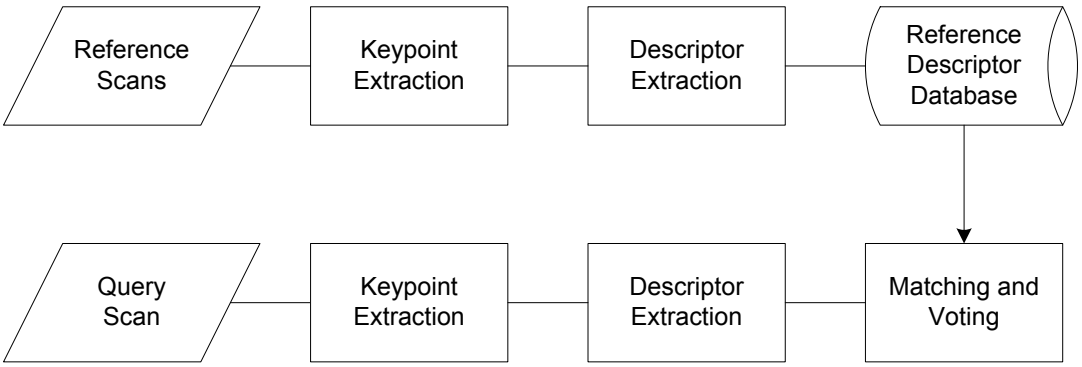


Figure 5.1: The standard procedure for evaluation of the descriptors.

1. *Leave-sequence-out*: Test and train sets are chosen to be from scans with different camera heights.

2. *Alternating contiguous frames*: The video sequences from different heights are divided into three contiguous sequences of equal length. Since there are three heights (videos) for each object in the datasets used, this gives nine video sequences for each object. Seven of these are randomly selected for training and the remaining two for test. Ten trials are performed and the results are averaged.

At the matching phase, the query descriptors are brute-force matched to the nearest descriptor in the reference descriptor database (see Figure 5.1) using Euclidean norm ($L^2 - norm$) and the final decision is made via a majority rule [52] as follows:

$$D(X) = \arg\max_{C} \sum_{i=1}^{K} I(f_i(X) = C), \tag{5.1}$$

where C is the class label, X is the object to be classified, f is a keypoint, K is the total number of keypoints on the query object and D is the final decision. For the *Matching and Voting* stage, OpenCV library [53] is used whereas for all the remaining stages, Point Cloud Library [19] is used. The performance of the descriptors are calculated as *average accuracy*, i.e., the average per-class effectiveness [54]:

$$\frac{1}{L} \sum_{i=1}^{L} \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i}, \tag{5.2}$$

where L is the total number of class labels and TP, TN, FP, FN are *true positives*, *true negatives*, *false positives* and *false negatives*, respectively. For extracting/detecting the keypoints, we have chosen the Intrinsic Shape Signatures 3D (ISS3D)

method [25], which is available in the PCL library. ISS3D has recently been shown to be among the top performing methods and it was reported to stand out for its performance, repeatability and efficiency [48, 50]. Our experiments also confirm these findings as detailed in Section 5.4.

5.2 The Datasets

The experiments were conducted on three different object recognition datasets in four configurations. The first dataset is the well-known RGB-D Object Dataset introduced by Lai et al. in 2011 [5]. This dataset was used in two different configurations. The first configuration is a subset that had been used by Luis A. Alexandre [16]. This subset is used for optimization and comprehensive analysis. The second configuration of this dataset consists of all the objects and is used for complementary analysis. The second dataset is the recently introduced BigBIRD ((Big) Berkeley Instance Recognition Dataset) by Singh et al. [7]. Lastly, the third dataset is the object scans used in the Amazon Picking Challenge at ICRA 2015 [8].

5.2.1 Dataset 1: Subset of the RGB-D Object Dataset

The RGB-D Object Dataset [5] consists of 300 common household objects in 51 categories. The objects were scanned with an RGB-D camera with 640×480 resolution from different angles and the total number of RGB-D images is around $250,000$.

As a first step in our experiments, a subset of this large dataset which contains 48 objects in 10 categories is chosen. The chosen subset was used by Luis A. Alexandre in a comprehensive evaluation of various descriptors that are available in PCL [16] and it contains the following categories: apple, ball, banana, bell pepper, binder, bowl, calculator, camera, cap and cell phone. Examples of segmented scans for each category are given in Figure 5.12.

In this subset, a total of 1421 point clouds are chosen as in [16]. The *leave-sequence-out* and *alternating contiguous frames* scenarios are applied for both category and instance-level recognition experiments. As in [5] and [16], for *leave-sequence-out*,

in the query set, the camera is mounted 45° above the horizontal axis relative to the turntable whereas in the reference set it is mounted 30° and 60° above. We refer to [55] for more details on the setup and query scans.

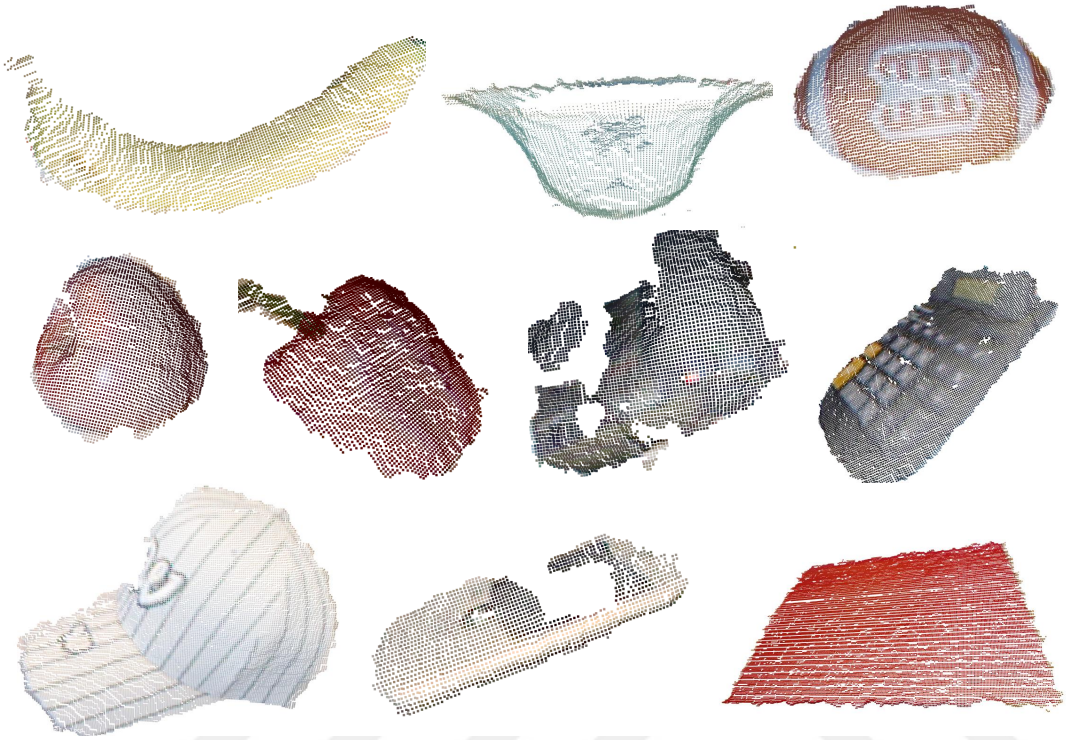


Figure 5.2: Examples of point clouds from the chosen 10 category subset of the RGB-D Object Dataset [5].

5.2.2 Dataset 2: RGB-D Object Dataset - All Objects

As our second dataset, the RGB-D Object Dataset with all 300 objects in 51 categories is used. Since the total number of images in the dataset as well as the number of scans per object is high, the scans in azimuth are sub-sampled by taking every twentieth sample. This yielded an average of 10 scans for each object for each video sequence (whole rotation on the turntable) from different camera heights; which produced a total of 9944 point clouds for test and training in total.

As in Dataset 1, for the *leave-sequence-out* scenario, the camera positions are chosen as 45° for the query set and 30° and 60° above the horizontal axis of the turntable for the reference set.

Figure 5.3: Sample scans from each 51 category of RGB-D Object Dataset [5] in alphabetical order from top left to bottom right.

5.2.3 Dataset 3: BigBIRD Dataset

BigBIRD is a recently introduced instance-level object recognition dataset introduced by Singh et al. [7] which is publicly available [6]. The RGB-D data was collected using a Carmine 1.09 sensor. The resolution of the RGB-D scans is the same as in Dataset 1, i.e., 640×480. The initial version of the dataset contains 100 objects and the dataset is being updated. At the time the tests were being performed, the dataset included a total of 123 objects. Some of the objects used in the experiments are shown in Figure 5.4.

However, in our tests, we excluded the transparent objects[1] due to their poor quality point clouds, as also stated in [7]. Two example scans can be seen in Figure 5.5. With the removal of the transparent objects, the resulting dataset contains 105 different objects.

BigBIRD is a very challenging dataset due to the extreme similarity between object instances. Not only many objects are similar in shape and size, but also product varieties of the same brand are labeled as different object instances - see Figure 5.6 for some samples.

In the BigBIRD dataset, the objects were scanned from 5 different polar angles and 120 azimuthal angles with a total of 600 images and point clouds per instance. As seen in Figure 5.7, the polar angles are named as `NP1`, `NP2`,...,`NP5` where `NP1` corresponds to a position where the sensors are located 0° with respect to the horizontal axis of the turntable, `NP5` corresponds to 90° and `NP2`, `NP3`, `NP4` located on a quarter circular arc in between [8]. In our experiments, for both test scenarios, we have used the poses similar to the experiments in the previous datasets. We have chosen the data obtained from positions `NP2`, `NP3` and `NP4` and for *leave-sequence-out* scenario, we have used `NP3` for the query and `NP2` and `NP4` for the reference sets. Additionally, not all azimuthal scans are used. The scans are sub-sampled by taking every tenth, resulting in approximately 12 scans per object. With the chosen views and sub-sampling of scans, a total of 3746 point clouds are used in experiments.

[1] The transparent objects are: `aunt jemima original syrup`, `bai5 sumatra dragonfruit`, `coca cola glass bottle`,`listerine`, `palmolive` (two instances), `softsoap` (five instances), `vo5` (three instances), `whiterain` (three instances) and `windex`.

Figure 5.4: Some of the objects in the BigBIRD dataset [6, 7].

(a) (b) (c) (d)

Figure 5.5: Example scans for transparent objects from the BigBIRD dataset [6, 7].

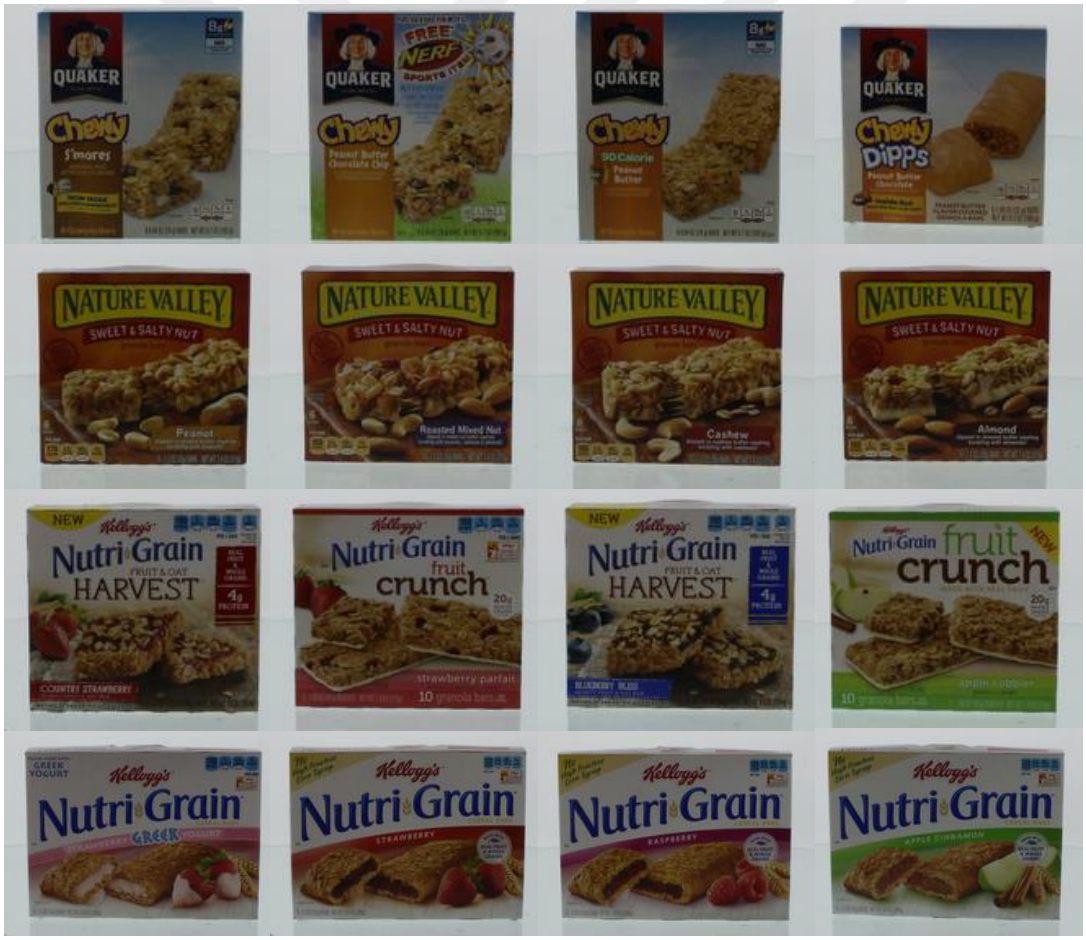


Figure 5.6: Sample RGB images (taken by the Carmine sensors) from the BigBIRD dataset [6, 7], each from another object.

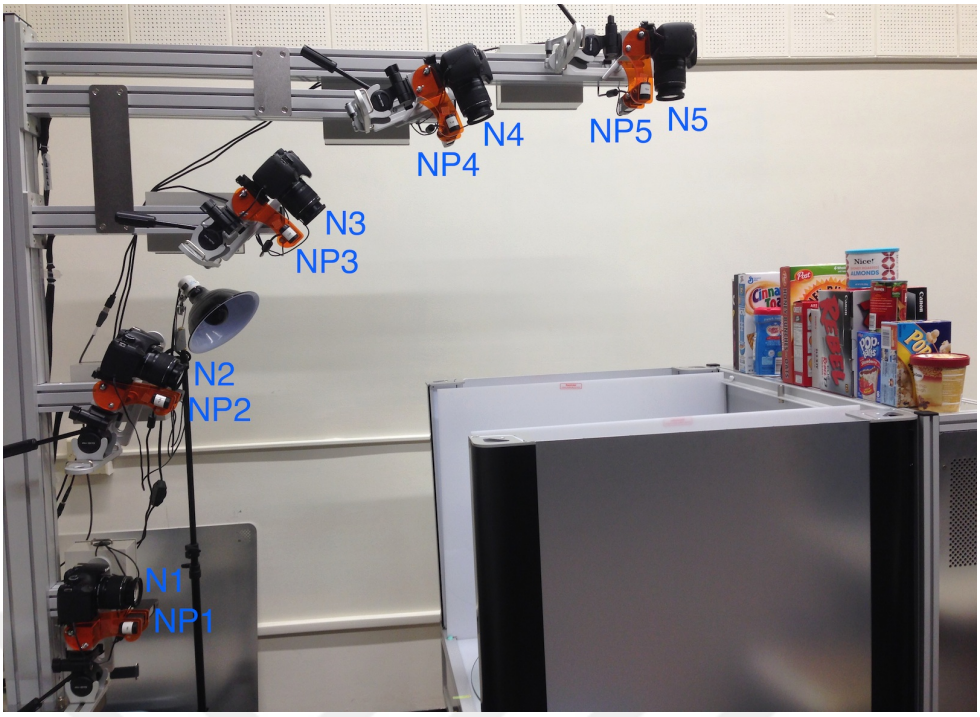


Figure 5.7: The sensor setup in the BigBIRD dataset [6] (image is used with author permission).

5.2.4 Dataset 3: Amazon Picking Challenge Dataset

The dataset was collected for the first Amazon Picking Challenge at ICRA 2015 using the same system setup (see Figure 5.7) as in the BigBIRD Dataset [7], [56] and is publicly available [8]. The dataset is composed of 26 different objects. Although some of the objects such as `safety works safety glasses`, `munchkin white hot duck bath toy` and `first years take and toss straw cups` have significantly below-average quality models, they are not excluded from the tests since they are not high in number. Some of the objects from the dataset including the challenging ones that have transparent parts are given in Figure 5.8. The same procedure used for the BigBIRD dataset (Section 5.2.3) is followed for choosing the scans for the experiments. This yielded a total of 949 point clouds to be used in the experiments.

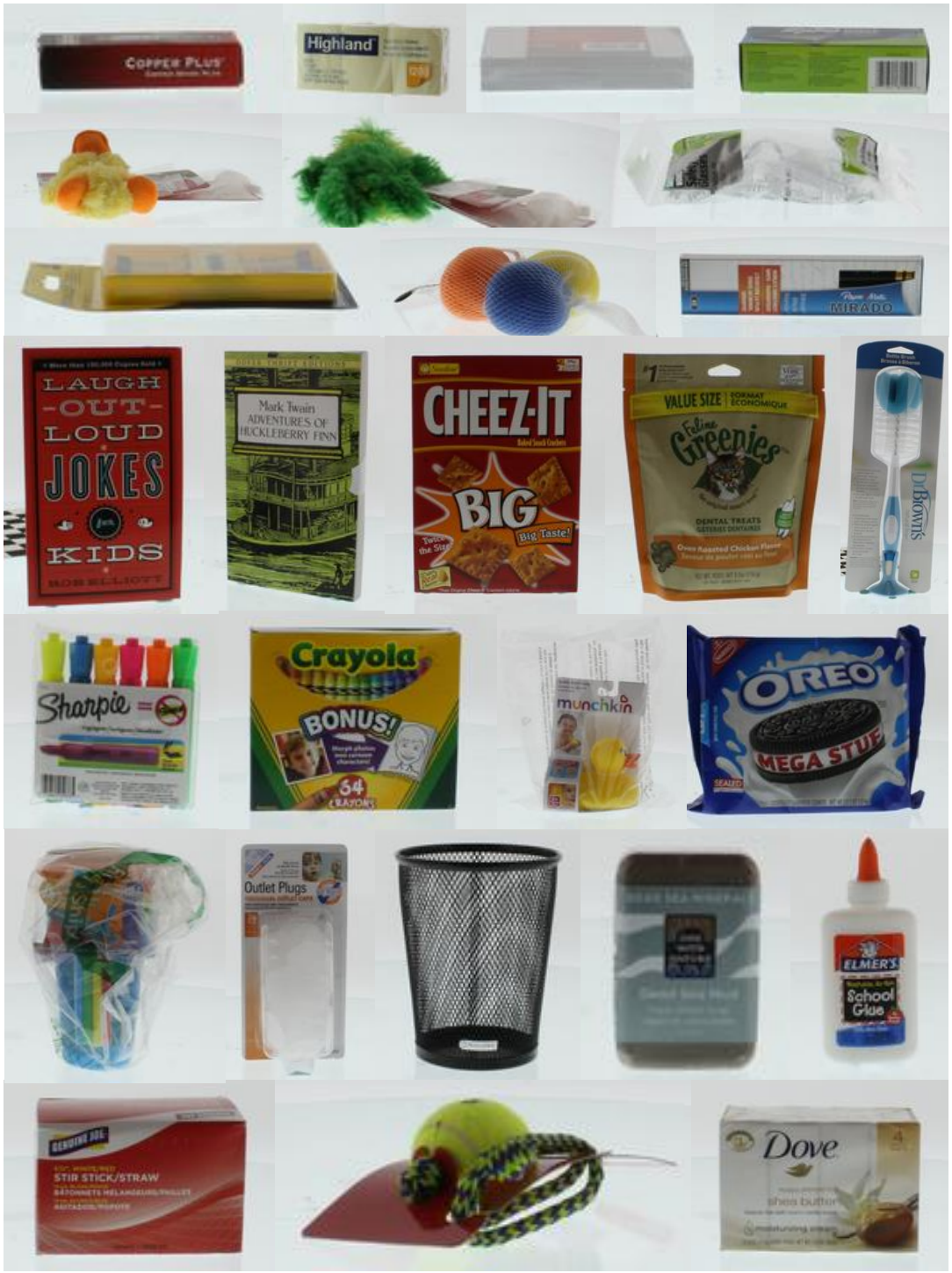


Figure 5.8: Some of the objects in the Amazon Picking Challenge dataset [8].

5.3 Tuning SPAIR/CoSPAIR: Choosing Number of Bins and Concentric Levels

There are two parameters in our descriptors: the number of concentric levels and the number of bins used for each sub-feature (angular relations given in Equations 3.4, 3.5, 3.6 for SPAIR; both angular relations and additional color histograms for CoSPAIR). To set these parameters, various experiments were conducted on Dataset 1: Subset of the RGB-D Object Dataset.

As the first step, we tested the performance of the SPAIR and CoSPAIR descriptor for various bin numbers. For 7 levels and a support radius of 10 cm, accuracy results are given in Figure 5.10. We see that 9 bins for each sub-feature provide the best accuracy considering instance-level recognition and second best with a minimal margin for category-level recognition. A similar analysis for CoSPAIR also yields similar results. Therefore, the number of bins is set to 9 for both SPAIR and CoSPAIR.

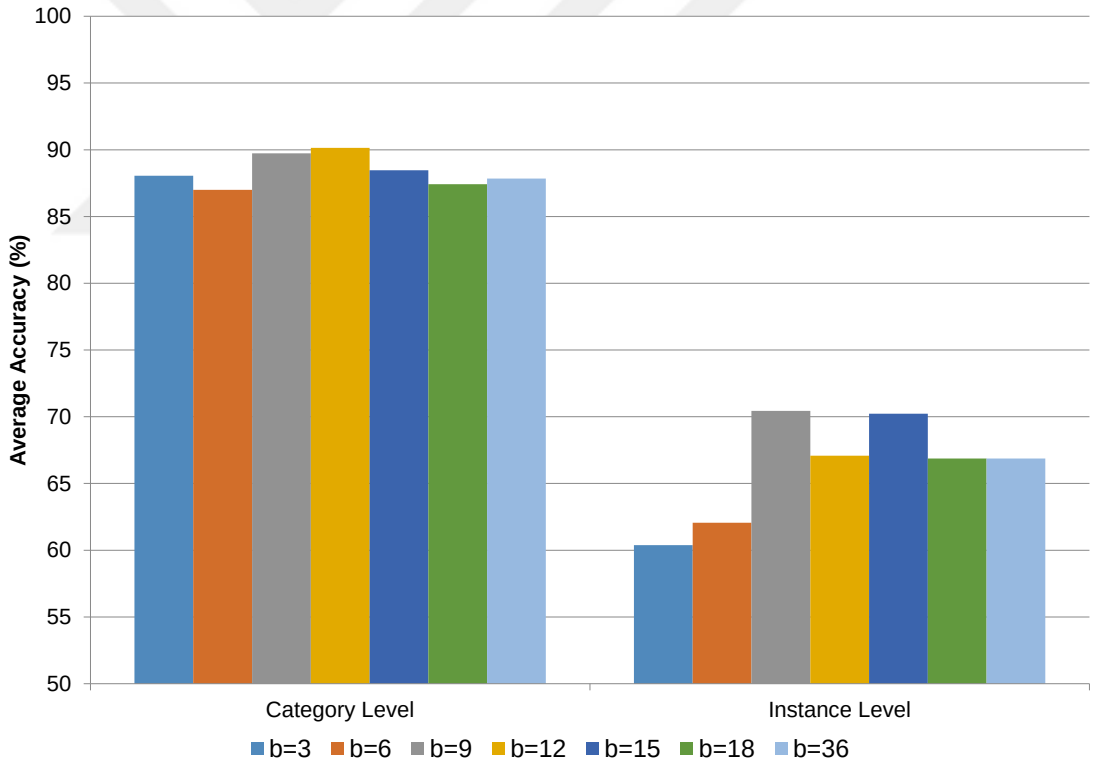


Figure 5.9: *Leave-sequence-out* average accuracy of SPAIR versus number of bins used in each level for each sub-feature where support radius is 10 cm and the number of levels is 7.

The second parameter is the number of the concentric levels. As our aim was to have

Table 5.1: Average accuracy of SPAIR versus number of bins used in each level for each sub-feature (L=7).

	b=3	**b=6**	**b=9**	**b=12**	**b=15**	**b=18**	**b=36**
Category Level	88.05	87.00	89.73	90.15	88.47	87.42	87.84
Instance Level	60.38	62.05	70.44	67.09	70.23	66.88	66.88

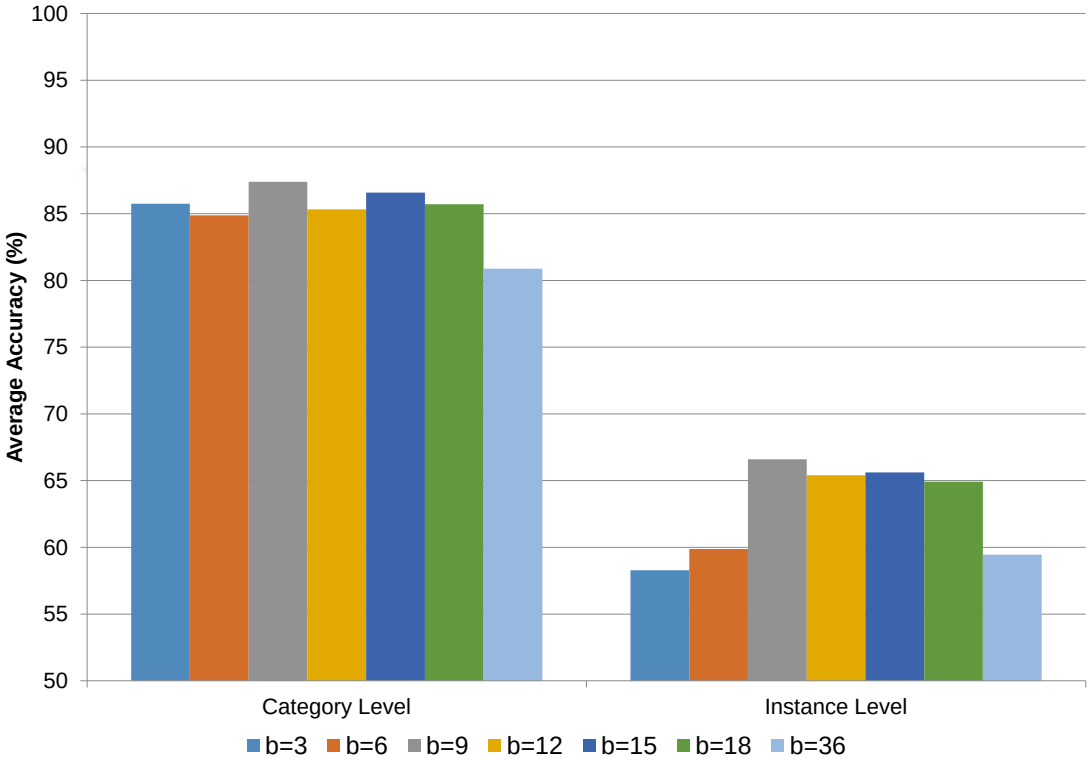


Figure 5.10: Average accuracy of SPAIR versus number of bins used in each level for each sub-feature (L=10).

Table 5.2: Average accuracy of SPAIR versus number of bins used in each level for each sub-feature (L=10).

	b=3	**b=6**	**b=9**	**b=12**	**b=15**	**b=18**	**b=36**
Category Level	85.74	84.87	87.40	85.32	86.58	85.71	80.88
Instance Level	58.28	59.87	66.60	65.41	65.62	64.92	59.45

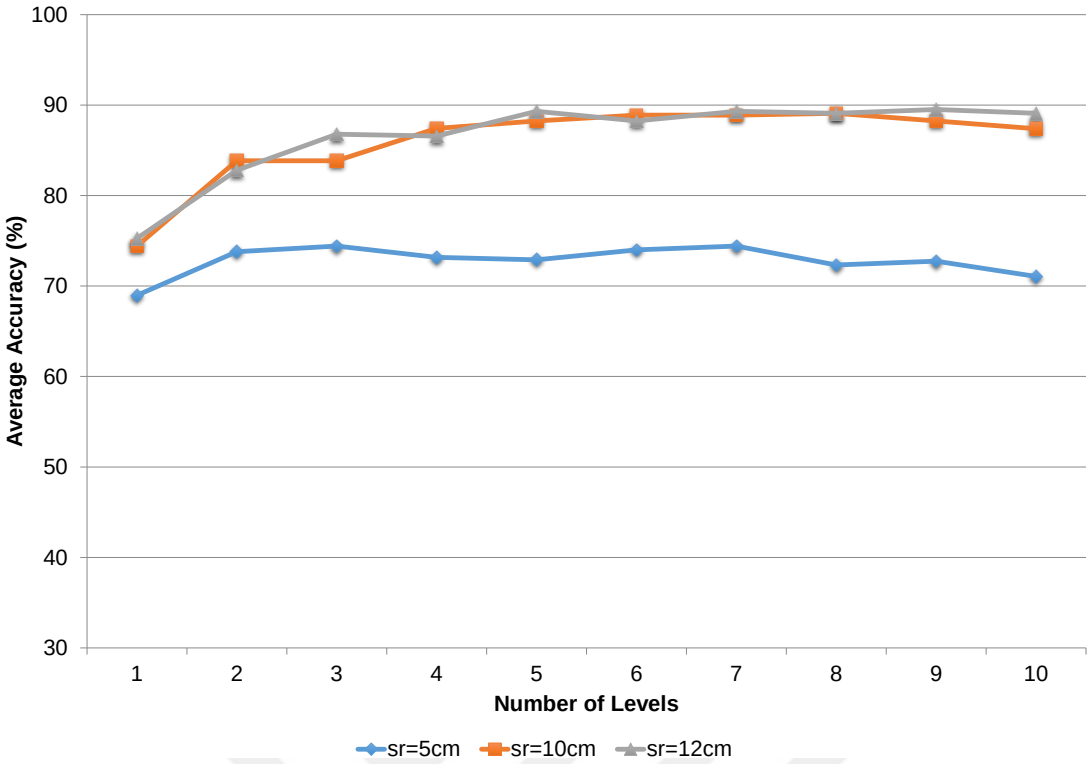
a fixed the number of levels regardless of the chosen support radius, experiments were conducted for various support radius sizes. The results are given in Figure 5.11a for category-level recognition and in Figure 5.11b for instance-level recognition. As can be observed from these figures, there is not a single particular number of levels where the accuracy is the highest for all support radius sizes. The performance is fairly stable after 4 levels with peak performances at around 7 and 8 levels. A similar analysis for CoSPAIR also reveals the same results. Therefore, the number of concentric levels was chosen to be 7 for all support radius sizes for both SPAIR and CoSPAIR.

Based on these choices, the size of the SPAIR descriptor becomes 189 due to 7 levels where each level consists of 3 histograms with 9 bins each. On the other hand, the size of the CoSPAIR descriptor is 378, i.e., double the size of the SPAIR descriptor due to the color histograms. In the remainder of the paper, the parameters of SPAIR and CoSPAIR are fixed and no further optimization is performed for Datasets 2, 3 and 4.

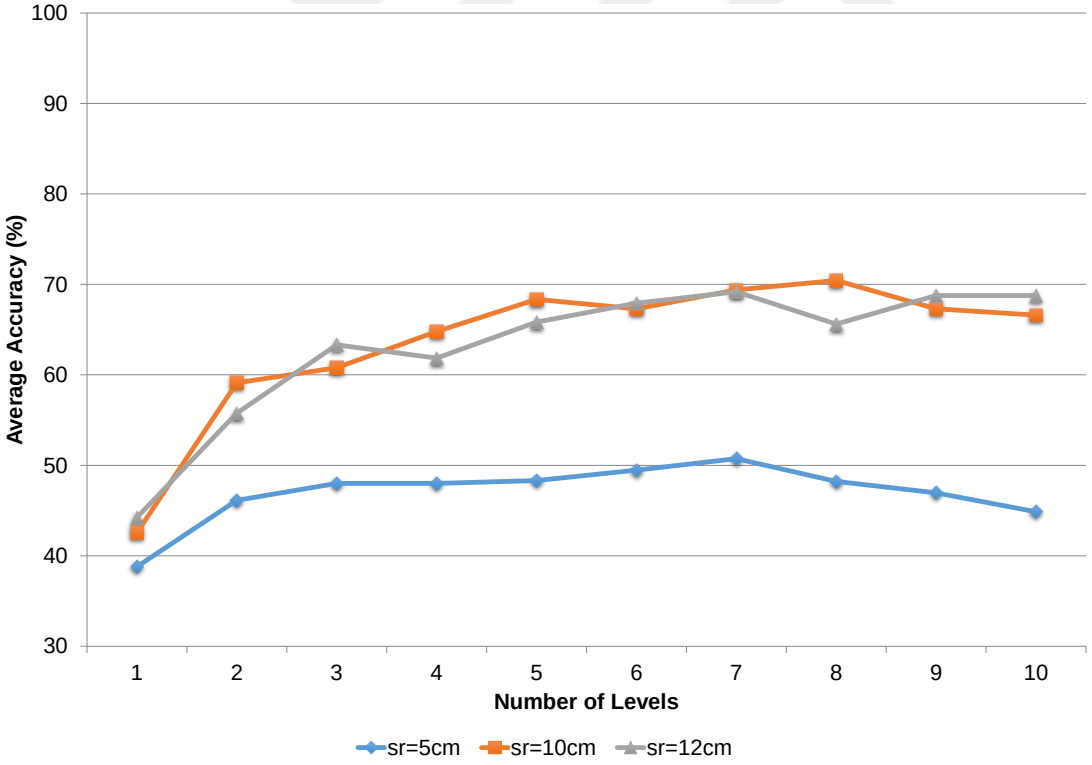
It should be noted that the parameters of the other compared descriptors are fixed in the Point Cloud Library at their best values and cannot be directly modified. Therefore, we used them as they are provided in the Point Cloud Library.

Table 5.3: Average accuracy of SPAIR vs number of concentric levels used to extract the descriptor: Category-level in *leave-sequence-out* scenario.

# of Levels	**sr=5cm**	**sr=10cm**	**sr=12cm**
L=1	68.97	74.42	75.26
L=2	73.79	83.86	82.81
L=3	74.42	83.86	86.79
L=4	73.17	87.42	86.58
L=5	72.90	88.26	89.31
L=6	74.00	88.89	88.26
L=7	74.42	88.89	89.31
L=8	72.33	89.10	89.10
L=9	72.75	88.26	89.52
L=10	71.07	87.40	89.10



(a)



(b)

Figure 5.11: Average accuracy of SPAIR vs number of concentric levels used to extract the descriptor in *leave-sequence-out* scenario: a) Category-level, b) Instance-level.

Table 5.4: Average accuracy of SPAIR vs number of concentric levels used to extract the descriptor: Instance-level in *leave-sequence-out* scenario.

# of Levels	**sr=5cm**	**sr=10cm**	**sr=12cm**
L=1	38.78	42.56	44.23
L=2	46.12	59.12	55.77
L=3	48.01	60.80	63.31
L=4	48.01	64.78	61.84
L=5	48.32	68.34	65.83
L=6	49.48	67.30	67.92
L=7	50.73	69.39	69.18
L=8	48.22	70.44	65.62
L=9	46.96	67.30	68.76
L=10	44.86	66.60	68.76

5.4 Effect of Keypoint Detection Methods

The performances of all the descriptors were also evaluated for various keypoint detection methods; ISS3D [25], Harris3D [19] and uniform sampling using a 3D voxel grid with a leaf size of 1 cm. The average accuracy results are given in Table 5.5. It can be observed that the keypoint detection methods affect all the tested descriptors similarly. Therefore, it is possible to choose a single extractor for all the descriptors. According to our evaluation, ISS3D performs better than Harris3D and its performance is very close to uniform sampling. Since ISS3D has been reported to stand out for its performance, repeatability and efficiency [48, 50] we used it as the keypoint detection method in our experiments.

Table 5.5: Average accuracy results of descriptors for different keypoint extraction methods where support radius is 10 cm in *leave-sequence-out* scenario.

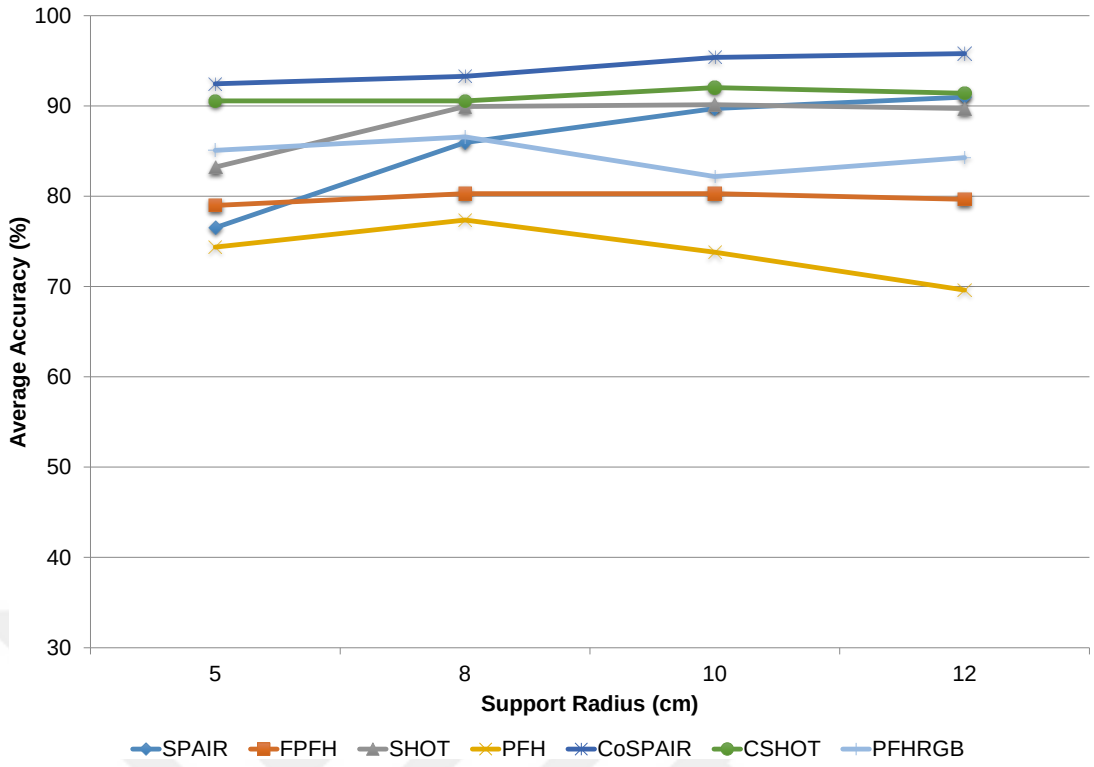
	Category Level			Instance Level		
	ISS3D	H3D	Uni.	ISS3D	H3D	Uni.
SPAIR	89.73	68.76	89.94	70.44	38.99	68.55
FPFH	80.29	66.88	81.93	51.36	37.53	51.05
SHOT	90.15	80.92	90.97	61.84	50.31	65.55
CoSPAIR	95.39	87.00	96.23	84.91	72.75	86.16
CSHOT	92.03	85.95	94.54	79.66	68.76	82.35

5.5 Results on Dataset 1: RGB-D Subset

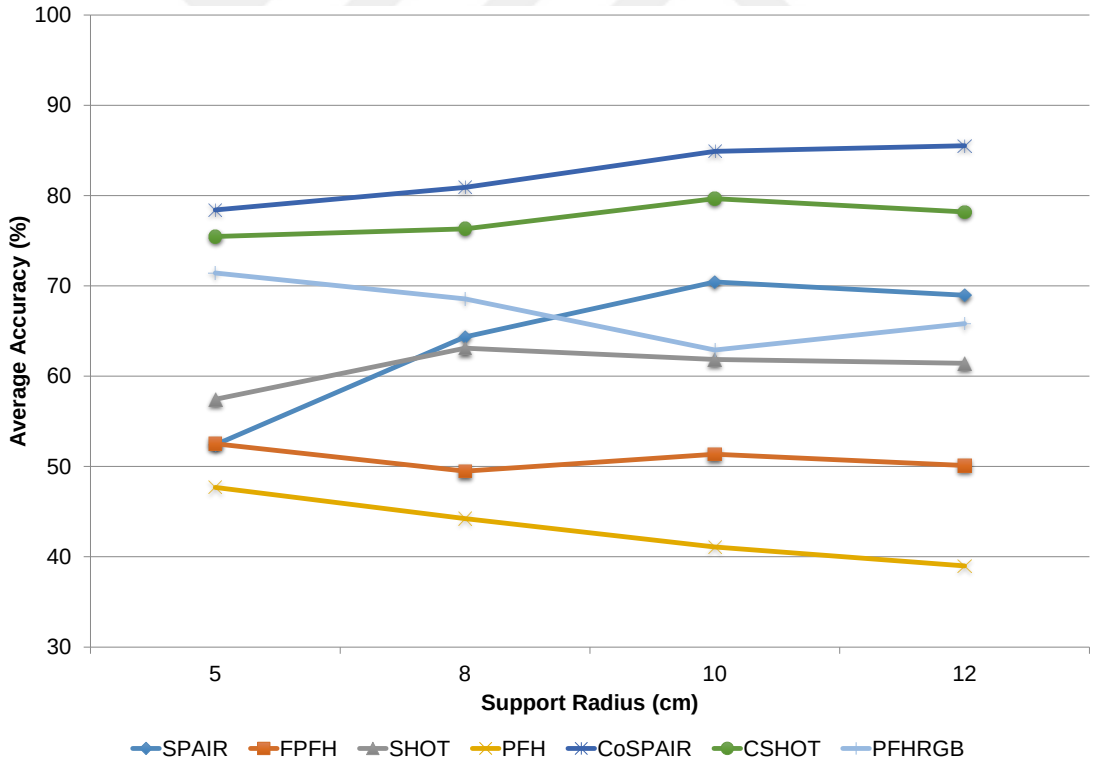
The average accuracy, average per class recall and precision results are given in Table 5.6 for category-level recognition and Table 5.7 for instance-level recognition. In addition, *leave-sequence-out* average accuracies are shown in Figure 5.12 to visualize the performance trend with respect to the support radius size.

Results show that, in this small dataset, CoSPAIR slightly outperforms the second best performer CSHOT in category-level recognition, except for the *Alternating contiguous frames* methodology for low support radius sizes. CoSPAIR outperforms CSHOT with a higher margin in instance-level recognition using both methodologies (*leave-sequence-out* and *alternating-contiguous-frames*). In the *leave-sequence-out* methodology, CoSPAIR achieves 85.53% average accuracy at 12 cm whereas CSHOT achieves 79.66% at 10 cm; in the *alternating-contiguous-frames* methodology, CoSPAIR achieves 91.96% average accuracy at 10 cm compared to CSHOT's 87.20% at 8 cm.

Among the shape-only descriptors, SPAIR performs slightly better for larger support radius sizes whereas SHOT performs better for smaller support radius sizes.



(a)



(b)

Figure 5.12: Average accuracy results for 10 category subset of RGB-D Object Dataset in *leave-sequence-out* scenario: a) Category-level, b) Instance-level.

Table 5.6: Category-level average accuracy, average recall and average precision results for the 10 category subset of RGB-D Object Dataset.

	sr = 5cm			**sr = 8cm**			**sr = 10cm**			**sr = 12cm**		
						Leave-sequence-out						
	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.
SPAIR	76.52	74.85	78.95	85.95	84.58	84.12	89.73	88.90	88.87	90.99	90.26	90.72
FPFH	78.99	78.15	76.31	80.29	79.90	77.08	80.29	80.35	77.13	79.66	80.00	76.99
SHOT	83.23	81.07	82.17	89.94	88.63	87.53	90.15	88.66	87.92	89.73	88.63	88.18
PFH	74.37	74.31	71.33	77.36	77.08	74.89	73.79	74.18	73.28	69.60	70.48	73.07
CoSPAIR	**92.45**	**92.01**	**92.73**	**93.29**	**92.90**	**92.01**	**95.39**	**95.30**	**94.96**	**95.81**	**95.68**	**95.44**
CSHOT	90.57	89.89	90.89	90.57	89.15	89.77	92.03	91.48	92.29	91.40	90.95	91.45
PFHRGB	85.08	84.72	84.60	86.58	85.98	85.99	82.18	82.00	83.36	84.28	83.37	82.72
						Alternating contiguous frames						
SPAIR	78.54	75.67	78.20	88.25	86.17	86.76	90.27	88.57	88.79	91.26	89.66	90.29
FPFH	81.25	79.46	78.48	82.92	81.39	80.92	81.34	80.31	79.60	80.38	79.52	79.17
SHOT	87.89	85.74	86.97	92.52	91.01	91.11	93.39	91.78	92.13	93.36	92.21	92.68
PFH	78.58	76.92	76.81	78.57	77.10	77.97	74.78	74.47	77.48	72.76	72.59	76.57
CoSPAIR	96.45	95.46	96.23	97.09	96.29	96.84	**97.98**	**97.42**	**97.61**	**97.82**	**97.14**	**97.43**
CSHOT	**97.02**	**96.48**	**96.85**	**97.76**	**97.38**	**97.75**	97.44	97.07	97.22	97.14	96.69	96.89
PFHRGB	92.98	91.72	92.35	93.96	92.88	93.45	91.15	89.99	90.62	92.37	91.01	91.62

Table 5.7: Instance-level average accuracy, average recall and average precision results for the 10 category subset of RGB-D Object Dataset.

	sr = 5cm			**sr = 8cm**			**sr = 10cm**			**sr = 12cm**		
						Leave-sequence-out						
	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.
SPAIR	52.41	52.55	51.10	64.36	64.62	66.09	70.44	70.54	72.75	68.97	69.25	70.81
FPFH	52.52	52.62	48.89	49.48	49.76	48.00	51.36	51.71	48.58	50.10	50.44	45.46
SHOT	57.44	57.58	59.81	63.10	63.66	63.01	61.84	62.36	62.16	61.43	61.89	64.18
PFH	47.69	47.85	45.51	44.23	44.51	44.16	41.09	41.39	40.65	38.99	39.26	40.13
CoSPAIR	**78.41**	**78.59**	**76.88**	**80.92**	**81.16**	**79.29**	**84.91**	**85.28**	**86.64**	**85.53**	**85.90**	**86.06**
CSHOT	75.47	76.02	74.20	76.31	76.88	76.24	79.66	80.17	77.85	78.20	78.75	75.99
PFHRGB	71.43	71.64	69.95	68.55	68.80	66.91	62.89	63.06	63.29	65.83	66.04	64.84
						Alternating contiguous frames						
SPAIR	55.45	55.07	55.98	65.11	64.98	65.85	66.87	66.76	67.56	66.76	66.69	67.42
FPFH	56.21	55.95	55.64	57.14	56.99	56.00	56.73	56.63	55.01	56.51	56.38	55.36
SHOT	62.62	62.54	64.72	65.71	65.78	67.75	66.12	66.18	68.62	65.85	65.87	68.09
PFH	52.18	52.02	50.94	49.69	49.60	48.32	47.50	47.45	48.46	46.68	46.72	48.90
CoSPAIR	**90.82**	**90.63**	**91.07**	**91.64**	**91.52**	**92.15**	**91.96**	**91.85**	**92.42**	**91.64**	**91.51**	**91.98**
CSHOT	87.16	87.08	88.11	87.20	87.19	88.73	86.15	86.12	87.71	85.87	85.84	87.26
PFHRGB	81.86	81.69	83.29	82.91	82.77	83.76	77.73	77.55	79.78	81.19	81.20	82.73

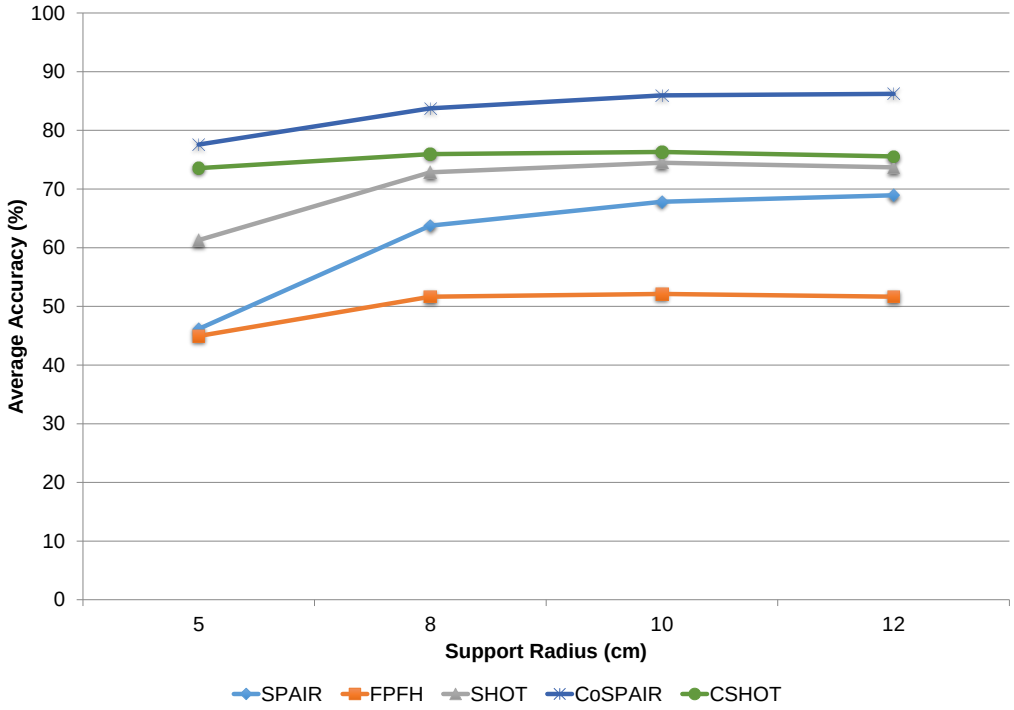
5.6 Results on Dataset 2: RGB-D All Objects

Next, we have evaluated the methods on the whole RGB-D Object Dataset (with all the available 300 objects in 51 categories), which is much more challenging than Dataset 1. The average accuracy, average per class recall and precision results are given in Table 5.8 for category-level recognition and Table 5.9 for instance-level recognition. Additionally, *leave-sequence-out* average accuracies are shown in Figure 5.13 to visualize the performance trend with respect to the support radius size. It should be noted that PFH and PFHRGB are excluded from this experiment because of these descriptors' prohibitively long extraction times on such a big dataset (see Section 5.9).

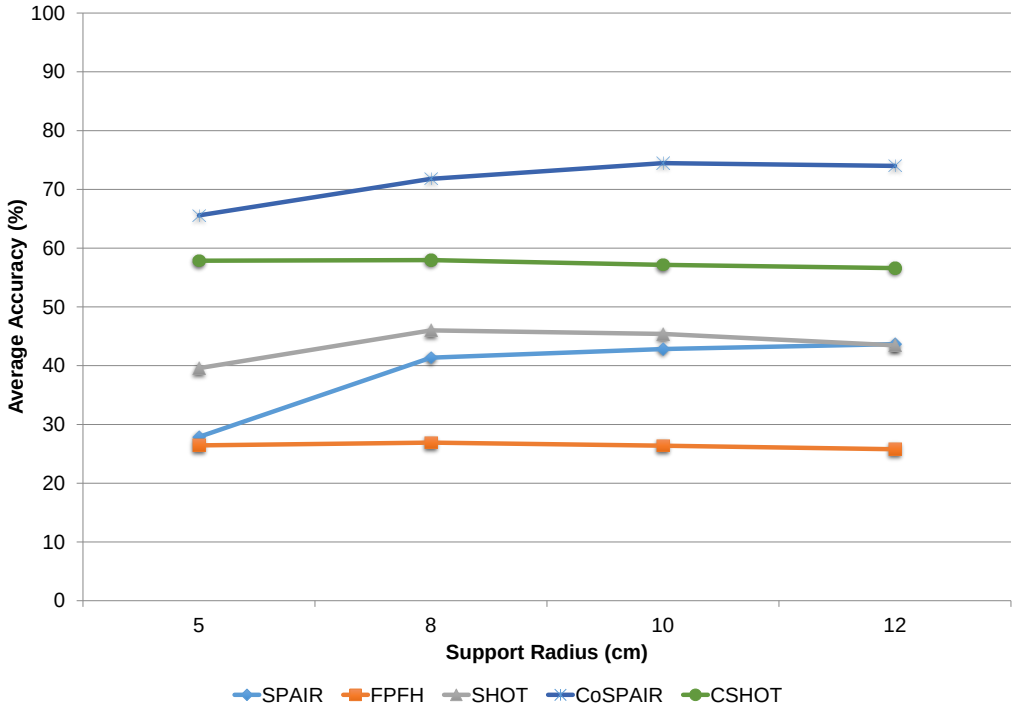
Table 5.8: Category-level average accuracy, average recall and average precision results for the RGB-D Object Dataset

	sr = 5cm			**sr = 8cm**			**sr = 10cm**			**sr = 12cm**		
						Leave-sequence-out						
	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.
SPAIR	46.16	48.61	51.63	63.77	64.31	66.93	67.84	67.11	69.29	68.93	67.85	69.83
FPFH	44.94	46.21	45.51	51.66	51.16	51.29	52.11	51.66	52.18	51.66	51.01	50.97
SHOT	61.28	63.06	64.92	72.85	72.86	73.13	74.49	73.90	74.25	73.70	73.03	73.08
CoSPAIR	**77.59**	**77.40**	**78.56**	**83.75**	**83.42**	**83.72**	**85.97**	**85.43**	**84.79**	**86.21**	**85.44**	**84.97**
CSHOT	73.55	72.74	74.59	75.95	74.71	77.03	76.31	74.86	77.54	75.55	74.14	76.10
						Alternating contiguous frames						
SPAIR	55.52	54.19	55.57	70.26	68.22	68.69	73.44	71.17	71.83	74.97	72.55	73.09
FPFH	55.97	53.45	52.45	61.60	58.47	58.82	62.40	59.10	58.96	62.48	59.12	59.03
SHOT	74.49	72.89	72.93	78.97	76.98	77.12	80.78	78.98	78.99	79.99	78.01	78.02
CoSPAIR	**92.88**	**91.98**	**92.84**	**94.98**	**94.30**	**94.43**	**95.68**	**95.03**	**95.20**	**96.15**	**95.49**	**95.63**
CSHOT	90.53	89.45	89.75	90.36	89.42	89.81	91.15	90.21	90.44	90.57	89.44	89.88

In this dataset, for all support radius sizes and for both test scenarios, CoSPAIR outperforms all other descriptors in both category and instance-level recognition. For the *leave-sequence-out* scenario, CoSPAIR achieves an average accuracy of 86.21% for a support radius of 12 cm in category-level recognition and 74.46% in instance-level recognition for a support radius of 10 cm whereas the second top performer CSHOT achieves 76.31% in category-level recognition for a support radius of 10 cm

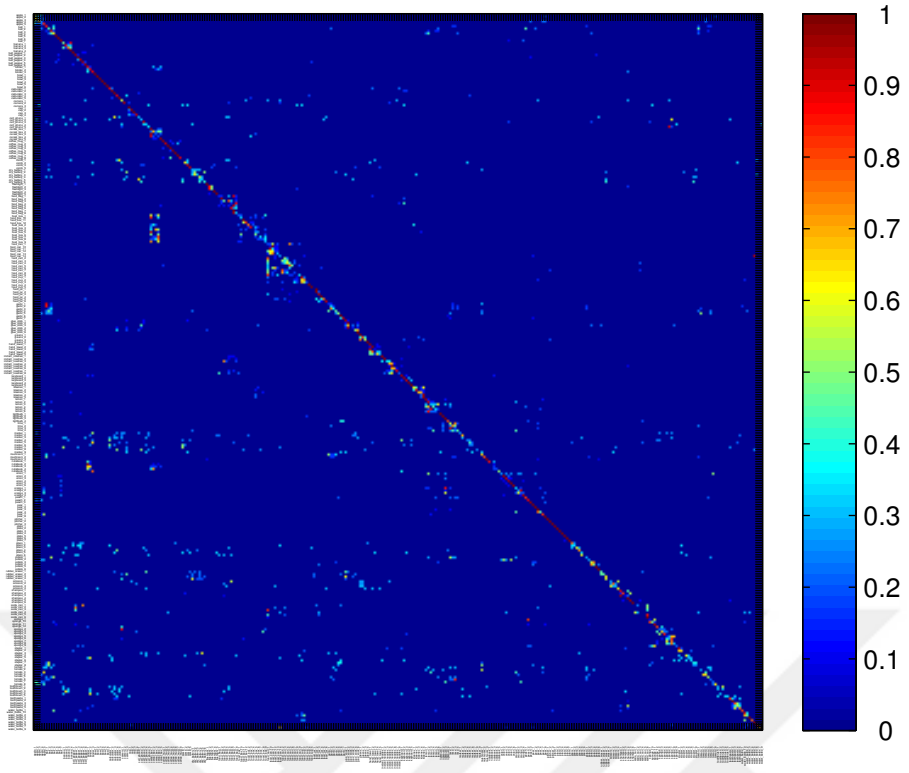


(a)

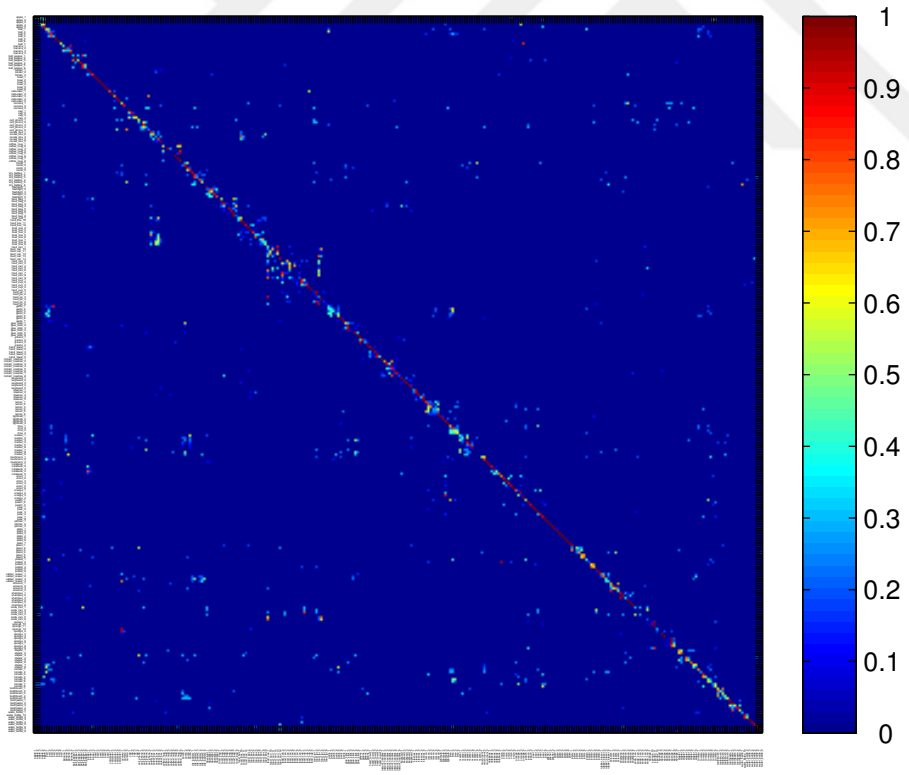


(b)

Figure 5.13: Average accuracy results for the whole RGB-D Object Dataset in *leave-sequence-out* scenario: a) Category-level, b) Instance-level.

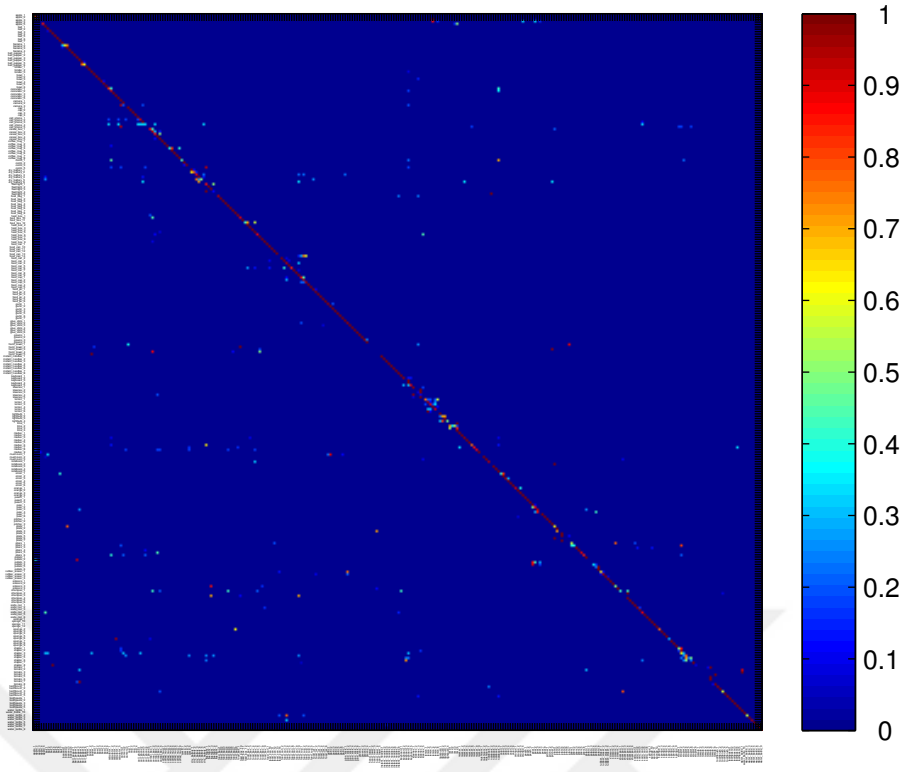


(a) SPAIR (sr = 10cm)

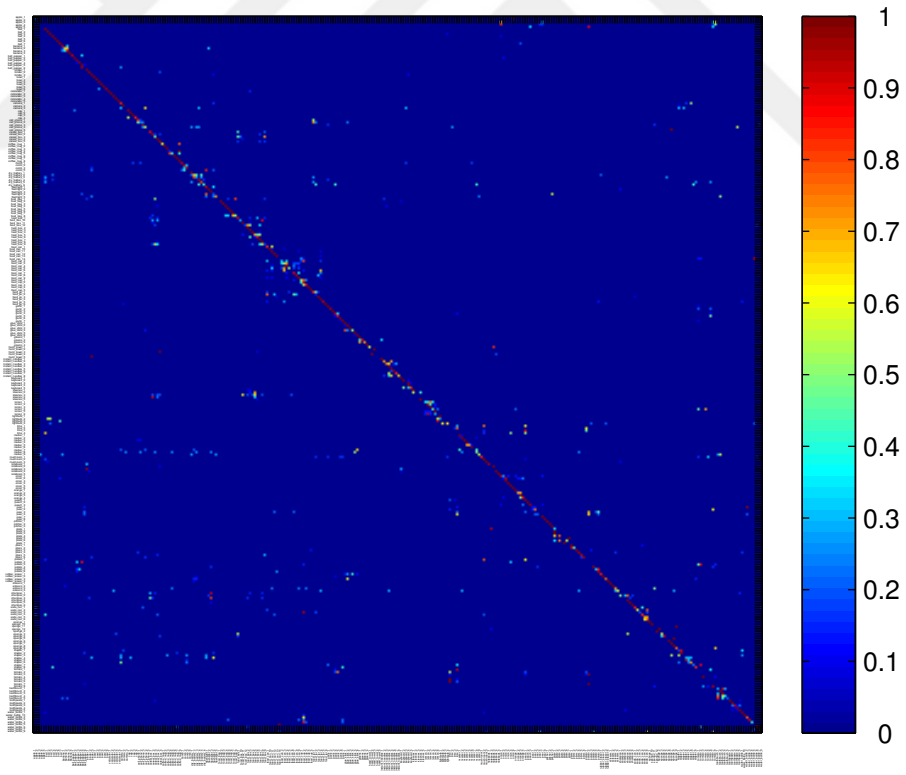


(b) SHOT (sr = 10cm)

Figure 5.14: Confusion matrices for the RGB-D Object Dataset - instance level in *leave-sequence-out* scenario.

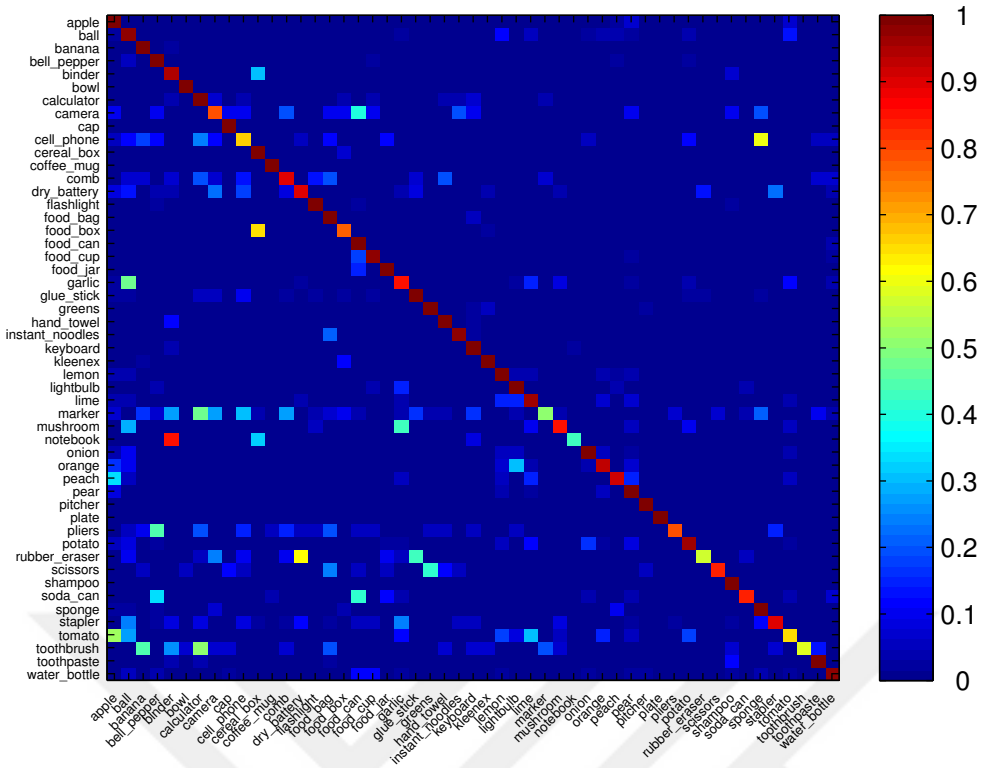


(c) CoSPAIR (sr = 10cm)

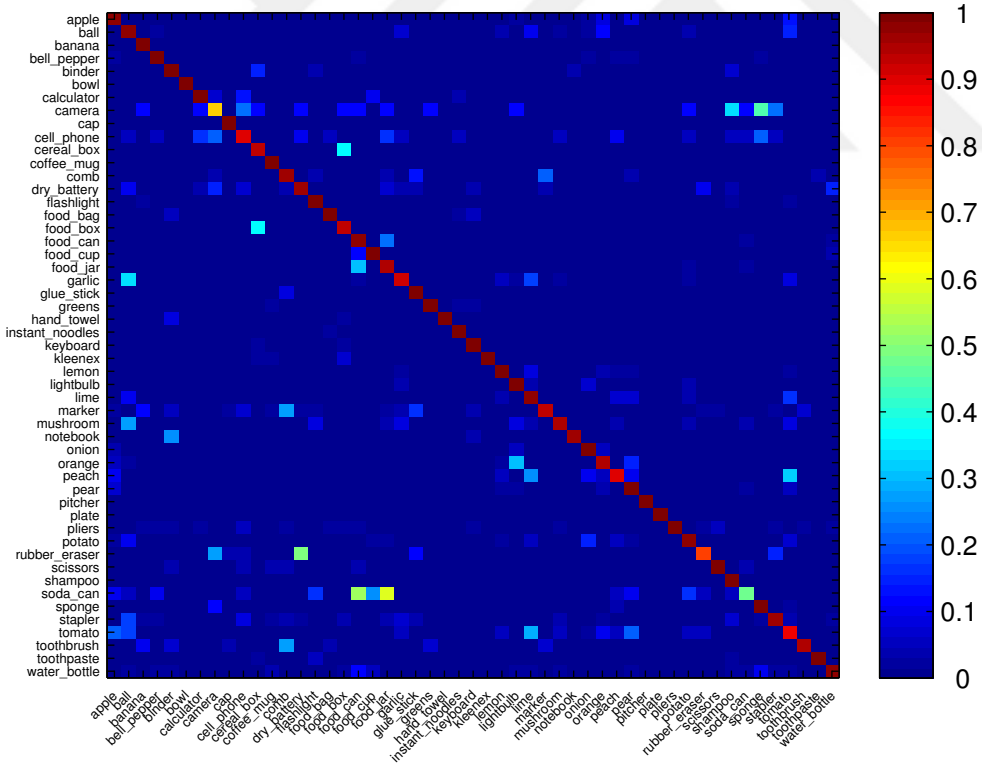


(d) CSHOT (sr = 10cm)

Figure 5.14: Confusion matrices for the RGB-D Object Dataset - instance level in *leave-sequence-out* scenario. (cont.)

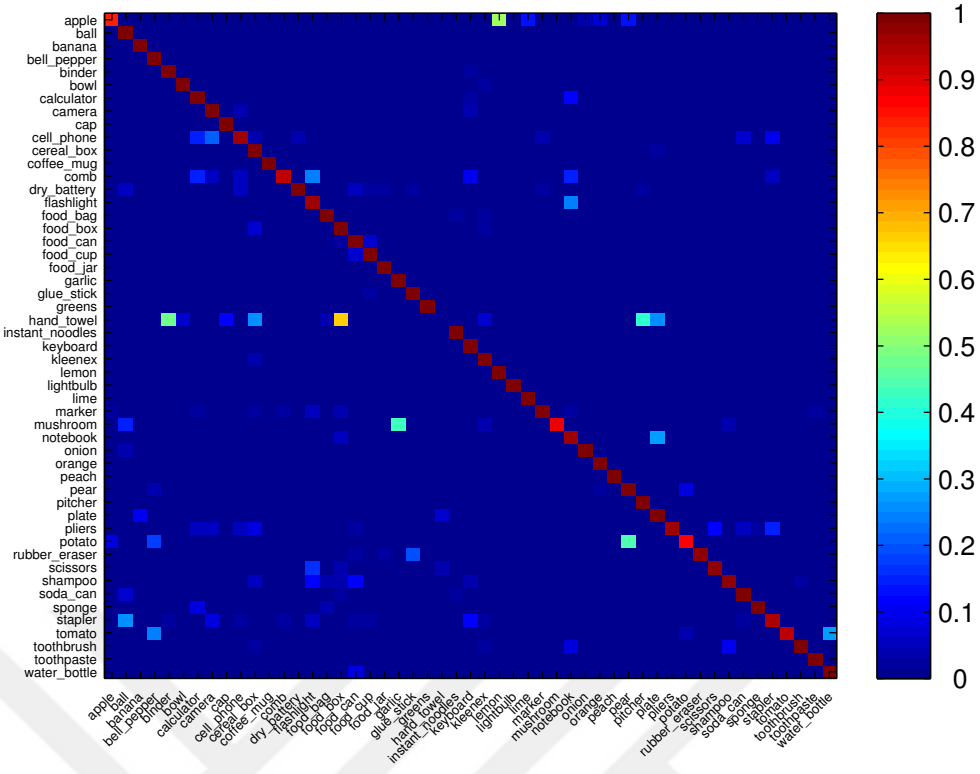


(a) SPAIR (sr = 10cm)

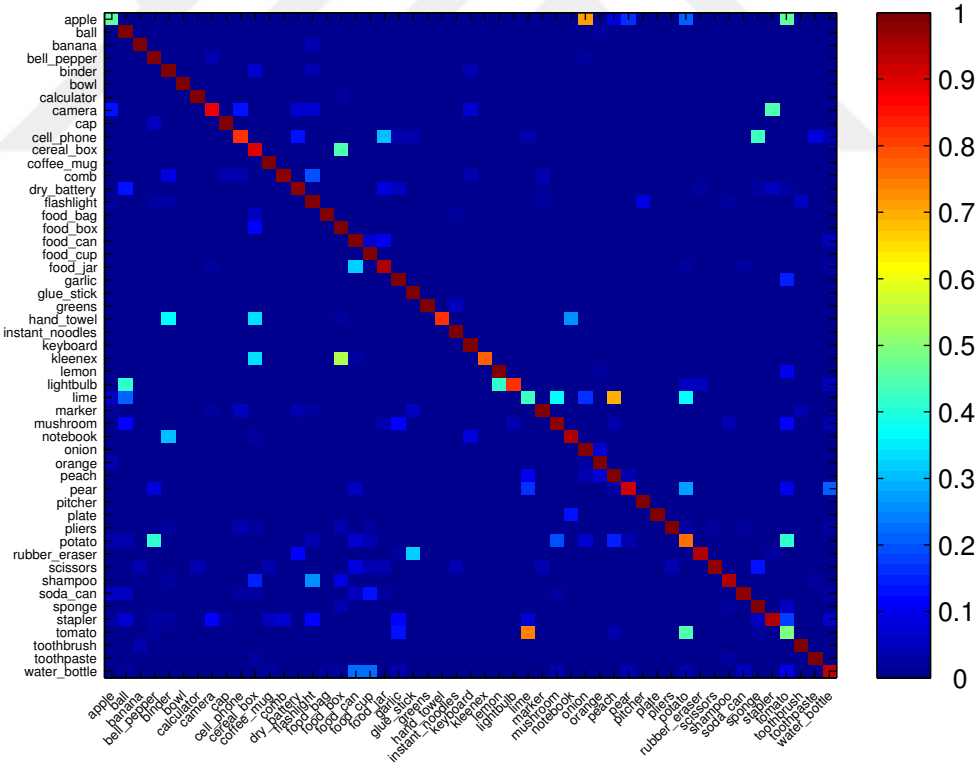


(b) SHOT (sr = 10cm)

Figure 5.15: Confusion matrices for the RGB-D Object Dataset - category level in *leave-sequence-out* scenario.



(c) CoSPAIR (sr = 10cm)



(d) CSHOT (sr = 10cm)

Figure 5.15: Confusion matrices for the RGB-D Object Dataset - category level in *leave-sequence-out* scenario (cont.)

Table 5.9: Instance-level average accuracy, average recall and average precision results for the RGB-D Object Dataset.

	sr = 5cm			**sr = 8cm**			**sr = 10cm**			**sr = 12cm**		
						Leave-sequence-out						
	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.
SPAIR	27.88	28.46	27.81	41.36	41.90	42.00	42.82	43.44	44.42	43.67	44.28	45.19
FPFH	26.42	27.22	25.10	26.91	27.34	26.29	26.39	26.99	25.49	25.78	26.44	25.12
SHOT	39.57	39.92	41.51	46.01	46.13	47.26	45.40	45.44	46.43	43.46	43.53	45.39
CoSPAIR	**65.56**	**64.26**	**64.54**	**71.79**	**70.42**	**69.23**	**74.46**	**73.22**	**72.46**	**74.01**	**72.79**	**71.66**
CSHOT	57.85	56.65	55.67	57.97	56.71	56.44	57.15	55.96	55.89	56.60	55.48	55.31
						Alternating contiguous frames						
SPAIR	36.88	36.87	36.10	49.45	49.10	48.58	51.83	51.44	51.28	52.18	51.74	51.27
FPFH	37.13	36.97	34.55	41.56	41.21	39.40	41.71	41.37	39.17	41.66	41.29	39.28
SHOT	50.89	50.67	51.74	53.69	53.31	54.77	55.09	54.80	55.84	54.29	53.95	54.76
CoSPAIR	**87.52**	**86.41**	**88.01**	**89.26**	**88.21**	**89.30**	**89.89**	**88.90**	**90.14**	**90.09**	**89.10**	**90.29**
CSHOT	81.17	80.28	81.76	78.57	77.73	79.92	79.95	79.15	81.08	79.12	78.24	80.03

and 57.97% in instance-level recognition for a support radius of 8 cm, leading to 16.49 percentage points (pp) performance difference. It is even higher if the same support radius is considered for all the descriptors; resulting up to 17.41 pp difference at 12 cm. For the *alternating-contiguous-frames* scenario, CoSPAIR outperforms competitors as well but with a slightly lower margin. CoSPAIR achieves an average accuracy of 96.15% for a support radius of 12 cm in category-level recognition and 90.09% in instance-level recognition for a support radius of 12 cm whereas the second top performer CSHOT achieves 91.15% in category-level recognition for a support radius of 10 cm and 81.17% in instance-level recognition for a support radius of 5 cm.

Among the shape-only descriptors, in both category-level and instance-level recognition, SHOT performs slightly better than SPAIR, where the performance margin is larger for lower support radii and smaller for larger support radii. Among the tested descriptors, FPFH has the least performance for all support radius sizes in both category-level and instance-level recognition.

In addition to performance results, the confusion matrices for SHOT, SPAIR, CSHOT and CoSPAIR are given in Figure 5.14 and 5.15. When the matrices for the top two

performing descriptors, CSHOT and CoSPAIR are investigated in detail, it is observed that, in category level, even though it used color information, CSHOT tends to confuse similarly shaped categories even though the color of the categories are different, i.e., `lime` with `peach` and `potato`, `tomato` with `garlic` and `potato`. CoSPAIR is observed to make similar mistakes but with less percentage. In instance level, CoSPAIR shows significant strength on differentiating instances of the same category compared to CSHOT.

5.7 Results on Dataset 3: The BigBIRD Dataset

Since the BigBIRD dataset is an instance-level dataset and no category information is specified, only the instance-level recognition results are reported for this dataset. The average accuracy, average per class recall and precision results are given in Table 5.10. In addition, *leave-sequence-out* average accuracies are shown in Figure 5.16 to visualize the performance trend with respect to the support radius.

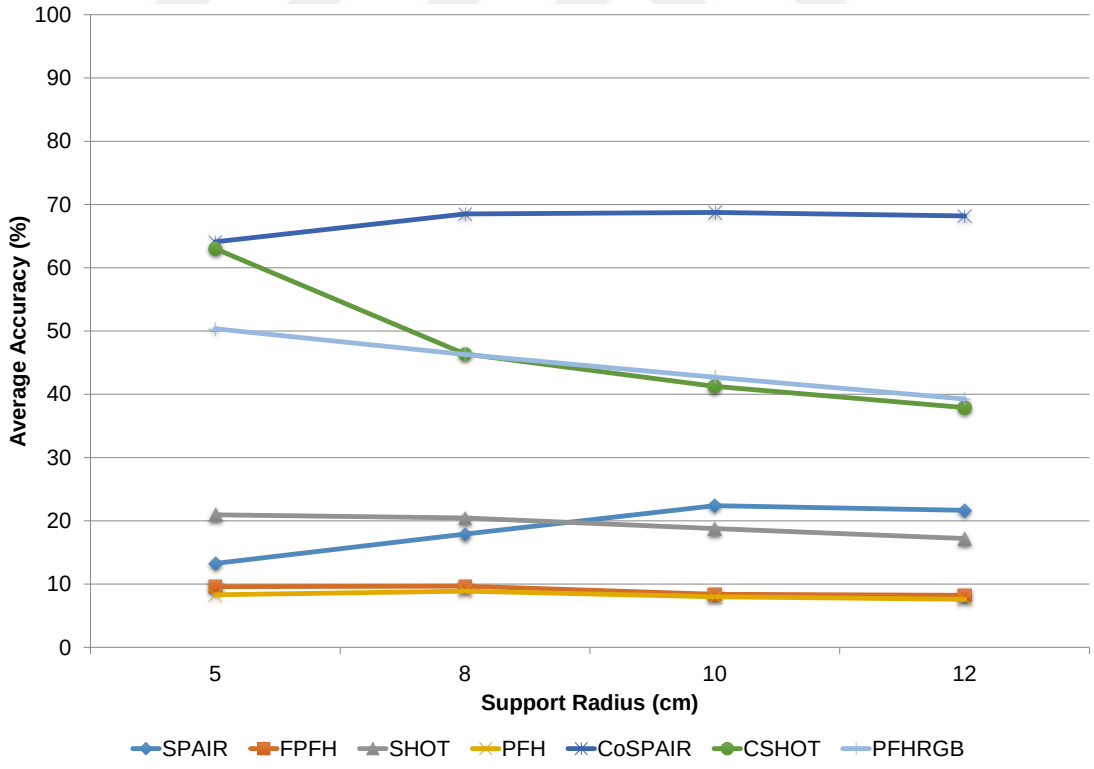
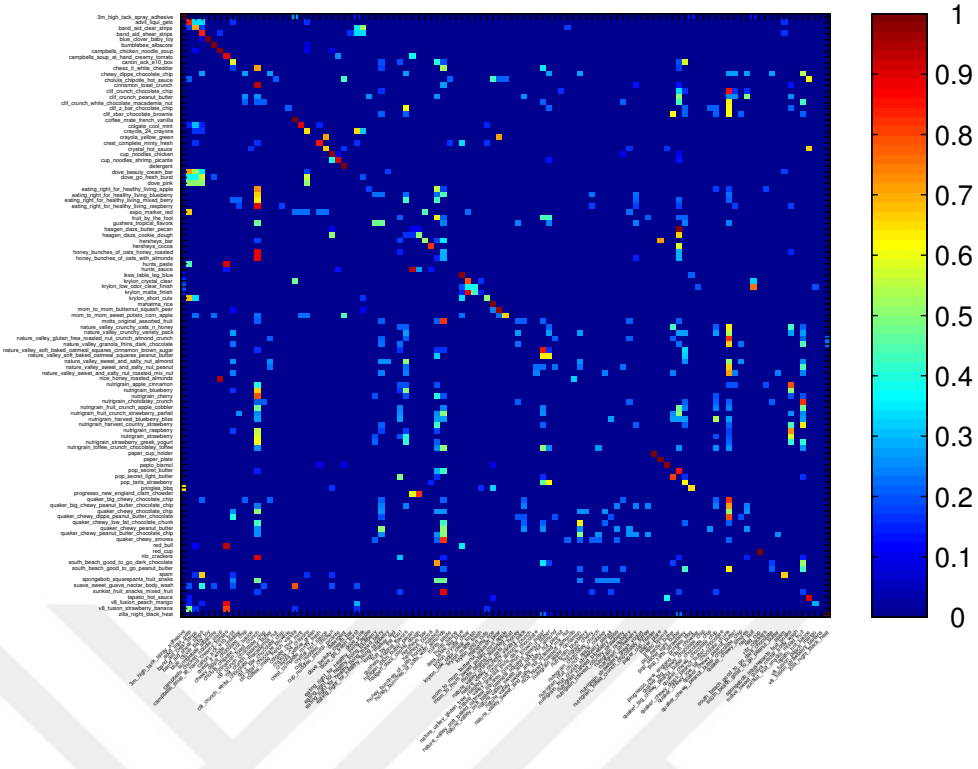


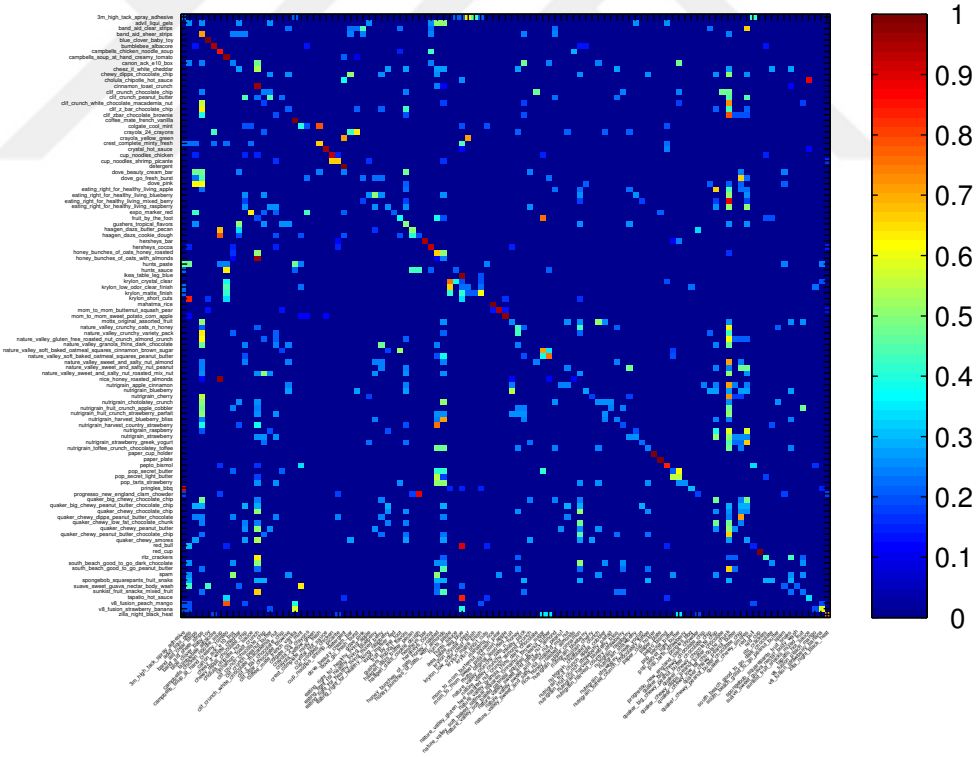
Figure 5.16: Instance-level average accuracy results for the BigBIRD dataset in *leave-sequence-out* scenario.

Table 5.10: Instance-level average accuracy, average recall and average precision results for the BigBIRD dataset.

	sr = 5cm			**sr = 8cm**			**sr = 10cm**			**sr = 12cm**		
						Leave-sequence-out						
	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.
SPAIR	13.27	13.14	12.34	17.91	17.75	18.61	22.38	22.17	22.24	21.66	21.45	20.86
FPFH	9.59	9.51	8.78	9.67	9.58	8.21	8.39	8.32	7.89	8.23	8.15	6.43
SHOT	20.94	20.77	24.18	20.46	20.29	22.11	18.79	18.63	19.93	17.19	17.05	16.16
PFH	8.31	8.21	6.43	8.87	8.79	7.35	7.99	7.91	8.46	7.59	7.52	5.46
CoSPAIR	**64.11**	**63.58**	**67.08**	**68.51**	**67.96**	**73.00**	**68.75**	**68.57**	**72.39**	**68.19**	**68.02**	**70.13**
CSHOT	62.99	62.48	64.85	46.36	46.00	50.07	41.25	41.31	46.79	37.89	37.59	40.39
PFHRGB	50.36	49.95	50.63	46.28	45.93	46.60	42.69	42.34	42.92	39.25	38.93	37.36
						Alternating contiguous frames						
SPAIR	24.31	24.14	27.43	33.61	33.43	35.72	36.79	36.70	38.82	37.96	37.86	40.13
FPFH	24.36	24.16	26.36	30.41	30.19	33.13	31.35	31.14	33.31	31.79	31.57	33.75
SHOT	35.88	35.65	38.35	42.27	42.19	43.25	43.52	43.39	44.56	43.88	43.77	45.00
PFH	21.85	21.71	24.67	23.41	23.22	25.98	24.14	23.94	27.92	23.46	23.27	27.43
CoSPAIR	**81.29**	**80.83**	**83.36**	**81.46**	**81.20**	**83.32**	**81.18**	**80.94**	**83.46**	**79.86**	**79.51**	**82.08**
CSHOT	64.93	64.54	67.80	62.44	62.12	65.88	61.96	61.67	65.30	61.55	61.21	64.71
PFHRGB	75.42	74.78	76.90	71.44	70.88	73.84	69.20	68.64	72.18	67.97	67.45	71.49

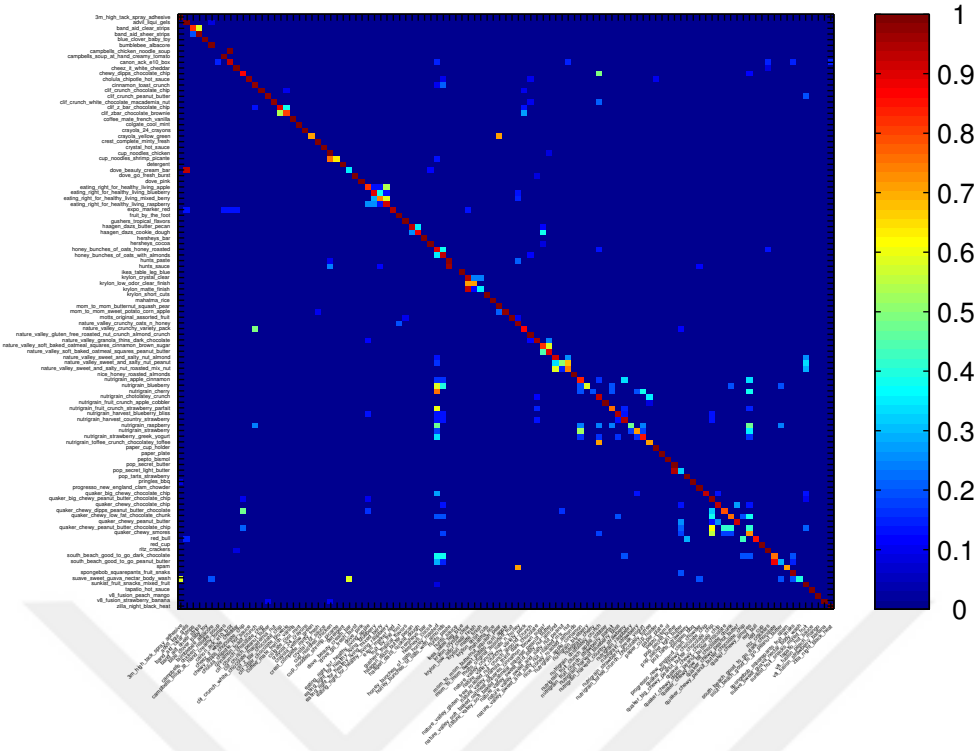


(a) SPAIR (sr = 10cm)

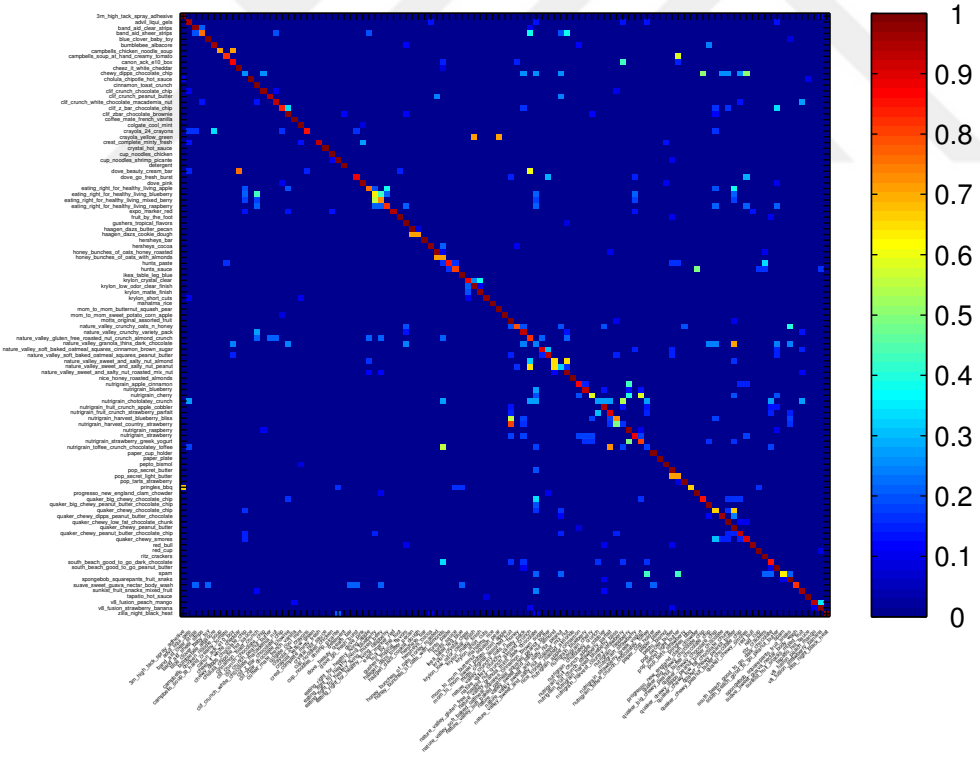


(b) SHOT (sr = 5cm)

Figure 5.17: Confusion matrices for the BigBIRD Dataset in *leave-sequence-out* scenario.



(c) CoSPAIR (sr = 10cm)



(d) CSHOT (sr = 5cm)

Figure 5.17: Confusion matrices for the BigBIRD Dataset in *leave-sequence-out* scenario (cont.)

Since this dataset is instance-level, and the difference between many instances are in texture/color (see Figure 5.6) shape-only descriptors perform extremely poor. However, the shape + texture/color descriptors perform fairly well considering the challenging nature of this dataset. The best performing descriptor is CoSPAIR for both test scenarios. For the *leave-sequence-out* case, CoSPAIR achieves 68.75% average accuracy for support radius of 10 cm whereas the second best performer CSHOT achieves 62.99% for support radius of 5 cm. For the *alternating-contiguous-frames* scenario, CoSPAIR outperforms competitors. It achieves 81.46% average accuracy at 8 cm whereas the second top performer PFHRGB achieves 75.42% for 5cm. Although the best achieved scores can be considered close, the performance gap increases with the increasing support radii. For the *leave-sequence-out* case, although the performance gap between CoSPAIR and CSHOT is 1.12 pp at 5 cm, the gap increases up to 30.3 pp at 12 cm. Lastly, for the *alternating-contiguous-frames* scenario, the performance gap is lowest, 16.36 pp at 5 cm and highest, 19.22 pp at 10 cm.

In addition to results, the confusion matrices for SHOT, SPAIR, CSHOT and CoSPAIR are given in Figure 5.17. When the matrices for the top two performing descriptors, CSHOT and CoSPAIR are investigated in detail, it is observed that CoSPAIR show particular strength generally on differentiating extremely similar objects (`eating right for healthy living`, `nature valley` and `nutrigrain` varieties). Such objects are shown in Figure 5.6.

5.8 Results on Dataset 4: The Amazon Picking Challenge Dataset

Like the BigBIRD, this dataset is an instance-level dataset and no category information is specified. Thus, only the instance-level recognition results are reported. For both scenarios, CoSPAIR performs better than the competitors for all the tested support radii. For *leave-sequence-out*, CoSPAIR achieves 90.71% average accuracy for support radius of 12 cm whereas the second best performer CSHOT achieves 85.90% for the same support radius. For *alternating contiguous frames* scenario, CoSPAIR achieves 91.63% average accuracy at 12 cm whereas the second top performer CSHOT achieves 87.36% for 10 cm.

In addition to results, the confusion matrices are given in Figure 5.19 as well. When the matrices for the top two performing descriptors, CSHOT and CoSPAIR are investigated in detail, it is observed that both CSHOT and CoSPAIR fails on `rollodex mesh collection jumbo pencil cup` most probably due to objects meshed surface that causes lack of data. CSHOT shows particular weakness on small boxy shaped `dove beauty bar` and confuses the object with similarly shaped `genuine joe plastic stir sticks` and `highland 6539 self stick notes` whereas CoSPAIR recognized this object with high accuracy. For the remaining objects, both CSHOT and CoSPAIR performs similarly.

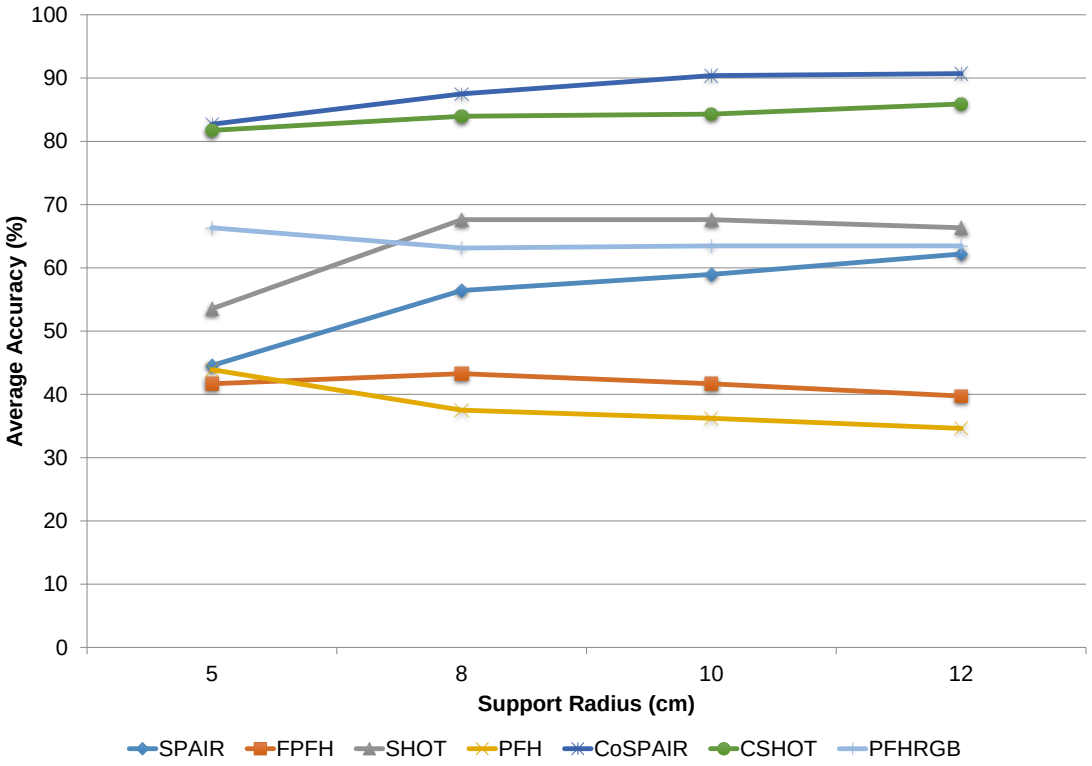
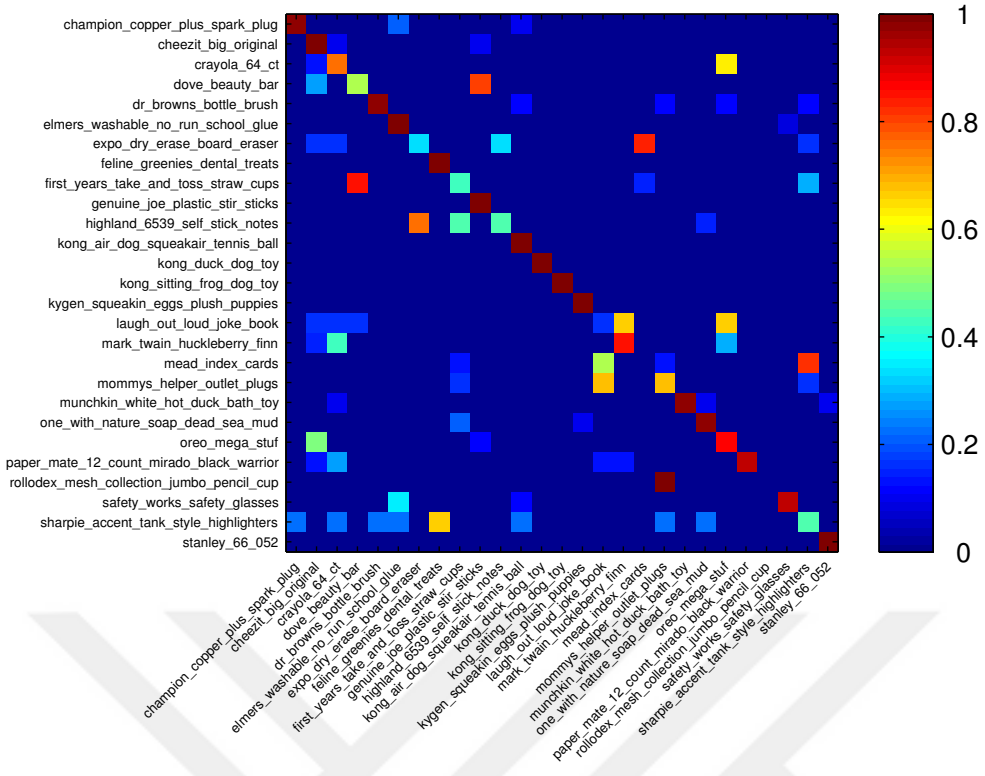


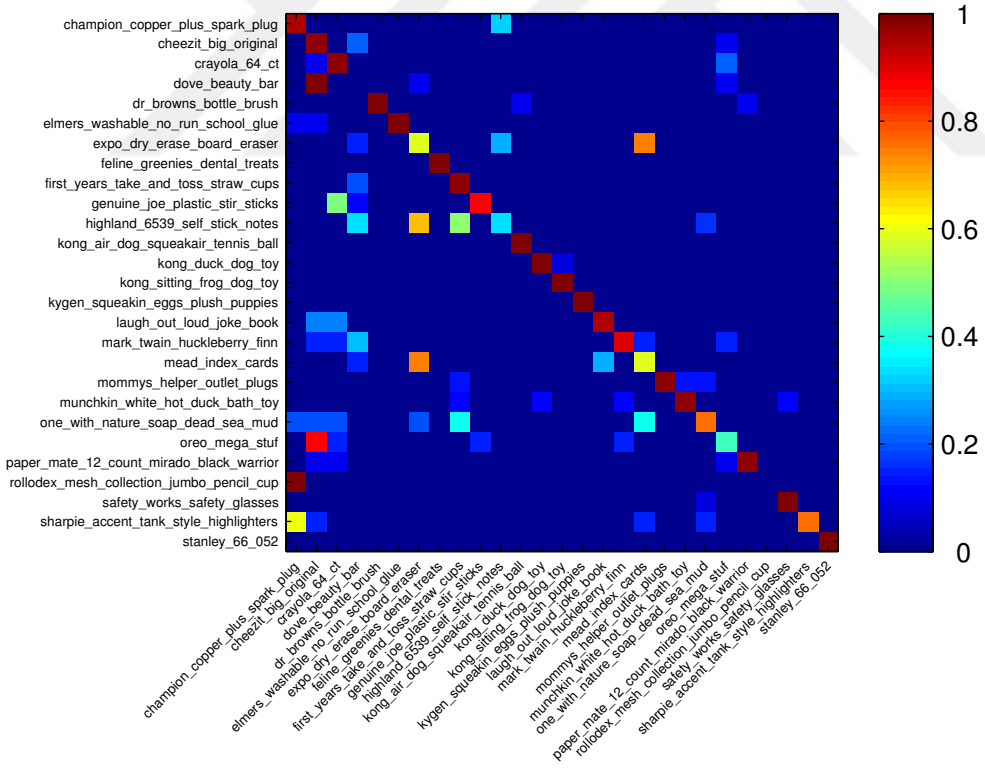
Figure 5.18: Instance-level average accuracy results for the Amazon Picking Challenge dataset in *leave-sequence-out* scenario.

Table 5.11: Instance-level average accuracy, average recall and average precision results for the Amazon Picking Challenge Dataset.

	sr = 5cm			**sr = 8cm**			**sr = 10cm**			**sr = 12cm**		
						Leave-sequence-out						
	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.	Acc.	Rec.	Prec.
SPAIR	44.55	42.66	39.14	56.41	54.15	54.09	58.97	56.73	55.61	62.18	59.81	59.80
FPFH	41.67	40.14	42.83	43.27	41.62	43.27	41.67	40.10	38.36	39.74	38.31	36.21
SHOT	53.53	51.61	51.32	67.63	65.19	68.43	67.63	65.25	67.71	66.35	64.01	68.46
PFH	43.91	42.25	45.07	37.50	36.06	31.97	36.22	34.83	33.26	34.62	33.33	31.39
CoSPAIR	**82.69**	**79.88**	**83.50**	**87.50**	**84.38**	**88.04**	**90.38**	**87.22**	**88.97**	**90.71**	**87.65**	**88.42**
CSHOT	81.73	78.95	81.16	83.97	81.05	82.20	84.29	81.36	82.13	85.90	82.90	83.72
PFHRGB	66.35	63.81	70.58	63.14	60.92	59.41	63.46	61.33	61.84	63.46	61.33	63.24
						Alternating contiguous frames						
SPAIR	46.18	45.29	49.17	60.49	59.45	62.80	64.89	63.80	65.57	67.10	66.05	68.07
FPFH	40.86	40.03	42.41	45.36	44.42	46.39	47.24	46.33	48.22	45.70	44.77	46.60
SHOT	62.51	61.59	63.56	75.00	74.02	76.22	75.82	74.73	76.63	76.06	75.03	76.40
PFH	42.45	41.67	41.44	42.83	42.15	44.07	42.07	41.35	43.33	40.44	39.95	41.65
CoSPAIR	**89.47**	**88.30**	**89.32**	**91.97**	**90.83**	**91.61**	**91.39**	**90.22**	**91.05**	**91.63**	**90.44**	**91.08**
CSHOT	84.06	82.99	84.45	86.88	85.79	86.68	87.36	86.26	87.58	87.02	85.92	87.12
PFHRGB	80.57	79.50	81.49	79.76	78.70	81.53	79.42	78.41	81.30	81.10	80.06	82.42

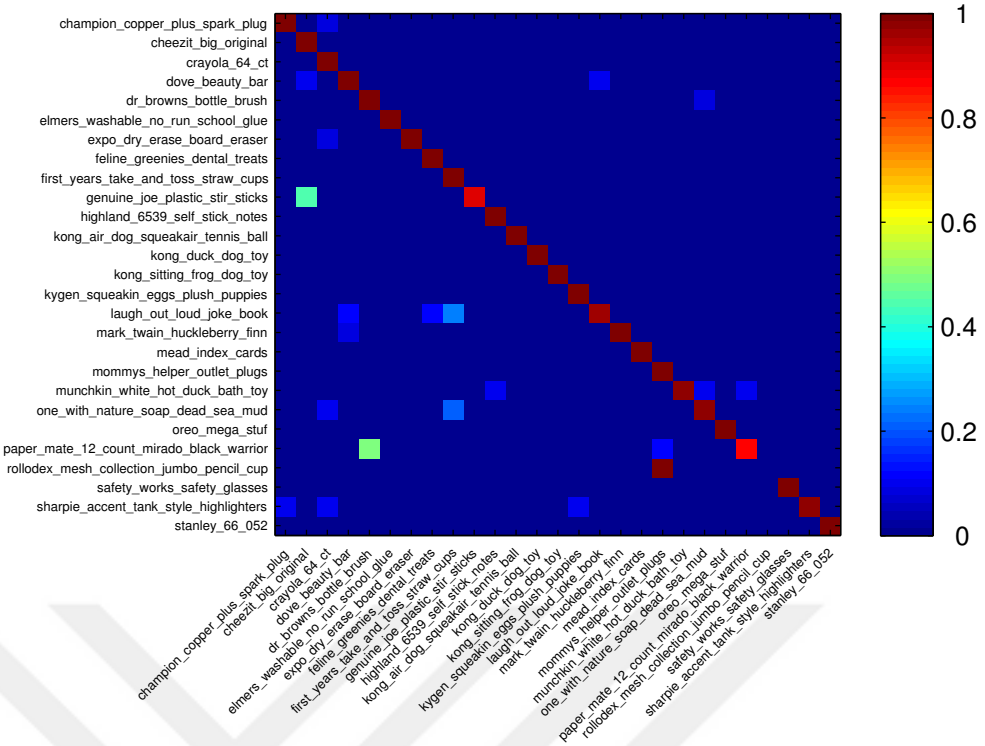


(a) SPAIR (sr = 12cm)

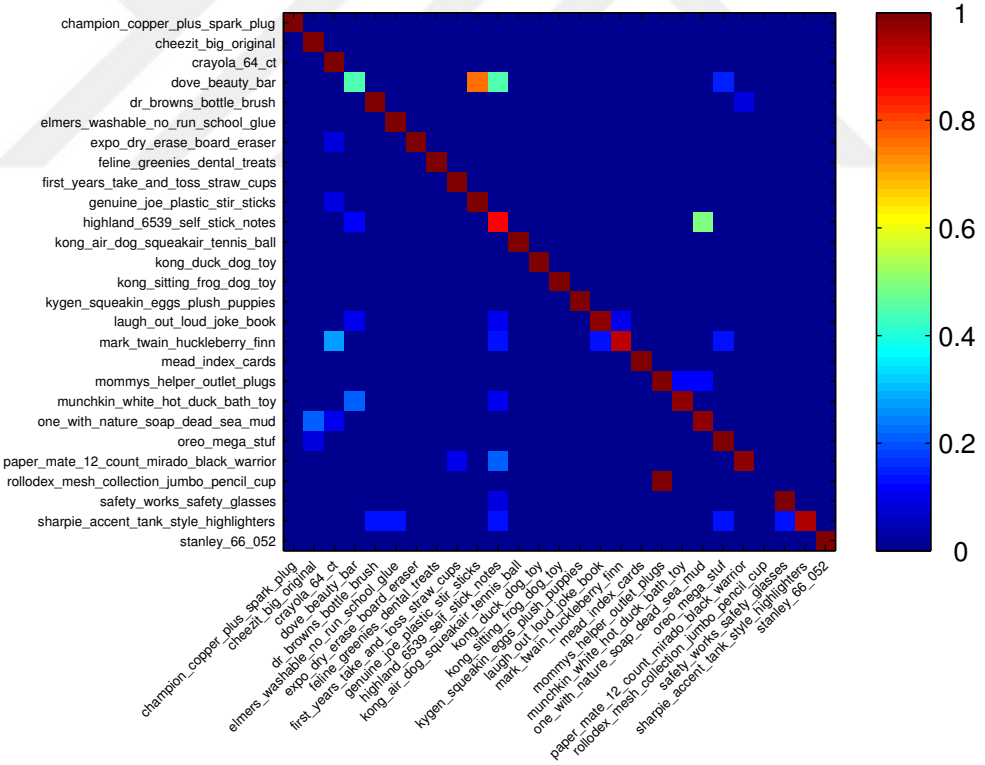


(b) SHOT (sr = 12cm)

Figure 5.19: Confusion matrices for the Amazon Picking Challenge dataset in *leave-sequence-out* scenario.



(c) CoSPAIR (sr = 12cm)



(d) CSHOT (sr = 12cm)

Figure 5.19: Confusion matrices for the Amazon Picking Challenge dataset in *leave-sequence-out* scenario (cont.)

5.9 Analysis of Extraction and Matching Times

For evaluating the extraction times, only a single scan for each category in the Dataset 1 with 1 cm uniform sampling is used. As a result, the query set for extraction times consists of 10 clouds with 3023 keypoints.

The average extraction times for a single keypoint/query point for 3 different support radius sizes are given in Table 5.12. As can be observed from the results, SHOT and CSHOT are very fast to extract whereas PFH and PFHRGB are prohibitively slow to use in practical applications. Moreover, while SPAIR and CoSPAIR are slower than SHOT, they are significantly faster than FPFH, PFH and PFHRGB. The main reason behind the speed of SHOT and CSHOT despite being longer is to use a single reference frame for each descriptor whereas SPAIR, CoSPAIR, PFH and RGBPFH fit a reference axis for each pair of points between which angular relations are computed.

Table 5.12: Average extraction times (ms) of the descriptors for a single keypoint/-query point. (Platform: i5 4670 CPU using a single core)

	sr=5cm	sr=10cm	sr=12cm
SPAIR	4.37	11.98	15.23
FPFH	16.83	49.22	63.53
SHOT	1.27	2.55	3.10
PFH	506.50	5456.31	9409.57
CoSPAIR	5.37	14.27	18.22
CSHOT	1.45	3.96	5.04
PFHRGB	918.67	10049.05	17304.95

And lastly, the brute-force matching times together with the size of the descriptors are given in Table 5.13. In this test, the full reference and query sets in the Dataset 1 were used where the query set contains 78,442 keypoints from 475 objects and the reference set contains 143,234 keypoints from 946 objects, thus the total number of comparisons were over 11 billion. Since the same matching method is used for all descriptors, the matching time is mainly related to the type and the length of the descriptors. As all the descriptors are of type *float*, descriptor length is the only factor affecting the matching performance. This can be directly seen from the results that FPFH, being the shortest descriptor, is the fastest to match and CSHOT, being the

largest, is the slowest to match.

Table 5.13: Lengths and matching times (seconds) of the descriptors. (Platform: i5 4670 CPU utilizing all 4 cores)

	Descriptor Length	Matching Time (s)
SPAIR	189	119
FPFH	33	34
SHOT	392	170
PFH	125	88
CoSPAIR	378	197
CSHOT	1344	581
PFHRGB	250	136

5.10 Performance vs. Size

The performance of the descriptors is further investigated to analyze the effect of size of objects. For this purpose, the objects are categorized into 5 depending on their sizes; 0-10 cm, 10-15 cm, 25-20 cm, 20-25 cm and 25+ cm. The distribution of objects onto these categories are given in Table 5.14.

Table 5.14: Object sizes in datasets

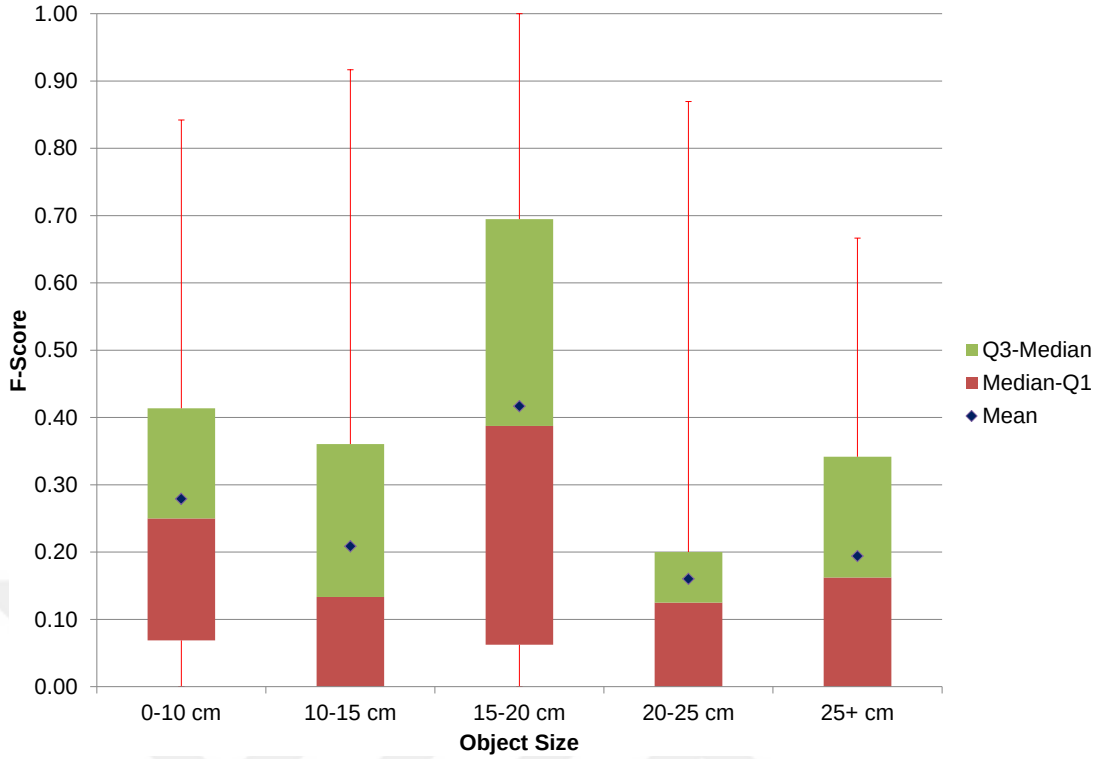
	Object Count		
Object Size	RGB-D	BigBIRD	Amazon
0-10 cm	83	4	4
10-15 cm	98	28	10
15-20 cm	42	28	7
20-25 cm	41	39	3
25+ cm	36	6	3
Total	300	105	27

The *box-and-whisker* plots for SHOT, SPAIR, CSHOT and CoSPAIR descriptors for the RGB-D dataset is given in Figure 5.20 and for the BigBIRD dataset in Figure 5.21. The used metric *F-Score* is directly calculated from the previously given *recall*

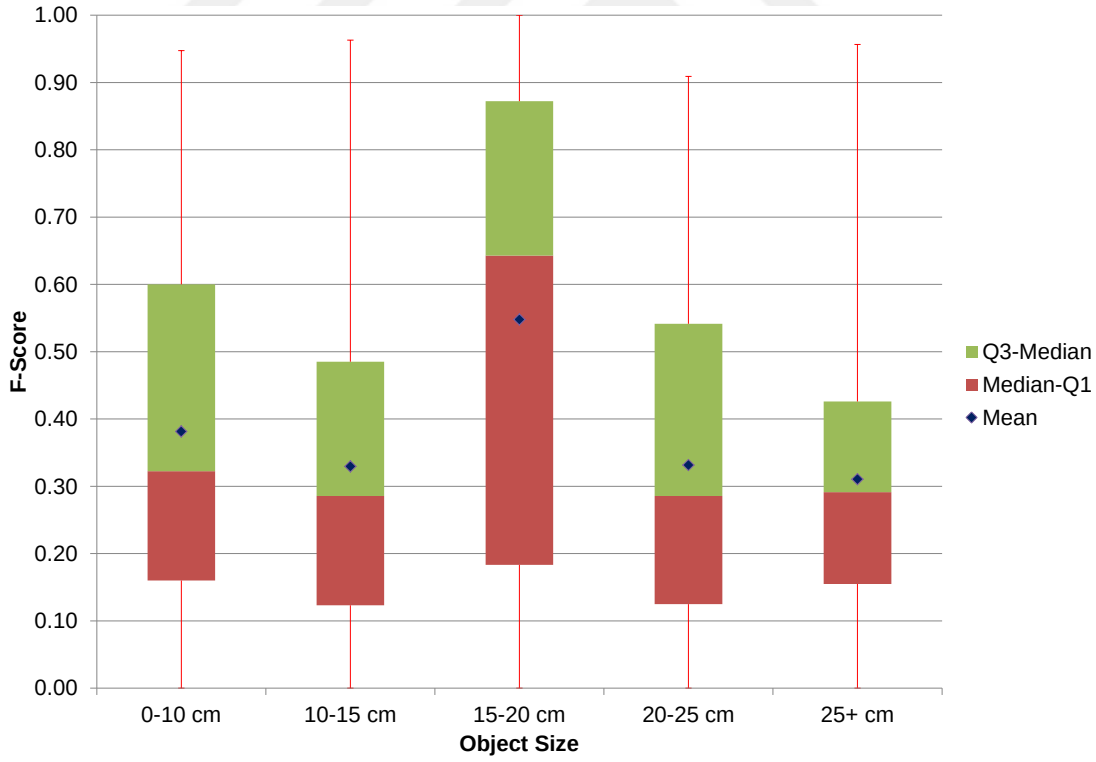
and *precision* values:

$$F - Score = 2 \times \frac{precision \times recall}{precision + recall}. \tag{5.3}$$

It can be observed from results that the proposed SPAIR and CoSPAIR descriptors behave very similar to SHOT and CSHOT. In RGB-D dataset, the tested descriptors perform best for medium sized (15-20 cm) objects without showing any significant weakness for particular object size whereas in BigBIRD dataset, they perform marginally for large objects (25+ cm), once again without showing any significant weakness in any object size.

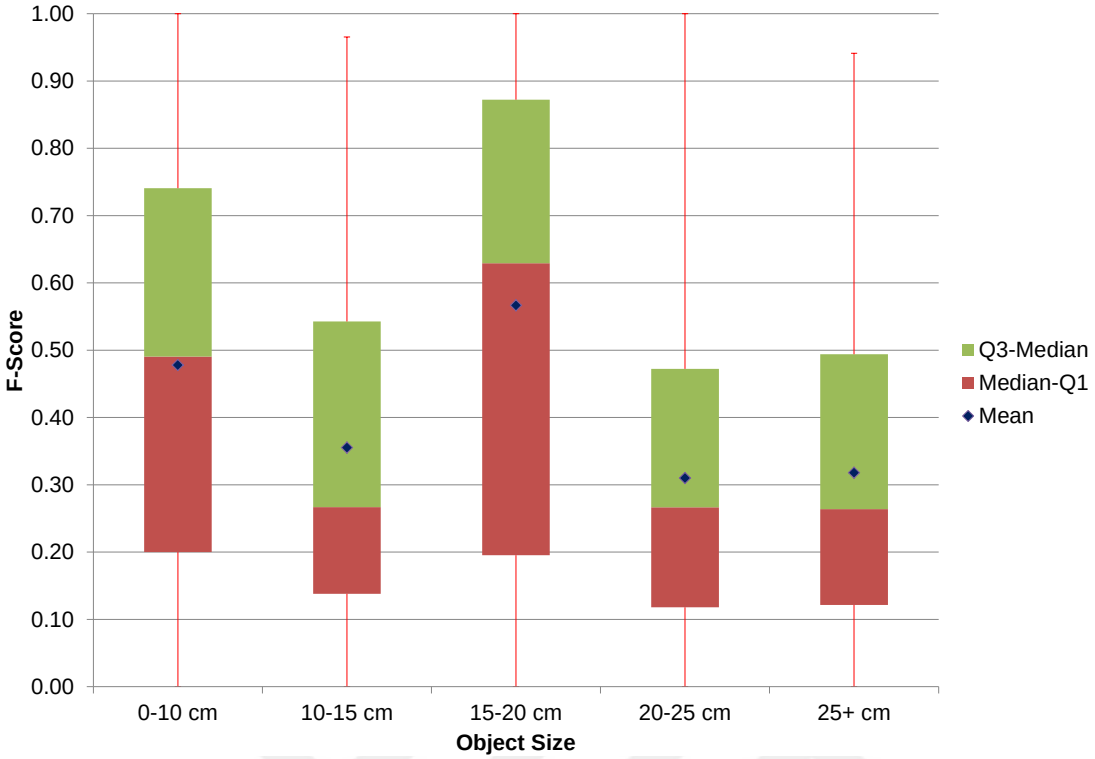


(a) SPAIR (sr=5cm)

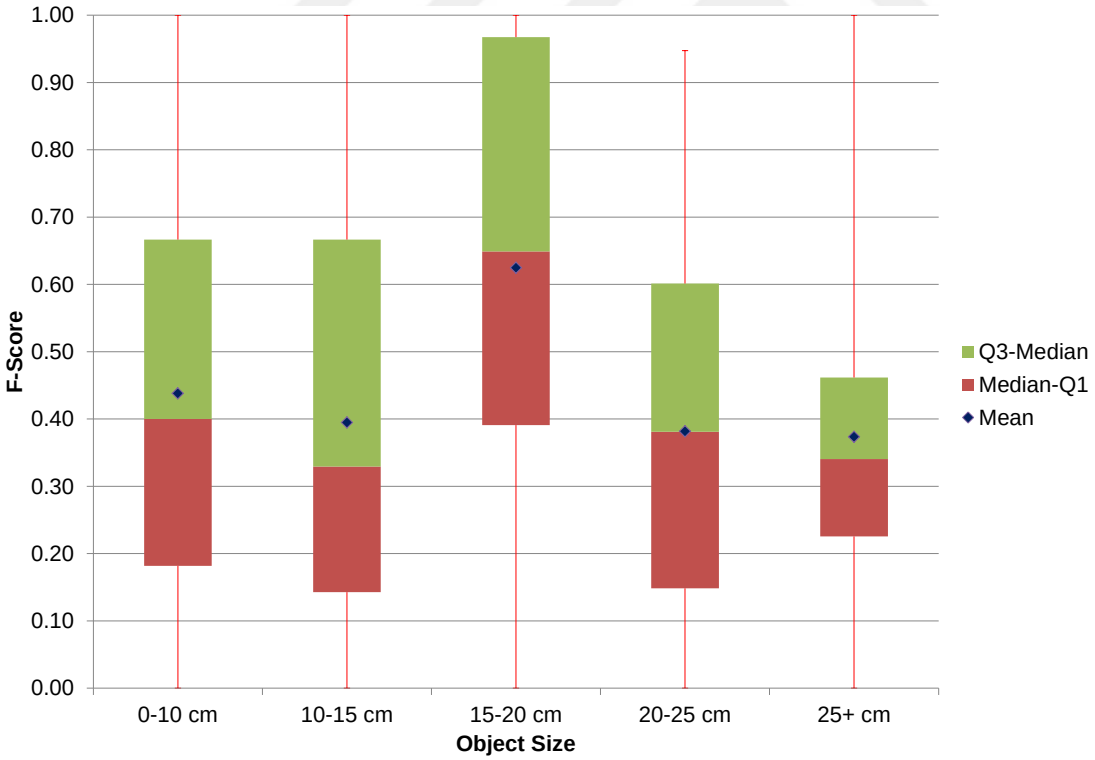


(b) SHOT (sr=5cm)

Figure 5.20: RGBD F-Score vs Size

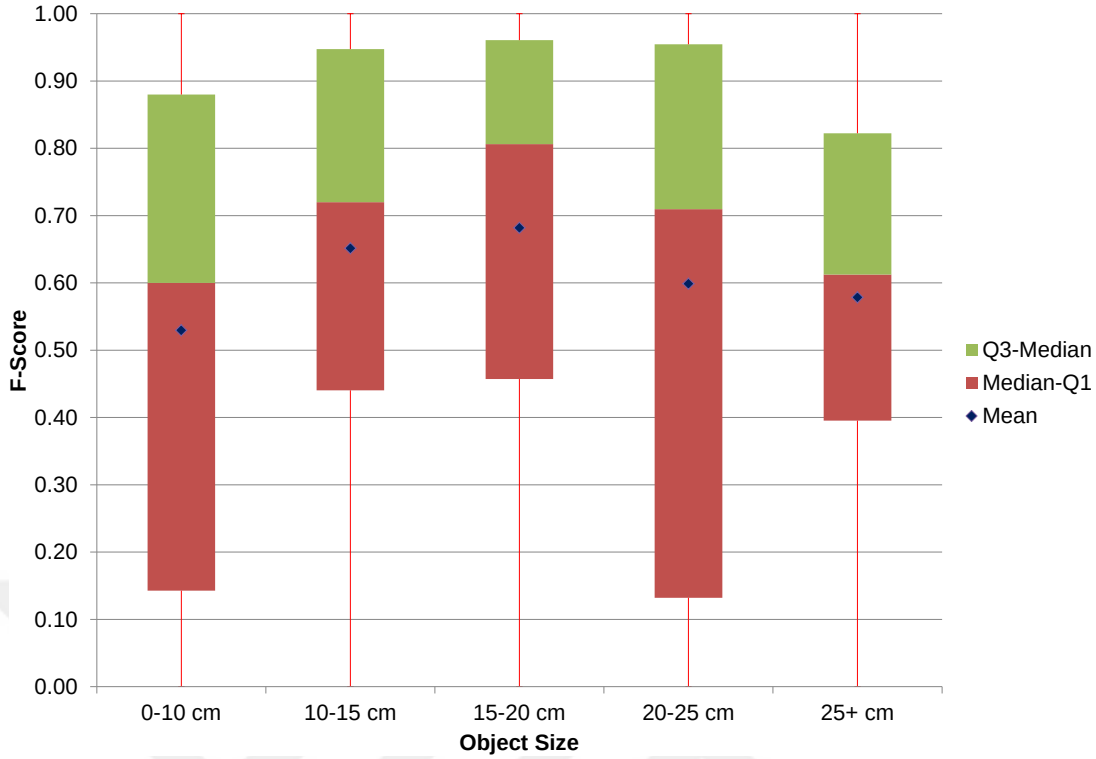


(c) SPAIR (sr=10cm)

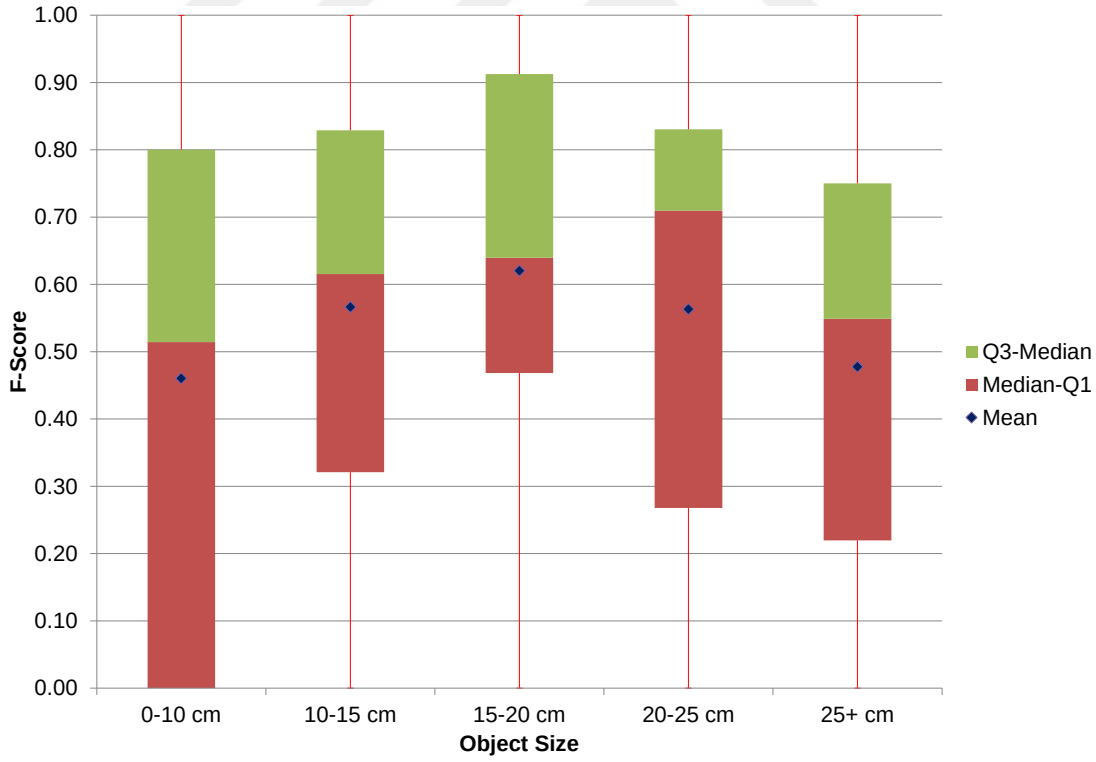


(d) SHOT (sr=10cm)

Figure 5.20: RGBD F-Score vs Size (cont.)

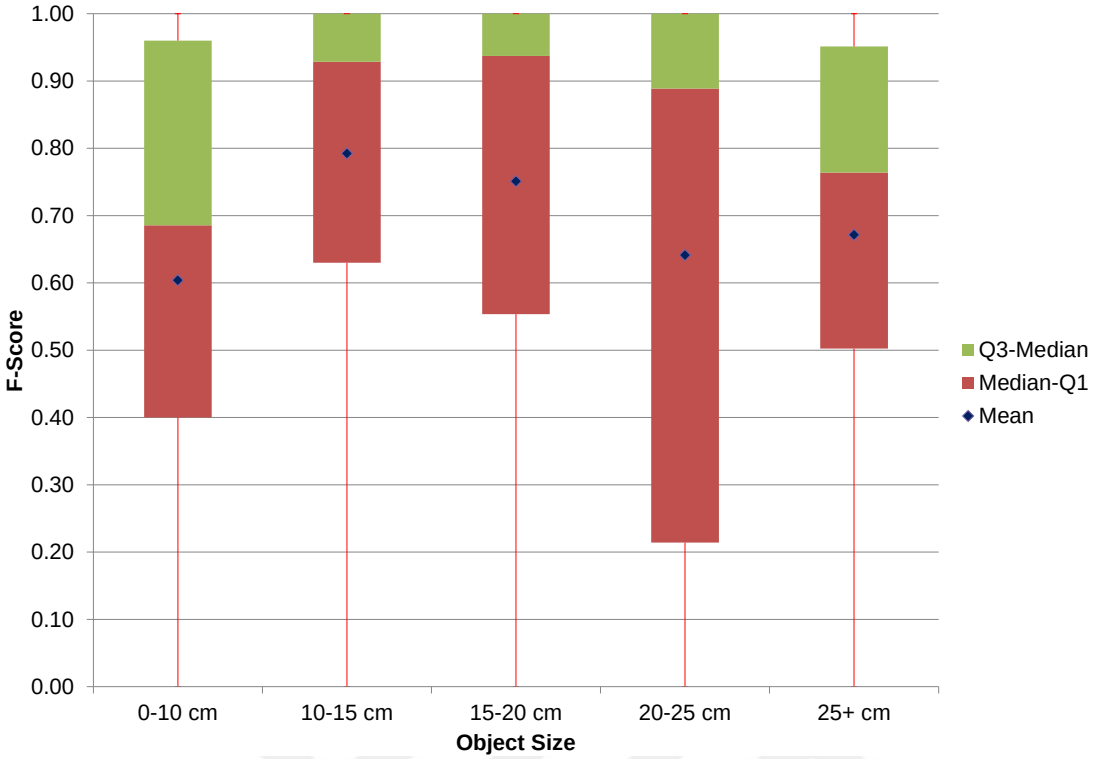


(e) CoSPAIR (sr=5cm)

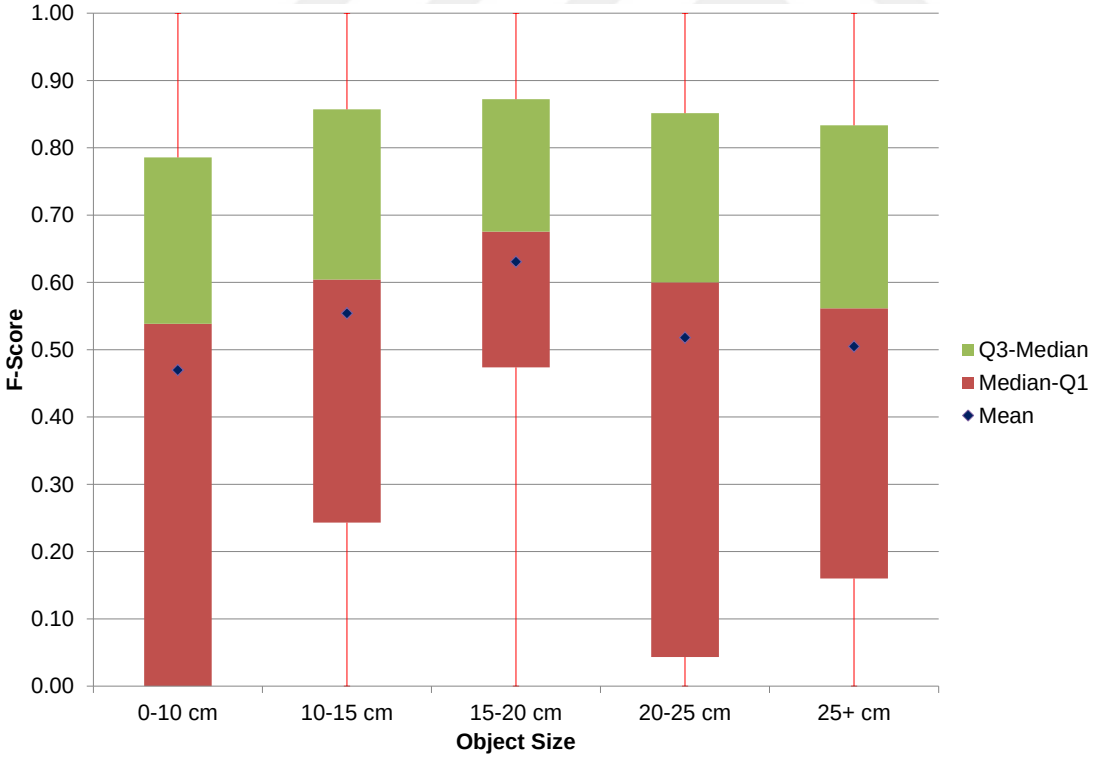


(f) CSHOT (sr=5cm)

Figure 5.20: RGBD F-Score vs Size (cont.)

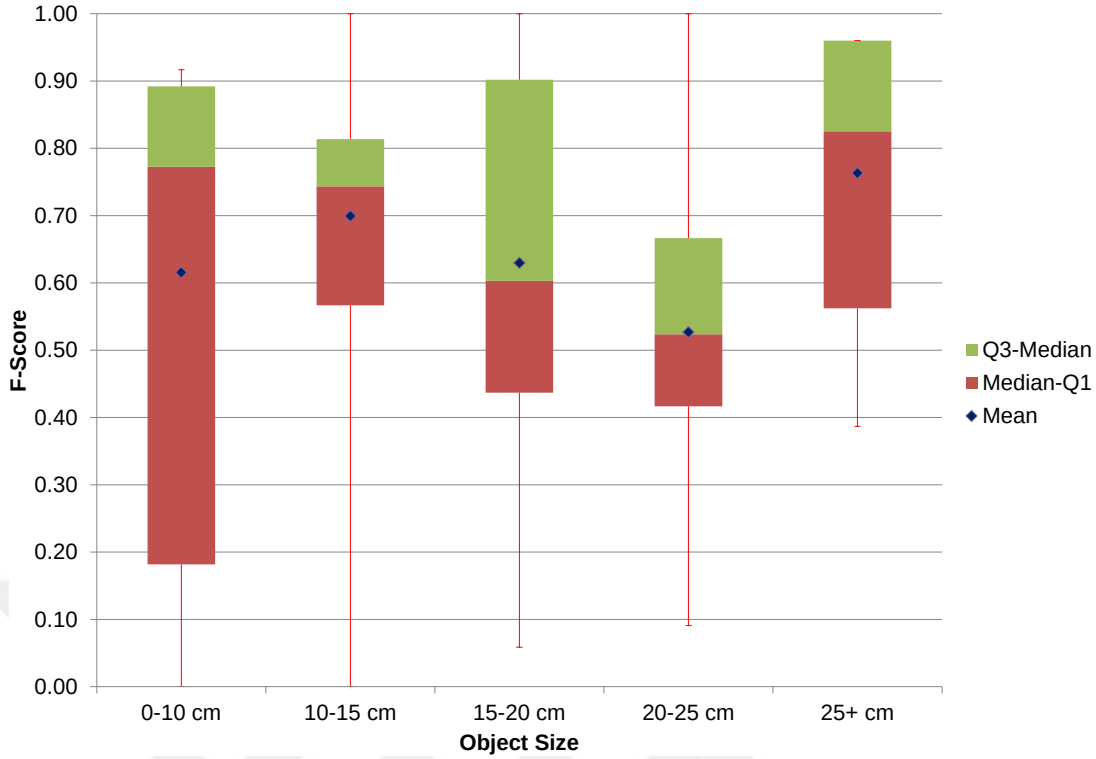


(g) CoSPAIR (sr=10cm)

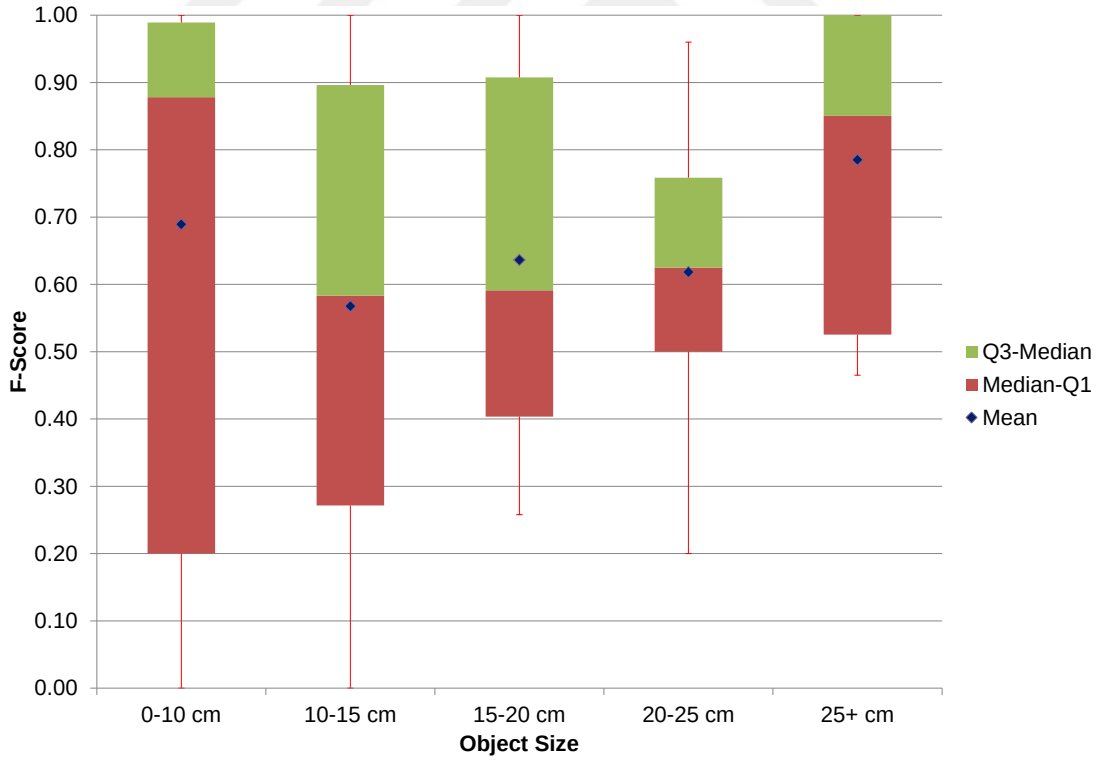


(h) CSHOT (sr=10cm)

Figure 5.20: RGBD F-Score vs Size (cont.)



(a) CSHOT (sr=5cm)



(b) CoSPAIR (sr=5cm)

Figure 5.21: BigBIRD - F-Score vs Size

CHAPTER 6

CONCLUSION

In recent years, research on 3D robot/computer vision, consequently 3D object recognition have been boosted due to advancements on 3D sensor technology especially the ones that simultaneously capture RGB and depth information. Although significant amount of 3D descriptors exist, not many of them (please refer to Table 2.1) take advantage of this technology (i.e., they lack color information). In this thesis, two novel local 3D descriptors are proposed; one that utilizes only shape information - *Histograms of Spatial Concentric Surflet-Pairs* (SPAIR) and a complimentary one that jointly utilizes shape and color information - *Colored Histograms of Spatial Concentric Surflet-Pairs* (CoSPAIR) that takes advantage of the RGB-D sensors. In these descriptors, the support radius is divided into concentric spherical shells and histograms of angular relations between surface normals are utilized. It has been demonstrated that such partitioning of space allows encoding enhanced spatial information more effectively. Ultimately, we have shown that the object recognition performance, especially in instance-level, can be significantly improved using the proposed descriptor CoSPAIR.

In Chapter 5, the proposed descriptors have been compared with the state-of-the-art local 3D descriptors that are available in the Point Cloud Library on three different publicly available object recognition datasets. The shape-only descriptor, SPAIR is shown to be one of the best in its class (shape-only) while CoSPAIR is shown to outperform the tested state-of-the-art descriptors both for category-level and instance-level object recognition. Up to 9.9 percentage points gain in category-level recognition and 16.49 percentage points gain in instance-level recognition over the second-

best performing descriptor in the RGB-D dataset have been observed. Additionally, it has been demonstrated that CoSPAIR, compared to the second best performer CSHOT, can differentiate very similar objects (same shape, different texture) more effectively.

6.1 Future Work

The descriptors proposed in this thesis can be improved in a number of ways in the future. The first of these improvements is regarding the spatial information that is integrated into the descriptors. In the proposed descriptors, the support radius is divided into equally-sized shells. Since the contribution of the central shells and the outer shells may be different, one may consider shells having varying thickness and learning the optimal division of the support radius into the shells. Another similar line of work to extend the system might be to use weighted combination of the histograms coming from different shells.

Secondly, the information extracted in each shell can be extended by other 3D or 2D information, such as, local 3D curvature, local 3D shape category via the method of shape index, 2D textural features. These can enrich the representation in each shell, and hence, effect the overall performance.

Thirdly, in this thesis, performance is evaluated for various "support radii" which corresponds to "scale' in 2D. It should be noted that the definition of a proper scale in 3D data differs from the scale-invariance concept of 2D features because of the metric data provided by 3D sensors [48]. Since the performance of the proposed descriptors depend on the chosen "support radius", to overcome such a limitation, multi-scale feature extraction techniques [28, 57, 58] might be applied.

Additionally, as stated in Chapter 4, the descriptors are extracted from all the detected keypoints. Instead of this approach, extracting the features from "salient regions/points" [59, 60] can be considered. Although there are many work on "saliency detection", recently Schtrom et al. extended this approach to 3D surfaces and point sets [61]. Their proposed method is highly efficient and competitive thus applying this method to detect salient regions might improve recognition performance consid-

erably.

Similarly, another possible research direction would be first detecting meaningful regions on the point cloud and applying "region covariance" technique proposed by Tuzel et al. [62]. This technique would provide fusion of multiple features which might be correlated and filter out the noise, resulting much more compact feature vectors with possibly better recognition performance.

Furthermore, the performance of the descriptors can be evaluated via training them with "Support Vector Machines" (SVM), instead of brute-force matching to the nearest descriptor in the reference descriptor database as explained in Section 5.1. Using SVM for training and matching would provide faster recognition times which is crucial in many real world applications.

Last but not least, we will share our descriptors with the robot / computer vision community through the Point Cloud Library (PCL) [17] to enable further research on this field.

REFERENCES

[1] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[2] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *Computer Vision-ECCV 2004*, pages 224–237. Springer, 2004.

[3] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014.

[4] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2155–2162, 2010.

[5] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1817–1824, 2011.

[6] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: (big) berkeley instance recognition dataset. `http://rll.berkeley.edu/bigbird/`.

[7] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 509–516, 2014.

[8] Arjun Singh, James Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Amazon picking challenge dataset. `http://rll.berkeley.edu/amazon_picking_challenge/`.

[9] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[10] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, and Jianwei Wan. 3d object recognition in cluttered scenes with local surface features: A survey.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(11):2270–2287, 2014.

[11] Intel realsense camera. http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-depth-camera.html.

[12] Google project tango. https://www.google.com/atap/project-tango/.

[13] Infineon 3d image sensor real3. http://www.infineon.com/cms/en/product/sensor/3d-image-sensor-real3/channel.html?channel=5546d4614937379a0149382e3e960078.

[14] Velodyne hdl-64e lidar. http://velodynelidar.com/hdl-64e.html.

[15] Trimble tx5. http://www.trimble.com/3d-laser-scanning/tx5.aspx.

[16] Luıs A Alexandre. 3d descriptors for object and category recognition: a comparative evaluation. In *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal*. Citeseer, 2012.

[17] The point cloud library (pcl). http://www.pointclouds.org/.

[18] Aitor Aldoma, Zoltan-Csaba Marton, Federico Tombari, Walter Wohlkinger, Christian Potthast, Bernhard Zeisl, Radu Bogdan Rusu, Suat Gedikli, and Markus Vincze. Point cloud library. *IEEE Robotics & Automation Magazine*, 1070(9932/12), 2012.

[19] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[20] Ceyhun Burak Akgul, Bülent Sankur, Yücel Yemez, and Francis Schmitt. 3d model retrieval using probability density-based shape descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1117–1133, 2009.

[21] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Computer Vision–ECCV 2010*, pages 356–369. Springer, 2010.

[22] Andrew Edie Johnson. *Spin-images: a representation for 3-D surface matching*. PhD thesis, Citeseer, 1997.

[23] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE International Conference on Robotics and Automation, ICRA'09*, pages 3212–3217, 2009.

[24] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, pages 3384–3391, 2008.

[25] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–696, 2009.

[26] Ajmal Mian, Mohammed Bennamoun, and Robyn Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2-3):348–361, 2010.

[27] Jan Knopp, Mukta Prasad, Geert Willems, Radu Timofte, and Luc Van Gool. Hough transform and 3d surf for robust three dimensional classification. In *Computer Vision–ECCV 2010*, pages 589–602. Springer, 2010.

[28] Andrei Zaharescu, Edmond Boyer, Kiran Varanasi, and Radu Horaud. Surface feature detection and description with applications to mesh matching. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 373–380, 2009.

[29] Federico Tombari, Samuele Salti, and Luigi Di Stefano. A combined texture-shape descriptor for enhanced 3d feature matching. In *18th IEEE International Conference on Image Processing (ICIP)*, pages 809–812, 2011.

[30] Aitor Aldoma, Markus Vincze, Nico Blodow, David Gossow, Suat Gedikli, Radu Bogdan Rusu, and Gary Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 585–592, 2011.

[31] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Transactions on Graphics (TOG)*, 21(4):807–832, 2002.

[32] Walter Wohlkinger and Markus Vincze. Ensemble of shape functions for 3d object classification. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2987–2992, 2011.

[33] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, pages 156–164, 2003.

[34] Petros Daras, Dimitrios Zarpalas, Dimitrios Tzovaras, Michael G Strintzis, D Tzovaras, and MG Strintzis. Shape matching using the 3d radon transform. In *Proceedings of 2nd International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, pages 953–960. IEEE, 2004.

[35] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. In *Computer graphics forum*, volume 22, pages 223–232. Wiley Online Library, 2003.

[36] Tony Tung and Francis Schmitt. The augmented multiresolution reeb graph approach for content-based retrieval of 3d shapes. *International Journal of Shape Modeling*, 11(01):91–120, 2005.

[37] Cheuk Yiu Ip, Daniel Lapadat, Leonard Sieger, and William C Regli. Using shape distributions to compare solid models. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 273–280. ACM, 2002.

[38] Ryutarou Ohbuchi, Kunio Osada, Takahiko Furuya, and Tomohisa Banno. Salient local visual features for shape-based 3d model retrieval. In *IEEE International Conference on Shape Modeling and Applications, SMI*, pages 93–102, 2008.

[39] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[40] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique shape context for 3d data description. In *Proceedings of the ACM workshop on 3D object retrieval*, pages 57–62. ACM, 2010.

[41] Eric Wahl, Ulrich Hillenbrand, and Gerd Hirzinger. Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification. In *IEEE Fourth International Conference on 3-D Digital Imaging and Modeling, 3DIM 2003*, pages 474–481, 2003.

[42] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for rgb-d based object recognition. In *Experimental Robotics*, pages 387–402. Springer, 2013.

[43] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[44] Donald Meagher. Geometric modeling using octree encoding. *Computer graphics and image processing*, 19(2):129–147, 1982.

[45] Radu Bogdan Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[46] Piotr Indyk. Chapter 39: Nearest neighbors in high-dimensional spaces. *Handbook of Discrete and Computational Geometry (2nd ed.)*, 2004.

[47] Klaas Klasing, Daniel Althoff, Dirk Wollherr, and Martin Buss. Comparison of surface normal estimation methods for range sensing applications. In *IEEE International Conference on Robotics and Automation. ICRA'09.*, pages 3206–3211. IEEE, 2009.

[48] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision*, 102(1-3):198–220, 2013.

[49] Silvio Filipe and Luıs A Alexandre. A comparative evaluation of 3d keypoint detectors. In *9th Conference on Telecommunications, Conftele*, pages 145–148, 2013.

[50] Sılvio Filipe and Luıs A Alexandre. A comparative evaluation of 3d keypoint detectors in a rgb-d object dataset. In *9th International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, 2014.

[51] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[52] Gareth James. *Majority vote classifiers: theory and applications*. PhD thesis, Stanford University, 1998.

[53] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[54] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[55] Luıs A Alexandre. Exact list of rgb-d dataset subset. `http://www.di.ubi.pt/~lfbaa/files/train_test_file_ids.tar.gz`.

[56] Karthik S Narayan, James Sha, Arjun Singh, and Pieter Abbeel. Range sensor and silhouette fusion for high-quality 3d scanning. *sensors*, 32(33):26, 2015.

[57] Erdem Akagündüz and Ilkay Ulusoy. 3d object recognition from range images using transform invariant object representation. *Electronics Letters*, 46(22):1499–1500, 2010.

[58] Hadi Fadaifard and George Wolberg. Multiscale 3d feature extraction and matching. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 228–235. IEEE, 2011.

[59] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[60] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[61] Elizabeth Shtrom, George Leifman, and Avishay Tal. Saliency detection in large point sets. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3591–3598. IEEE, 2013.

[62] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *Computer Vision–ECCV 2006*, pages 589–600. Springer, 2006.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Loğoğlu, K.Berker
Nationality: Turkish (TC)
Date and Place of Birth: 15.11.1982, Gebze
Marital Status: Married
Phone: 0 532 6704955
E-mail: berkerlogoglu@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
M.S.	Electrical and Electronics Engineering Department, METU	2007
B.S.	Electrical and Electronics Engineering Department, METU	2004

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
March 2013 - Present	ARGELA	Senior Engineer
December 2012 - March 2013	Turk Telekom	Senior Engineer
April 2006 - March 2013	TUBITAK - UZAY	Researcher / Senior Researcher

PUBLICATIONS

Journal

[1] K. Berker Logoglu, Sinan Kalkan, Alptekin Temizel, "CoSPAIR: Colored Histograms of Spatial Concentric Surflet-Pairs for 3D object recognition", Robotics and Autonomous Systems, Volume 75, Part B, January 2016, Pages 558-570, ISSN 0921-8890, http://dx.doi.org/10.1016/j.robot.2015.09.027.

[2] Soysal, Medeni, K. Berker Loğoğlu, Mashar Tekin, Ersin Esen, Ahmet Saracoğlu, Banu Oskay Acar, Ezgi Can Ozan et al. "Multimodal concept detection in broadcast media: KavTan." Multimedia Tools and Applications: 1-46.

Book Chapter

[3] Temizel, A., T. Halici, B. Logoglu, T. T. Temizel, F. Omruuzun, and E. Karaman. "Experiences on image and video processing with CUDA and OpenCL." GPU Computing Gems, Morgan Kaufmann, Boston (2011): 547- 567.

Proceedings

[4] Tekin, Mashar, A. Saracoglu, Ersin Esen, Medeni Soysal, B. Logoglu, Hakan Sevimli, T. K. Ates et al. "Multimodal concept detection on multimedia data-RTUK SKAAS KavTan system." In Signal Processing and Communications Applications Conference (SIU), 2012 20th, pp. 1-4. IEEE, 2012.

[5] Loğoğlu, K. Berker, Ahmet Saracoğlu, Ersin Esen, and A. Aydin Alatan. "Gender Classification via Gradientfaces." In Computer and Information Sciences, pp. 245-251. Springer Netherlands, 2010.

[6] Sevimli, Hakan, Ersin Esen, Tuğrul K. Ateş, Ezgi C. Ozan, Mashar Tekin, K. Berker Loğoğlu, Ayça Müge Sevinç, Ahmet Saracoğlu, Adnan Yazici, and A. Aydin Alatan. "Adult image content classification using global features and skin region detection." In Computer and Information Sciences, pp. 253-258. Springer Netherlands,

2010.

[7] Loğoğlu, K. B., and T. K. Ateş. "Speeding-up Pearson Correlation Coefficient calculation on graphical processing units." In Signal Processing and Communications Applications Conference (SIU), 2010 IEEE 18th, pp. 840-843. IEEE, 2010.

[8] Saracoğlu, A., M. Tekin, E. Esen, M. Soysal, K. B. Loğoğlu, T. K. Ateş, A. M. Sevinç et al. "Generalized visual concept detection." In Signal Processing and Communications Applications Conference (SIU), 2010 IEEE 18th, pp. 621-624. IEEE, 2010.

[9] Saracoğlu, Ahmet, Ersin Esen, Medeni Soysal, Tuğrul K. Ateş, Berker Loğoğlu, Mashar Tekin, Talha Karadeniz et al. "TÜBİTAK UZAY at TRECVID 2009: High-Level Feature Extraction and Content-Based Copy Detection."