DENSITY-BASED SHAPE DESCRIPTORS AND SIMILARITY LEARNING FOR 3D OBJECT RETRIEVAL

Ceyhun Burak Akgül

submitted to

Ecole Nationale Supérieure des Télécommunications, Paris
&
Boğaziçi University, Istanbul

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

December 2007

"Naturally, the four mathematical operations - adding, subtracting, multiplying, and dividing - were impossible. The stones resisted arithmetic as they did the calculation of probability. Forty disks, divided, might become nine; those nine in turn divided might yield three hundred."

Jorge Luis Borges – Blue Tigers

Acknowledgements

I am very grateful to my two thesis supervisors, Bülent Sankur and Francis Schmitt, and my co-supervisor Yücel Yemez for all they brought to this work, for expressing their critical minds on every occasion, and especially for being the *devil's advocate*. I've heard this expression for the first time from Bülent Sankur, now a long time ago. So you can imagine how much I was surprised when I heard it again from Francis Schmitt at my arrival to ENST[1]. This was an indication for the things to come, an indication for the "future" which is now the "past" that I will never regret. The difficulty and the joy of making a thesis in two institutions is to satisfy more than one person with his work, but also to profit from more than one exceptional mind to perfect it. I had this very chance of working with these three people who have shown me different dimensions of pursuing a doctoral thesis. In particular, I would like to thank Francis Schmitt for introducing me into the fascinating 3D domain and for sharing his vast knowledge with me ; I would like to thank Yücel Yemez for always indicating the good pointers and also for our discussions during the toughest moments.

I would like to express my gratitude to Bülent Sankur for discovering the researcher that I am, for all I've learned from him, for his trust in me... There is a very long list of things for which I have to express my gratitude. I hope he knows how much I am grateful.

I would like to thank Nozha Boujemaa for accepting the presidency of the jury. I would like to thank Ethem Alpaydın and Atilla Başkurt for being thesis reporters and jury members especially with many constraints concerning time and travel. Their appreciation for my work encourages me to continue in research. And also... Note the date of the thesis defense : it's 19th December 2007. That day, the metro wasn't working properly in Paris, and it was extremely hard to find a taxi. The jury members made their own way to be on time in the morning, even if they had to walk in Paris streets. I would like to thank all the jury members for this too.

I would like to thank Isabelle Bloch for her comments on the thesis presentation just a week before the defense. Her remarks had wonderful effects on the great day. I would like to thank Burak Acar for being a member of the thesis follow-up committee. I would like to thank Selim Eskiizmirliler for his support as a genuine Galatasaray brother.

I would like to thank Jérôme Darbon, exceptional researcher and "Roi de la Butte-aux-Cailles", Geoffroy Fouquier "Le Duc" and David Lesage "L'Aventurier" for their friendship and unconditional support. From Boğaziçi side, I would like to thank Helin Dutağacı and Erdem Yörük : these two people have always helped me without knowing and without expecting anything. In particular, I would like to thank Helin Dutağacı for this fantastic job of generating an expert classification for the Sculpteur database.

[1]This work has been done in two research labs ENST/TSI and BUSIM (Boğaziçi University) in the framework of the joint doctoral thesis programme between ENST and Boğaziçi University. A part of my studies in France has been funded by CROUS de Paris.

Again the joy of making a thesis in two institutions is to double the number of mates on this way that we call "to make a thesis". I would like to thank Tony Tung et Carlos Hernández-Esteban : I made a lot use of the tools they've developed during the Sculpteur project. Without discriminating between generations, I would like to thank all the people that I've known at TSI and at BUSIM. In particular, my sincere thanks to Jérémie, Julien, Eve, Alex, Gero, Antonio, Camille, Olivier, Ebru, İpek, Çağatay and Oya. I had very nice moments with them. I would like to thank also Sophie-Charlotte, Najib and Saïd with whom I shared many coffe breaks ; Sofiane, my last office mate and the postdocs Sylvie and Vincent.

I would like to thank all my friends for their appreciation and the courage they gave me during all the way towards the end. In particular, I am grateful to Fulden, Eren, Funda and Başak for being there on this great day and night ! Fulden and Eren had non-professsional remarks on the presentation, that I used a lot during the defense. Even they didn't know, in their eyes I found the courage that I needed.

I would like to thank Fulden, not only being present in the thesis defense, but especially being there since the beginning of the world and time.

Finally, a thousand thanks to my mother Emel, my father Neşat and my sister Ayşegül. To defend a thesis, I needed their constant support they gave me very generously and without fatigue. There is no word to express my gratitude to them. I hope this thesis shows that their supports and sacrifices weren't in vain.

Abstract

Next generation search engines will enable query formulations, other than text, relying on visual information encoded in terms of images and shapes. The 3D search technology, in particular, targets specialized application domains ranging from computer aided-design and manufacturing to cultural heritage archival and presentation. Content-based retrieval research aims at developing search engines that would allow users to perform a query by similarity of content.

This thesis deals with two fundamentals problems in content-based 3D object retrieval:

(1) How to describe a 3D shape to obtain a reliable representative for the subsequent task of similarity search?

(2) How to supervise the search process to learn inter-shape similarities for more effective and semantic retrieval?

Concerning the first problem, we develop a novel 3D shape description scheme based on probability density of multivariate local surface features. We constructively obtain local characterizations of 3D points on a 3D surface and then summarize the resulting local shape information into a global shape descriptor. For probability density estimation, we use the general purpose kernel density estimation methodology, coupled with a fast approximation algorithm: the fast Gauss transform. The conversion mechanism from local features to global description circumvents the correspondence problem between two shapes and proves to be robust and effective. Experiments that we have conducted on several 3D object databases show that density-based descriptors are very fast to compute and very effective for 3D similarity search.

Concerning the second problem, we propose a similarity learning scheme that incorporates a certain amount of supervision into the querying process to allow more effective and semantic retrieval. Our approach relies on combining multiple similarity scores by optimizing a convex regularized version of the empirical ranking risk criterion. This score fusion approach to similarity learning is applicable to a variety of search engine problems using arbitrary data modalities. In this work, we demonstrate its effectiveness in 3D object retrieval.

Contents

Introduction

Communication media have appeared in many guises during the never ending Information Age. The passage from textual exchange of ideas to audiovisual communication had been one of the major breakthroughs in the last century. Visual information in the form of image and video has now become so common that we cannot even imagine a world without photos, television and motion pictures. The advent of high-speed graphics hardware now offers a new dress to visual information: the digital 3D object.

3D objects arise in a number of disciplines ranging from computer aided-design and manufacturing (CAD/CAM) to cultural heritage archival and presentation. Other shades of the application spectrum include architecture, medicine, molecular biology, military, virtual reality and entertainment. Access to large 3D object databases occurring in these fields demands effective and efficient tools for indexing, categorization, classification and representation of the 3D data. Content-based retrieval addresses this challenging task using compact shape representations and intelligent search paradigms.

The next generation search engines will enable query formulations, other than text, relying on visual information encoded in terms of images and shapes. The 3D search technology, in particular, targets specialized application domains like the ones mentioned above [1]. In a typical 3D search scenario, the user picks a query from a 3D model catalogue and requests from the retrieval machine to return a set of “similar" database models in decreasing relevance. Content-based retrieval research aims at developing search engines that would allow users to perform a query by similarity of content. A request can be made for a number of objects, which are the most similar to a given query or to a manually entered query specification [2].

3D object retrieval hinges on shape matching, that is, determining the extent to which two shapes resemble each other [3]. The approaches to shape matching fall into two main categories: matching by feature correspondences and matching by global descriptors. The general strategy in the former approach is to compute multiple local shape features for every object and then, to assess the similarity of any pair of objects as the value of a distance function determined by the optimal set of feature correspondences at the optimal relative transformation [4]. The global descriptor-based paradigm (or feature vector-based in a different terminology [2]), on the other hand, reduces the shape characteristics to vectors or graph-like data structures, called *shape descriptors* [2, 3, 5], and then, evaluates the similarity degree between the descriptor pairs. We call this similarity degree as *the matching score* between the two shapes. In the retrieval mode, the matching scores between a query and each of the database models are sorted out. The retrieval machine then displays database models in descending scores. Effective retrieval means that the shapes displayed at the top of the list better match the query shape than the rest of the list.

In this thesis, we focus exclusively on the descriptor-based matching paradigm. A global shape descriptor is considered as a mapping from the space of 3D objects to some

finite-dimensional vector space. Accordingly, for each 3D object in the database, the retrieval system stores a vector of numerical attributes as a representative. This vector is expected to encode the information about the object's shape in such a way to allow fast and reliable similarity searches. The global descriptor-based paradigm is more suitable to machine learning tasks other than retrieval, such as object recognition and unsupervised classification. In fact, a 3D shape descriptor, which is effective in retrieval, is also expected to be effective in classification.

We address two research challenges concerning content-based 3D object retrieval:

- **Shape Descriptors**. We develop a novel 3D shape description scheme based on probability density of multivariate local surface features. We constructively obtain local characterizations of 3D points on a 3D surface and then summarize the resulting local shape information into a global shape descriptor. For probability density estimation, we use the general purpose kernel density estimation (KDE) methodology [6, 7], coupled with a fast approximation algorithm: the fast Gauss transform (FGT) [8, 9]. The conversion mechanism from local features to global description circumvents the correspondence problem between two shapes and proves to be robust and effective. Experiments that we have conducted on several 3D object databases show that density-based descriptors are very fast to compute and very effective for 3D similarity search.

- **Similarity**. We propose a similarity learning scheme that incorporates a certain amount of supervision into the querying process to allow more semantic and effective retrieval. Our approach relies on combining multiple similarity scores by optimizing a convex regularized version of the empirical ranking risk criterion [10, 11]. This score fusion approach to similarity learning is applicable to a variety of search engine problems using arbitrary data modalities. In this work, we demonstrate its effectiveness in 3D object retrieval.

In Chapter 1, we introduce the descriptor-based 3D object retrieval paradigm along with the associated research challenges. In Section 1.1, we fix ideas on issues such as descriptor requirements and similarity matching. In Section 1.2, we present the 3D object databases on which we have experimented during the thesis work. In Section 1.3, we review state-of-the-art 3D shape description techniques. In Section 1.4, we recapitulate the notion of computational similarity and introduce the problem of securing invariance at the matching stage. In Section 1.5, we define some of the performance assessment tools employed in information retrieval.

In Chapter 2, we develop the density-based shape description framework. Section 2.1 of this chapter is devoted to local characterization of a 3D surface and also addresses the associated computational issues such as feature calculation and feature domain sampling. In Section 2.2, after providing a brief overview of KDE in general terms, we explain how this powerful statistical tool can be used in the context of 3D shape description. Density-based shape description comes also with a set of dedicated tools, exploiting the probability density structure as we present in Section 2.3: *marginalization* and *probability density pruning*. Section 2.4 demonstrates that pdf-based descriptors are suitable for guaranteeing invariance to extrinsic effects, such as the object pose in the 3D space, at the matching stage. In this section, starting from the change of variables formula, we develop a similarity measure, which is invariant to coordinate axis relabelings and mirror reflections.

In Chapter 3, we describe our score fusion approach to the similarity learning problem in the context of content-based retrieval. In Sections 3.1 and 3.2, we lay down the problem of learning a scoring function by ranking risk minimization. In Section 3.3, we provide a support vector machines (SVM) based solution and present a score fusion algorithm. In Section 3.4, we design two specific applications, *bimodal* and *two-round* search protocols, where the algorithm can be employed.

In Chapter 4, we provide extensive experimental results. In Sections 4.1 and 4.2, we address the problem of setting the bandwidth parameter involved in KDE and illustrate the regularization behavior and the robustness properties of our descriptors. In Section 4.3, we deal with the target selection problem, coined as determining the pdf evaluation points in the feature domain. In Section 4.4, we assess the performance of standard similarity measures on density-based descriptors and demonstrate the superiority of the invariant scheme developed in Section 2.4 in experimental terms. In Section 4.5, we invoke dedicated descriptor manipulation tools of Section 2.3 to render our descriptors storage-wise efficient. In Sections 4.6 and 4.7, we experiment with two information fusion options, *feature-level* and *score-level* fusions, to bring the retrieval performance of the density-based framework at its best. In Section 4.6, we also illustrate the use of marginalization for non-heuristic feature space exploration to discover the most discriminative feature subsets. In Section 4.8, we contrast our descriptors to their counterparts in the literature and provide a performance landscape of the state-of-the-art shape descriptors. In Section 4.9, we analyze the performance of our framework across four semantically different 3D object databases of varying mesh quality. Finally, Section 4.10 is devoted to statistical learning-based score fusion and shows how more effective retrieval can be achieved using the algorithm developed in Chapter 3.

In Chapter 5, we conclude and discuss future research directions.

Chapter 1

3D Object Retrieval

In this chapter, we formulate the 3D object retrieval problem and describe how a typical retrieval system proceeds to access 3D objects by content. We focus exclusively on descriptor-based systems and the associated research challenges. In the paradigm of descriptor-based retrieval, for each database object, the system stores a representation containing a numerical summary of the object's shape. Such representations are called shape descriptors, which usually are vectors in some high-dimensional vector space. When a query is presented, the system calculates its descriptor(s) and compares it to those of the stored objects using a distance function, which measures dissimilarity. The system sorts the database objects in terms of increasing distance values. The items at the top of the ranked list are expected to resemble the query more than those at the end. The number of retrieved objects can be determined, either implicitly, by a range query in which case the system returns all objects within a user-defined distance, or explicitly, by fixing the number of objects to return.

In the following section, we provide an overview of the research challenges that we address in this thesis, in view of the comprehensive surveys [2, 3, 5]. In parallel, we state our approaches to deal with specific problems associated with these challenges. In Section 1.2, we present the 3D object databases on which we have experimented during the thesis work. In Section 1.3, we provide a general taxonomy on the state-of-the-art 3D shape descriptors. In Section 1.4, we briefly describe the notion of computational similarity and introduce the problem of securing invariance at the matching stage. Finally in Section 1.5, we conclude the chapter with the most commonly used performance measures adopted in information retrieval.

1.1 Research Challenges in 3D Object Retrieval

Descriptor-based 3D object retrieval is open to many research challenges. Along the same line as in [2, 3, 5], we classify them under the following headings:

- **Description Modalities for 3D Similarity Search.** So far, 3D descriptor research has concentrated generally on shape, as given by the object's surface or its interior, rather than other attributes like color and texture. This tendency is not only due to the fact that most of the similarity information is born within the shape but also because color and texture attributes are not always guaranteed to be present [2]. Our focus has also been on designing 3D descriptors based on shape, especially on surface shape information.

There is a multitude of formats to represent the 3D shape [12]. In CAD/CAM applications, a 3D object is usually represented by a collection of parameterized surface patches or using constructive solid geometry techniques. In medical imaging, scanning devices output voxel data or point clouds. Implicit surfaces, superquadrics, NURBS (non-uniform rational B-splines) and point-based surfaces constitute alternative forms of the state-of-the-art surface representations. The most popular format, on the other hand, is the polygonal -usually triangular- mesh, which arises in CAD/CAM and finite element analysis as well as in virtual reality, entertainment, and web applications. Furthermore, 3D scanning devices usually come with built-in triangulation software, thus favoring the use of this particular representation. Although a certain application usually demands a specific representation, which is more suitable for its tasks, one can always switch from one format to another. As will be explained in more detail in Section 1.2, we will work with 3D triangular mesh databases. Consequently, the descriptors designed in this thesis largely exploit this particular representation, yet preserve general applicability, especially to 3D point cloud data.

- **Descriptor Requirements.** There are two criteria that matter the most in designing a 3D shape descriptor: *effectiveness* and *efficiency*, [2, 3]. In general, a shape descriptor can be viewed as a mapping from the 3D object space to some high-dimensional vector space. The common objective in 3D descriptor research is to design such mappings in a way to preserve the maximum shape information with as low-dimensional a vector as possible. The informativeness requirement is called *effectiveness*, and the parsimony requirement as *efficiency*. On one hand, a shape descriptor is required to be effective in the sense of containing necessary discrimination information for retrieval and classification on a large 3D database. On the other hand, the descriptor should be sufficient and moderate in size to allow fast extraction and search for practical systems. These two criteria are in general competing, but also in some way complementary. To preserve all the shape information contained by the representation form provided, the description methodology should be exhaustive enough for reconstruction. As a result, the descriptor might be very high-dimensional. However, an approach producing a very high dimensional vector hampers the fast extraction and search requirements, reducing the efficiency of the overall system. Furthermore, the curse of dimensionality appearing in many disguises as far as learning in a large database is of concern, such an approach might lack generalization ability [13, 14], which is fundamental to classification. From this perspective, the objectives of effectiveness and efficiency may not be completely orthogonal to each other. For the time being, since there is no universal theory for designing such mappings, the effectiveness and efficiency of a descriptor (or the system using that particular descriptor) are evaluated on experimental terms. Regarding these issues, we would like to emphasize that, for retrieval and classification, not every shape detail is necessary and should even be discarded in the most principled way possible. Our thesis work has been guided by this token.

  Robustness constitutes another requirement: a descriptor should be insensitive to small shape variations [2, 3] and topological degeneracies. Accordingly, a descriptor should be more or less invariant to such defects and/or variations. A key point to consider is that *the similarity degree between two descriptors corresponding to two objects of the same semantics should always be greater than the similarity degree*

between two descriptors coming from different semantics. Along this line of argument, we postulate that a good shape descriptor should smooth out or eliminate individual shape details and enhance shared global properties. We believe that these global properties are induced by the semantic concept class that the 3D object belongs to. What would be nice, but is also difficult, is to have a certain amount of control over elimination of individual details and enhancement of global properties.

- **Invariance.** Along with effectiveness, efficiency and robustness, there exist other descriptor requirements relying on well-founded mathematical bases. According to a widely accepted definition, the shape of an object is the geometrical information that remains after the effects of translation, rotation, and isotropic rescaling have been removed [15]. Such effects are denominated collectively as similarity transformations. A shape descriptor or the associated matching scheme should be invariant against these effects. Invariance can be secured in two different ways:

  - *Invariance by description.* Either the descriptor is invariant by design, or the 3D object undergoes a preprocessing step where it is normalized to have a centered canonical reference frame and scale. It is hard to advocate for one or the other in terms of retrieval effectiveness. Several research groups favor their own choices, supporting their claims with experiments [16, 17]. Our opinion is that descriptors, which are invariant by design, come usually with a certain loss of shape information that might be valuable for a specific application. On the other hand, finding a canonical 3D reference frame on a per object basis is still an open problem. Principal component analysis (PCA) and its variants [18, 17] constitute a monopolistic tool for 3D pose normalization although they are not always very stable to variations of the object's shape even in a semantically well-defined class and might result in counter-intuitive alignments. Recently, Podolak et al. proposed a per-object alignment method based on finding symmetry axes [19]. Whenever such symmetries exist within the object, this approach may be promising and useful for obtaining semantically more meaningful reference frames. Nevertheless, the computational simplicity of PCA makes it still an attractive and widely used pose normalization tool. Regarding this issue, our standpoint is rather operational: we think that one should not refrain from the use of pose normalization schemes when the descriptor fails to be invariant by design.
  - *Invariance by matching.* Invariance can also be secured by minimizing a certain notion of distance between two descriptors, by holding one descriptor fixed and altering the other under the effect of the transformations that the 3D object might undergo. The invariance achieved in this way comes with no loss of shape information, but the matching becomes computationally more involved than merely taking, say, a Minkowski distance between descriptor vectors. In this approach, it is essential that the description algorithm is able to reflect the effect of the transformation directly to the descriptor, without recomputing it at every possible transformation of the object. In Section 1.4, we formalize this idea, and in Section 2.4, we show that the density-based shape description framework has this ability.

- **Similarity.** Whatever descriptor one obtains, there is always the ambiguity about the similarity criterion to be associated. Generally, it is not known in advance which

distance or norm would be the most suitable for retrieval and classification. The usual practice is to experiment with a set of distance functions and report their retrieval performances. The distance function yielding the best retrieval score is considered as the most suitable for the particular descriptor and database tested. On the other hand, for content-based retrieval applications, statistical learning theory provides a mathematical framework to learn the appropriate distance function, or similarity in general, under the heading of statistical ranking [11, 20, 21, 22]. In the present work, we tackle the similarity learning problem using a statistical learning-based score fusion scheme as described in Chapter 3.

Additional challenges concern the design of index structures associated with the similarity function used in the retrieval system, the notion of partial similarity and the availability of ground truth data [2]. The former two problems are not in the scope of the present work. The latter ground truth problem constitutes a side interest for the thesis, as we explain in the next section. We refer the reader to references [2, 3, 5] for more information on these issues.

1.2 3D Object Databases

A Google search for “3D Model" keyword returns 1,840,000 entries as of August 2007. Furthermore, research groups from all around the world have initiatives on developing and maintaining experimental search engines using 3D objects. It can be conjectured that the advent of fully operational 3D engines on the web is now a matter of time. In Appendix A, we provide a list of some private and publicly available 3D object databases used for research purposes. We note that there also exist many commercial repositories on the web from where 3D models can be purchased.

In the thesis, we have experimented with four different databases. All of them consist of 3D models given by triangular meshes, though they differ substantially in terms of content and mesh quality. These are:

- Princeton Shape Benchmark (PSB) [23],
- Sculpteur Database (SCU) [24, 25],
- SHREC’07 Watertight Database (SHREC-W) [26],
- Purdue Engineering Shape Benchmark (ESB) [27].

PSB is a publicly available database containing 1814 models, categorized into general classes such as animals, humans, plants, household objects, tools, vehicles, buildings, etc. [23] (see Figure 1.1). An important feature of the database is the availability of two equally sized sets. One of them is a training set (90 classes) reserved for tuning the parameters involved in the computation of a particular shape descriptor, and the other for testing purposes (92 classes), with the parameters adjusted using the training set.

SCU is a private database containing over 800 models corresponding to mostly archaeological objects residing in museums [24, 25]. So far, 513 of the models have been classified into 53 categories of comparable set sizes, including utensils of ancient times such as amphorae, vases, bottles, etc.; pavements; and artistic objects such as human statues (part and whole), figurines, and moulds. An example set of SCU objects is shown in Figure 1.2. The database is augmented by artificially generated 3D objects such as spheres, tori, cubes,

cones, etc., collected from the web. The meshes in SCU are highly detailed and reliable in terms of connectivity and orientation of triangles. The following figures illustrate the significant distinction between PSB and SCU in terms of mesh resolution. The average number of triangles in SCU and in PSB is 175250 and 7460 respectively leading to a ratio of 23. SCU meshes contain 87670 vertices on the average while for PSB this number is 4220. Furthermore, the average triangular area relative to the total mesh area is 33 times smaller in SCU than in PSB.

SHREC-W has been released for the Watertight track of the Shape Retrieval Contest (SHREC) in 2007 [28, 26]. It consists of 400 watertight meshes of high resolution, classified into 20 equally sized classes such as human, cup, glasses, octopus, ant, four-legged animal, etc. Classification semantics in SHREC-W are largely induced by topological equivalences as shown in Figure 1.3. Accordingly, SHREC-W constitutes a challenging test environment for geometry-based shape description methods.

ESB is another database that has been used in the SHREC'07 event and consists of 865 closed triangulated meshes, which represent engineering parts (Figure 1.4) [28, 29]. This dataset is classified into a ground truth classification with two levels of hierarchy. Overall there are three super-classes, namely, flat-thin object, rectangular-cubic prism, and solid of revolution, which are further categorized into 45 classes. It is particularly interesting to see the performance of 3D shape descriptors on such a database, as CAD offers an important application domain for content-based 3D shape retrieval.

The existence of widely accepted *ground truth* is a crucial aspect for objective and reproducible effectiveness evaluation [2]. Accordingly, obtaining ground truth data constitutes a fundamental problem, which should be addressed by researchers working in the field of content-based retrieval. The most rigorous attempt to determine 3D object semantics has been made by the Princeton group [23] and PSB has been a standard test environment since 2004. The approach to generate ground truth data for PSB has been to associate, to each semantically homogeneous group of objects, a class name representing an atomic concept (e.g., a noun in the dictionary) (see [23] for details). The process has been carried out by computer science students that were not expected to be experts in semantics. Sculpteur, on the other hand, is a more specialized database containing cultural heritage objects that ideally require expert intervention for ground truth classification. Manufacturing styles, periods, places and artists constitute basic entities of an *ontology* that would explicitly specify the domain of cultural heritage information. However, is it possible to derive such entities from the shape information contained in an object alone? The answer is not clear for the time being. Furthermore, the priorities of field experts, such as archaeologists, art historians, museum scientists, can be somewhat orthogonal to what is aimed at in machine learning. For instance, to an archaeologist, even a scratch on the surface of an ancient amphora may contain valuable information to determine its manufacturing period. On the other hand, a retrieval or classification algorithm would most probably consider this scratch as noise or small shape variation that should be discarded in the first place. Nevertheless, these conceptual difficulties have not prevented us from creating our own ground truth for SCU. Together with Helin Dutağacı[1], we have created a hierarchical classification for SCU objects. In determining the categories, we have been driven by form and functionality. The first two levels of hierarchy in our classification are given in the sequel.

[1]Helin Dutağacı is with the Boğaziçi University Signal and Image Processing Laboratory: http://busim.ee.boun.edu.tr/~helin/.

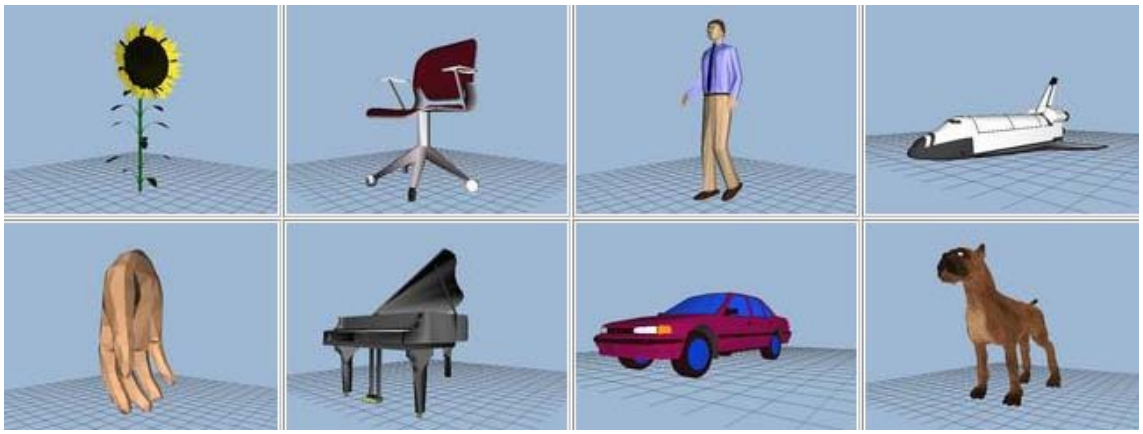

Figure 1.1: An example set of objects from the Princeton Shape Benchmark

Figure 1.2: An example set of objects from the Sculpteur Database

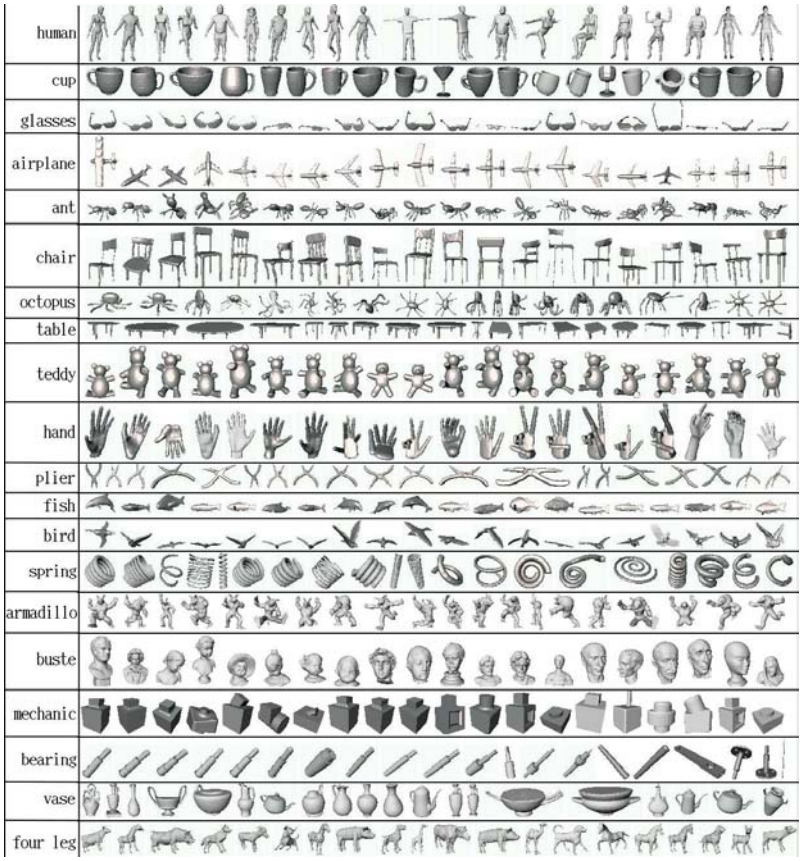

Figure 1.3: The SHREC-W Database and the associated classification

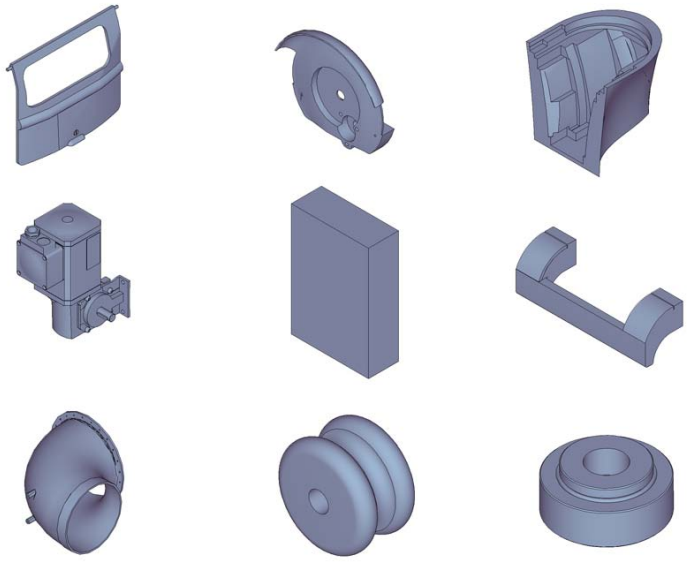

Figure 1.4: An example set of objects from the Purdue Engineering Shape Benchmark

- **Utensil**
  - Jar/Jag/Vase
  - Sugar caster
  - Amphora
  - Pilgrim bottle
  - Bowl
  - Carafe
  - Lamp
- **Computer-Generated Artificial Object**
  - Humanoid
  - Chess piece
  - Sphere
  - Torus
  - Eight
  - Cone
  - Cube
- **Pavement**
- **Artistic Object**
  - Statue
  - Mould
  - Relievo
  - Diverse

1.3 Background on 3D Shape Descriptors

Although the research on 3D shape descriptors for retrieval and classification has started just a decade ago or so, there is a considerable amount of work reported so far. The most up-to-date and complete reviews in this rapidly evolving field are given in [2, 5, 3]. In addition, the reference [23] is useful for a quick scan of practical schemes. Due to the variety and abundance of the methods, there is no universally accepted taxonomy of 3D shape descriptors. In our review, we preferred to provide a general classification, emphasizing the specific way to exploit the geometrical or topological shape information contained in the 3D object. More detailed accounts can be found in [2, 3, 5]. According to our generality criterion, we ended up with the categories shown in Table 1.1.

In the following, we describe the first three categories of our taxonomy, i.e., *histogram-based*, *transform-based*, and *graph-based* descriptors. Regarding *2D Image-based* methods, the work in [54] provides a comprehensive overview. For the remainder, we invite the reader to consult the references given in Table 1.1, or for a quicker scan, the surveys in [2, 3, 5, 54].

Table 1.1: A Taxonomy of 3D Shape Descriptors

Category	Examples
Histogram-Based	Cord and Angle Histograms [18, 30, 31]
	Shape Distributions [32]
	Generalized Shape Distributions [33]
	Shape Histograms [34]
	Extended Gaussian Images [35, 36]
	3D Hough Transform [37, 38]
	Shape Spectrum [37]
Transform-Based	Voxel-3D Fourier Transform (3DFT) [39]
	Distance Transform-3DFT and Radial Cosine Transform [40]
	Angular Radial Transform [41]
	PCA-Spherical Harmonics [42, 43, 44]
	Rotation Invariant Spherical Harmonics [45, 16]
	Spherical Wavelet Transform [46]
Graph-Based	Multiresolution Reeb Graphs [47, 48]
	Skeletal Graphs [49]
2D Image-Based	Silhouette Descriptor [17]
	Depth Buffer Descriptor [17]
	Lightfield Descriptor [50]
Other Methods	Spin Images [51]
	3D Zernike Moments [52]
	Reflective Symmetry Descriptor [53, 19]

1.3.1 Histogram-Based Methods

A vast majority of 3D shape descriptors can be classified under the heading of histogram-based methods. The term *histogram* is referred to as an accumulator that collects numerical values of certain attributes of the 3D object. In this respect, not all the methods presented in the sequel are true histograms in the rigorous statistical sense of the term, but they all share the philosophy of accumulating a geometric feature in bins defined over the feature domain.

In [18, 30], Paquet et al. have presented *cord and angle histograms* (*CAH*) for matching 3D objects. A cord, which is actually a ray segment, joins the barycenter of the mesh with a triangle center. The histograms of the length and the angles of these rays (with respect to a reference frame) are used as 3D shape descriptors. One shortcoming of all such approaches that simplify triangles to their centers is that they do not take into consideration the variability of the size and shape of the mesh triangles. First, because triangles of all sizes have equal weight in the final distribution; second, because the triangle orientations can be arbitrary, so that the centers may not represent adequately the impact of the triangle on the shape distribution. In [31], following similar ideas as in [18, 30], Paquet and Rioux have considered the angles between surface normals and the coordinate axes. In these approaches [18, 30, 31], the histograms are constructed always in univariate manner although it is also possible to consider multivariate histograms. Paquet and Rioux have argued that the bivariate histogram of the angles between the surface normal direction and the first two axes of the reference frame is sensitive to the level of detail at which the object

is represented. They have supported their claim by the example of two pyramids: one with the sides formed by inclined planes and the other with the sides formed by a stairway-like makeup. However, our experience shows that considering multivariate information proves to be more effective than merely concatenating univariate histograms for a retrieval application, as explained and supported by experiments in Sections 4.6.2 and 4.8.1.

In the *shape distributions* approach [32], Osada et al. have used a collection of shape functions, i.e., geometrical quantities computed by randomly sampling the 3D surface. Their shape functions list as the distance of a surface point to the origin, the distance between two surface points (*D2*), the area of the triangle defined by three surface points, the volume of the tetrahedron defined by four surface points and the angle formed by three random surface points. The descriptors become then the histograms of a set of these shape functions. The randomization of the surface sampling process improves the estimation over Paquet et al.'s [18] approach, since in this way, one can obtain a more representative and dense set of the surface points. The histogram accuracy can be controlled by changing the sample size. This quite appealing method suffers from the fact that the shape functions mentioned above are not specific enough to describe the 3D shape effectively. The poor retrieval performance of these approaches has been usually attributed to their global nature. The more recent *generalized shape distributions* (*GSD*) [33] partly overcome this difficulty by a "3D" histogram where two dimensions account for local and global shape signatures and one for distances between local shape pairs. However, the improvement provided by *GSD* is not sufficient to raise this methodology to the discrimination level of its competitors.

Ankerst et al. have used *shape histograms* for the purpose of molecular surface analysis [34]. A shape histogram is defined by partitioning the 3D space into concentric shells and sectors around the center of mass of a 3D model. The histogram is constructed by accumulating the surface points in the bins (in the form of shells, sectors, or both) based on a nearest-neighbor rule. Ankerst et al. illustrate the shortcomings of Euclidean distance to compare two shape histograms and make use of a Mahalanobis-like quadratic distance measure taking into account the distances between histogram bins. Since the approach proceeds with voxel data, 3D objects represented by polygonal meshes need to be voxelized prior to descriptor extraction.

Extended Gaussian image (*EGI*), introduced by Horn [35], and its variants [36, 55, 56] can be viewed as another class of histogram-based 3D shape descriptors. An *EGI* consists of a spherical histogram with bins indexed by (θ_j, ϕ_k), where each bin corresponds to some quantum of the spherical azimuth and elevation angles (θ, ϕ) in the range $0 \leq \theta < 2\pi$ and $0 \leq \phi < \pi$. The histogram bins accumulate the count of the spherical angles of the surface normal per triangle, usually weighted by triangle area. An important extension has been proposed by Kang and Ikeuchi who considered the normal distances of the triangles to the origin [36]. Accordingly, each histogram bin accumulates a complex number whose magnitude and phase are the area of the triangle and its signed distance to the origin respectively. The resulting 3D shape descriptor is called *complex extended Gaussian image* [36].

In [37, 38], Zaharia and Prêteux have introduced the *3D Hough transform* descriptor (*3DHT*) as a histogram constructed by the accumulation of points over planes in 3D space. A plane is uniquely defined by the triple (d, θ, ϕ), where d is its normal distance to the origin, and the pair (θ, ϕ) is the azimuth and elevation angles of its normal, respectively. A finite family of planes can be obtained by the uniform discretization of the parameters (d, θ, ϕ) over the domain $0 \leq d \leq d_{max}$,$0 \leq \theta < 2\pi$, and $0 \leq \phi < \pi$. This family of planes corresponds to a series of spherical histograms where each bin is indexed by

(d_i, θ_j, ϕ_k). To construct the Hough array, one creates planes at orientation (θ_j, ϕ_k) passing through the center $\mathbf{g}$ of a mesh triangle, and then calculates its quantized normal distance d_i to the origin. If the resulting value is positive, then the bin corresponding to the threesome (d_i, θ_j, ϕ_k) is augmented by a weight $w_{jk}^{\mathbf{g}}$. Zaharia and Prêteux have used as the corresponding weight factor, the area-weighted and thresholded absolute dot product between the normal of the triangle and the normal of the plane (θ_j, ϕ_k). *3DHT* can be considered as a generalized version of *EGI*. In fact, for a given d_i, (θ_j, ϕ_k)-bins correspond to an *EGI* at distance d_i, except for the way the contributions of the triangles are assessed. It can be conjectured that the *3DHT*-descriptor captures the shape information better than the *EGI*-descriptor.

In [37], Zaharia and Prêteux have presented the *shape spectrum* descriptor for 3D retrieval proposed within the MPEG-7 framework for multimedia content description. This descriptor consists of the distribution of the shape index feature, which is introduced by Koenderink and van Doorn [57]. The shape index is a function of the two principal curvatures. Its invariance with respect to rotation, translation and scale is appealing. Nevertheless, the unreliability of curvature estimation leads to a lack of robustness. Zaharia and Prêteux have tried to alleviate this shortcoming by augmenting the shape index histogram by two additional attributes named planar surface and singular surface. Although experiments conducted by the authors with this descriptor on several 3D databases have shown good retrieval results; in other instances reported in [2, 25], the shape spectrum descriptor has failed to provide adequate discrimination.

1.3.2 Transform-Based Methods

Two research groups have had considerable impact in 3D shape descriptors research to date: the Princeton group[2] and the Konstanz group[3]. Interestingly, transform methodologies from classical signal processing, such as *3D Fourier transform* and *spherical harmonics transform* to a larger extent, have been the main tool used in a great deal of descriptors developed by these two groups.

Vranić and Saupe from the Konstanz group have used 3D Fourier transform (3DFT) to map the rasterized version of a 3D triangular mesh from the voxel grid to the frequency domain [39]. As 3DFT is not rotation-invariant, the voxelization operation is performed after pose normalization via PCA. In fact, the voxel data can also be used as a 3D shape descriptor on its own [39]. By switching from the spatial domain to the frequency domain via 3DFT, one can obtain a descriptor with a reduced size by discarding high frequency components. This truncation has the additional benefit of filtering out individual shape details that are irrelevant for retrieval and classification. The idea of using 3DFT on a voxelized grid has also been pursued by Dutağacı et al. [40]. The authors have explored possible voxelization options as well as radial cosine transform as an alternative to 3DFT. Furthermore, by considering the sum of magnitudes of 3DFT coefficients at the same frequency shell, they have obtained a rotation-invariant descriptor, bypassing PCA step at the expense of lost shape information.

The spherical harmonics transform (SHT) [58] have become a very popular tool in the field of 3D shape descriptors. The Konstanz group uses SHT to transform spherical functions densely sampled over the surface of a PCA-normalized object [42, 43, 44]. Again, the array of spherical function values may serve as a descriptor on its own albeit with very

[2] http://www.cs.princeton.edu/gfx/proj/shape/
[3] http://infovis.uni-konstanz.de/research/projects/SimSearch3D/

high dimensionality. SHT is suitable to reduce the descriptor size considerably, yet without losing too much shape information. The so called *ray-based* or *extent* (*EXT*) descriptor gives the SH-transformed version of the maximal distance from the center of mass as a function of the spherical angle [42]. In [43], Vranić has improved this descriptor by considering a collection of extent functions evaluated at concentric spheres with different radii, again with SHT-mapping. This latter descriptor, called as *radialized extent* descriptor (*REXT*), proves to be highly discriminating on different databases [23, 2].

Funkhouser et al. [45] from the Princeton group have developed a rotation-invariant descriptor using SHT. The method requires the 3D object to be voxelized in a binary fashion. 3D voxel data can be interpreted as a collection of spherical functions $f_r(\theta, \phi)$, where r corresponds to the distance from the origin of the voxel grid and (θ, ϕ) to spherical coordinates. The binary function is sampled for a sufficient number of radii $r = 1, \ldots, R$ and angles (θ, ϕ). Each function at a specific radius is SH-transformed and the energies contained in low frequency-bands are stored in the final descriptor. In [16], Kazhdan et al. have provided mathematical support for rotation invariance of the descriptor. Basically, this mathematical justification relies on the fact that the energy in a certain frequency band of the ST does not change when the object is rotated around its center of mass. In the same work [16], they have also demonstrated that many existing descriptors can be rendered rotation-invariant by this approach. The rotation invariance of this class of descriptors should be understood with caution as it comes with a certain loss of shape information [16, 59]. In fact, the use of SHT for 3D shape description has been a matter of debate between the Princeton and Konstanz groups. The latter argues that PCA normalization should be applied prior to SHT and the magnitude of transform coefficients should be used as the descriptor, while the former claims that PCA is unstable and rotation invariance should be secured by considering the energies in different bands of the transform domain. Both groups support their claims with retrieval experiments in large databases and favor their individual standpoints. This discrepancy might be due to database differences and/or specific implementation details.

1.3.3 Graph-Based Methods

Graph-based approaches are fundamentally different from other vector-based descriptors. They are more elaborate and complex, in general harder to obtain; but they have the potential of encoding geometrical and topological shape properties in a more faithful and intuitive manner than vector-based descriptors. However, they do not generalize easily to all representation formats and they require dedicated dissimilarity measures and matching schemes. Due to their complexity in extraction and matching stages, they are not very efficient for general-purpose retrieval applications. We note that, using tools from spectral graph theory, some part of the information contained in a graph can be encoded in the form of numerical descriptions. Nevertheless, the lack of a vector representation by construction prevents the use of a great deal of learning algorithms for classification. Although the emphasis in the present work is on vector-based descriptors, we include two representative studies for the completeness of the account: *multiresolution Reeb graphs* [47, 48] and *skeletal graphs* [49]. The works in [60, 61, 62] and references therein include a comprehensive list of other graph-based representations and matching methods.

Hilaga et al. [47] have introduced the concept of *topology matching* for 3D object retrieval. The algorithm relies on constructing *Reeb graphs* at multiple levels of resolution of a function μ defined over the object's surface. The function μ can be the height of a point on

the surface, the curvature value or the integrated geodesic distance at that point. According to the function chosen, the resulting descriptor enjoys certain invariance properties. Each node in each such graph corresponds to a connected component of the object in the sense that μ-values in that component fall within the same interval determined by the resolution at which the graph is constructed. Parent-child relationships between nodes represent adjacent intervals of these μ-values for the contained object parts. Furthermore, a graph at a coarser level is encoded as the ancestor of a graph at a finer level. Obviously, at the coarsest level, the graph consists of a single node accounting for the whole object. The theoretical support of this multiresolution approach is that, for a given object, as the resolution gets finer, the nodes of the resulting graph corresponds to the singular points of the μ-function. It has been demonstrated that such critical point locations are valuable in studying the topology of the underlying object [63]. Sophisticated heuristics have been proposed to match two graphs for similarity assessment in [47] and in [25, 48]. Moreover, in the latter two works, the graph has been augmented with other vector-based descriptors to improve the discrimination ability.

In [49], the authors have described a method for searching and comparing 3D objects via *skeletal graph matching*. The objective is to build an interactive system that allows part matching. The visualization of the results is facilitated by skeletal graphs, which also help the user to refine and interactively change his/her query. The skeletal graph is obtained from object voxel data as a directed acyclic graph (DAG). Each node of the DAG is associated with a set of geometric features and a signature vector that encodes topological information. The latter is called a topological signature vector (TSV), which is derived from the eigendecomposition of the graph's adjacency matrix. The matching procedure consists of two stages where first a topology matching between the query database graphs is performed on a per node basis. The second (optional) stage consists of geometry matching. It can be used to refine the possible set of retrieved database objects. However, the authors have not elaborated on this issue further.

1.4 Similarity Measures

The assessment of similarity between two objects is usually performed by computing a dissimilarity measure between their corresponding descriptors. Accordingly, throughout the thesis, we will use the terms *similarity* and *dissimilarity* (or *distance*) interchangeably. The context will clarify the distinction. A similarity function $sim : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$ can be viewed as the abstract inverse to a dissimilarity function $dist : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$, where $\mathcal{F}$ is the space of generic descriptors f. Ideally, we expect that $dist(f, f')$ decreases ($sim(f, f')$ increases), as the level of semantic similarity between the corresponding objects gets higher. The dissimilarity measure $dist$ may enjoy the following properties [3]:

(1) *Identity*: $\forall f \in \mathcal{F}, dist(f, f') = 0$.

(2) *Positivity*: $\forall f \neq f' \in \mathcal{F}, dist(f, f') > 0$.

(3) *Symmetry*: $\forall f, f' \in \mathcal{F}, dist(f, f') = dist(f', f)$.

(4) *Triangle inequality*: $\forall f, f', f'' \in \mathcal{F}, dist(f, f'') \leq dist(f, f') + dist(f', f'')$.

A dissimilarity measure satisfying all of the four properties listed above is said to be a *metric*. A *pseudo-metric* satisfies all metric properties but *positivity*, and a *semi-metric* satisfies only the first three properties. A function satisfying the *triangle inequality* is often

desirable, since it can make retrieval more efficient [64]. These properties are abstract in the sense that the dissimilarity function can be defined on arbitrary descriptor spaces (e.g., for graphs, or even directly for 3D objects). In the present work, we restrict ourselves to the case of vector-based descriptors, that is, when the descriptor space $\mathcal{F}$ is a finite vector space. Classical dissimilarity measures for vector-based descriptors are L^p-distances ($p > 0$), Bhattacharyya distance, χ^2-divergence [65], symmetricized versions of Kullback-Leibler divergence [65], earth mover's distance [66] and histogram intersection distance [67] (see Appendix B for definitions).

Regarding similarity measures, we would like to underline two points:

- A distance function $\overline{dist}$ on a finite-dimensional vector space $\overline{\mathcal{F}}$ usually arises as a discretized version of a continuous functional $dist$ on an infinite space of continuous functions $\mathcal{F}$. Suppose that f is a function describing a certain object by mapping multidimensional attributes $t \in \mathbb{R}^m$ to reals. A functional $dist$, quantifying the amount of variation between f and another descriptor function f' can be generically written as

$$dist(f, f') = \int_{t \in \mathbb{R}^m} \eta\left(f(t), f'(t)\right) dt,$$

  where η is a point-wise dissimilarity function, e.g., for L^1, $\eta(\cdot, \cdot) = |\cdot - \cdot|$. In practice, an object is described by a finite vector of $f(t)$-values, i.e., $\mathbf{f} = [f(t_1), \ldots, f(t_N)]$, in which case the above integral should be discretized as

$$\overline{dist}(\mathbf{f}, \mathbf{f}') = \sum_n \eta\left(f(t_n), f'(t_n)\right) \Delta t_n, \tag{1.1}$$

  where Δt_n is the discretization step size. If the space of multidimensional attributes is uniformly partitioned, Δt_n becomes constant, hence affects the value of the integral by a constant amount for all descriptors, and consequently, it can be dropped. Otherwise, it must be taken into account in distance calculation.

- The dissimilarity between two objects can be made invariant against certain types of transformations Γ by the following formula:

$$dist_{\Gamma\text{-invariant}}(\mathbf{f}, \mathbf{f}') = \min_{\Gamma \in \mathcal{G}_\Gamma} dist(\mathbf{f}, \Gamma(\mathbf{f}')),$$

  where $\mathcal{G}_\Gamma$ is the group of transformations that the objects might have been undergone prior to descriptor extraction. Since we want that the distance function captures only intrinsic shape differences and commonalities, $\mathcal{G}_\Gamma$ should consist of extrinsic effects such as translation, rotation, isotropic rescaling or a combination of these. The above formula is practical only when the transformation can be directly effected on the descriptor, without recomputing it for every possible transformed version of the object. In Section 2.4, we concretize this idea of securing invariance at matching stage and develop such a measure, easily applicable for density-based shape descriptors.

1.5 Evaluation Tools for Retrieval

In this section, we summarize the most commonly used statistics for measuring the performance of a shape descriptor in a content-based retrieval application [23].

- **Precision-Recall curve.** For a query q that is a member of a certain class $\mathcal{C}$ of size $|\mathcal{C}|$, *Precision* (vertical axis) is the ratio of the relevant matches K_q (matches that are within the same class as the query) to the number of retrieved models K_{ret}, and *Recall* (horizontal axis) is the ratio of relevant matches K_q to the size of the query class $|\mathcal{C}|$:

$$\begin{aligned} Precision &= \frac{K_q}{K_{ret}}, \\ Recall &= \frac{K_q}{|\mathcal{C}|}. \end{aligned}$$

  Ideally, this curve should be a horizontal line at unit precision.

- **Nearest Neighbor (NN).** The percentage of the first-closest matches that belong to the query class. A high NN score indicates the potential of the algorithm in a classification application.

- **First-tier (FT) and Second-tier (ST).** First-tier is the recall when the number of retrieved models is the same as the size of the query class and second-tier is the recall when the number of retrieved models is two times the size of the query class.

- **E-measure**. This is a composite measure of the precision and recall for a fixed number of retrieved models, e.g., 32, based on the intuition that a user of a search engine is more interested in the first page of query results than in later pages. E-measure is given by

$$\mathrm{E} = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}.$$

- **Discounted Cumulative Gain (DCG)**. A statistic that weights correct results near the front of the list more than those appearing later, under the assumption that the user is interested more with the very first items displayed. To calculate this measure, the ranked list of retrieved objects is converted to a list L, where an element L_k has value 1 if the kth object is in the same class as the query and otherwise has value 0. Discounted cumulative gain at the kth rank is then defined as

$$\mathrm{DCG}_k = \begin{cases} L_k, & k = 1, \\ \mathrm{DCG}_{k-1} + \frac{L_k}{log_2(k)}, & \text{otherwise.} \end{cases}$$

  The final DCG score for a query $q \in \mathcal{C}$ is the ratio of $\mathrm{DCG}_{K_{max}}$ to the maximum possible DCG that would be achieved if the first $|\mathcal{C}|$ retrieved elements were in the class $\mathcal{C}$, where K_{max} is the total number of objects in the database. Thus DCG reads as

$$\mathrm{DCG} = \frac{\mathrm{DCG}_{K_{max}}}{1 + \sum_{k=2}^{C_q} \frac{1}{log_2(k)}}.$$

- **Normalized DCG (NDCG)**. This is a very useful statistic based on averaging DCG values of a set of algorithms on a particular database. NDCG gives the relative performance of an algorithm with respect to the others tested under similar circumstances. A negative value means that the performance is below the average; similarly a positive value indicates an above-the-average performance. Let $\mathrm{DCG}^{(A)}$

be the DCG of a certain algorithm A and $\mathrm{DCG}^{(avg)}$ be the average DCG values of a series of algorithms on the same database, then NDCG for the algorithm A is defined as

$$\mathrm{NDCG}^{(A)} = \frac{\mathrm{DCG}^{(A)}}{\mathrm{DCG}^{(avg)}} - 1.$$

All these quantities are normalized within the range $[0, 1]$ (except NDCG) and higher values reflect better performance. In order to give the overall performance of a shape descriptor on a database, the values of a statistic for each query are averaged over all available queries to yield a single average performance figure. The retrieval statistics presented in this work are obtained using the utility software included in PSB [23].

Chapter 2

Density-Based 3D Shape Description

Density-based shape description is a generative model, aiming to encode geometrical shape properties contained within a class of 3D objects. This generative model relies on the idea that, associated with each shape concept, there is an underlying random process, which induces a probability law on some local surface feature of choice. We assume that this probability law admits a probability density function (pdf), which, in turn, encodes intrinsic shape properties to the extent achieved by the chosen feature. Shared or individual aspects of two shape concepts can be quantified by measuring the variation between their associated feature pdfs. The surface feature can be *general*, such as the distance from a predefined origin, or *specific*, for instance, involving local differential structure on the surface. As one moves from *general* to *specific*, discrimination power of a local feature and its pdf increase. General features can be joined together in order to obtain more specific multivariate features. With its ability to process multivariate local feature information, the density-based framework generates a family of 3D shape descriptors on which we elaborate in this chapter.

A density-based descriptor of a 3D shape is defined as the sampled pdf of some surface feature, such as radial distance or direction. The feature is local to the surface patch and treated as a random variable. At each surface point, one has a realization (observation) of this random variable. For instance, if the surface is given in terms of a triangular mesh as it is generally assumed in this work, the set of observations can be obtained from vertices and/or triangles. To set the notation, let S be a random variable defined on the surface of a generic 3D object O and taking values within a subspace $\mathcal{R}_S$ of $\mathbb{R}^m$. Let $f_{S|O} \triangleq f_S(\cdot|O)$ be the pdf of S for the object O. This pdf can be estimated using the set of observations $\{s_k \in \mathcal{R}_S\}_{k=1}^K$ computed on the object's surface. In the sequel, random variables appear as uppercase letters while their specific instances as lowercase. Suppose furthermore that we have specified a finite set of points within $\mathcal{R}_S$, denoted as $\overline{\mathcal{R}}_S = \{t_n \in \mathcal{R}_S\}_{n=1}^N$, called *the target set*. The density-based descriptor $\mathbf{f}_{S|O}$ for the object O (with respect to the feature S) is then simply an N-dimensional vector whose entries consist of the pdf samples at the target set, that is, $\mathbf{f}_{S|O} = [f_S(t_1|O), \ldots, f_S(t_N|O)]$.

Density-based shape description consists of three main stages:

(1) First, in *the design stage*, we choose good local features that accumulate to global shape descriptors. Good features are computationally feasible and discriminative (Sections 2.1.1 and 2.1.2).

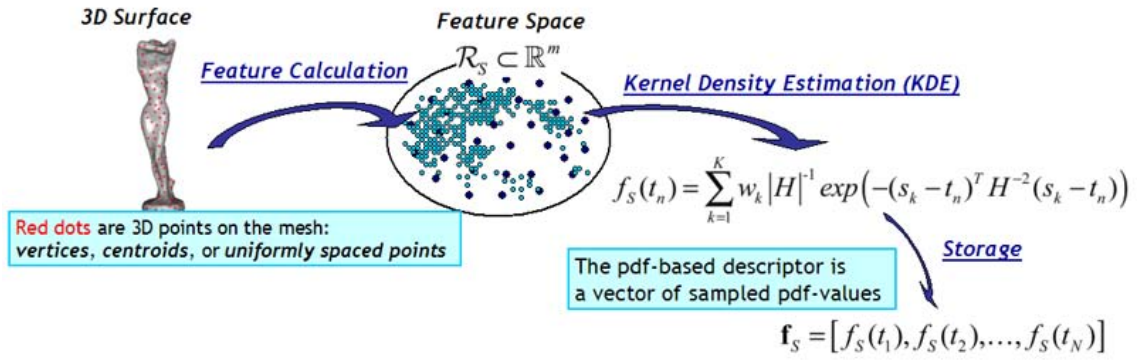

Figure 2.1: Density-based shape description

(2) Second, in *the target selection stage*, we focus on determining the pdf evaluation points sampled in $\mathcal{R}_S$, i.e., determining the target set $\overline{\mathcal{R}}_S$ (Section 2.1.3).

(3) Finally, we address *the computational stage*, in search of an efficient computational scheme to estimate $f_S(t|O)$ at designated targets $t \in \overline{\mathcal{R}}_S$. In the present work, we use the kernel density estimation (KDE) approach coupled with a fast algorithm, the fast Gauss Transform (FGT) [9] (Section 2.2).

The final output of these stages is the shape descriptor vector $\mathbf{f}_{S|O}$, whose components $f_S(t_n|O)$ are the pdf values evaluated at the target set $\overline{\mathcal{R}}_S$. Figure 2.1 illustrates this descriptor extraction process.

The fact that the description scheme is based on pdfs allows one to use this special structure for several ends. For instance, in Section 2.3, we present two descriptor manipulation tools: *marginalization* and *probability density pruning*. Marginalization integrates out the information contained in a subset of feature components from a multivariate pdf. This can help us to explore eventual redundancies of certain components in a multivariate local feature. Probability density pruning, on the other hand, eliminates negligible pdf values from the descriptor by thresholding the prior feature density f_S, which is calculated by averaging conditional pdfs $f_{S|O_u}$ over a representative set of objects $\mathcal{O} = \{O_u\}$. Both of these tools can be employed to reduce descriptor dimensionality without loss of performance. Another advantage that the pdf structure offers is to secure invariance against certain types of object transformations at the matching stage, without the need of recomputing the descriptor for every possible transformation. In Section 2.4, we develop such a similarity measure and show its invariance under specific conditions. In Section 2.5, we finalize the chapter by providing an implementation summary of the density-based shape description algorithm.

2.1 Local Characterization of a 3D Surface

2.1.1 Local Surface Features

In this section, we describe the local geometric features that we use to characterize 3D surfaces (see Figure 2.2). Our approach is inductive in the sense that we start by simple

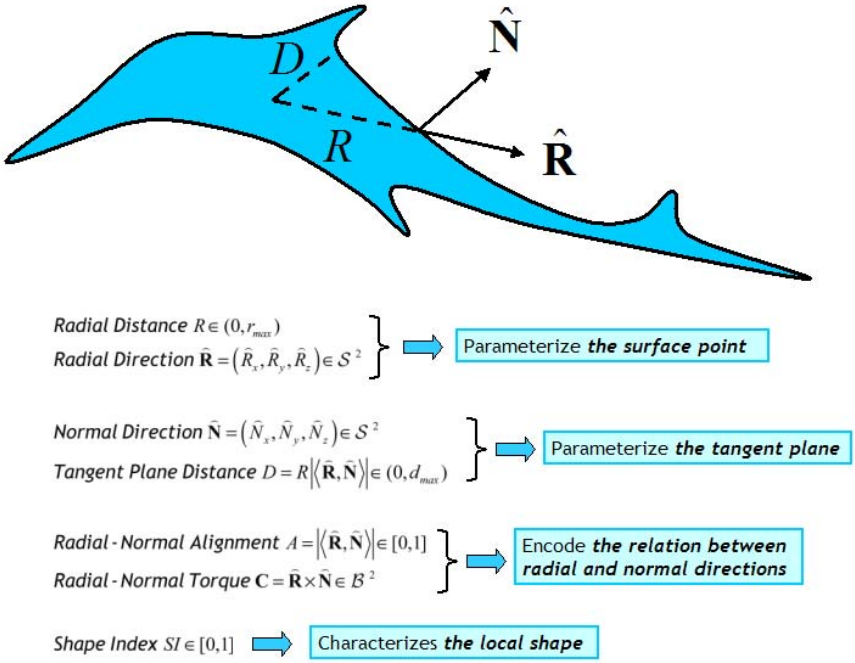

Figure 2.2: Illustration of local surface features

features with minimal requirements about the underlying surface and continue with more sophisticated ones in order to arrive to an extensive pointwise characterization.

Zero-Order Features

The most basic type of local information about a point lying on a 3D surface are its coordinates. Zero-order features require solely that the underlying surface be continuous without any further higher-order differential structure: a condition, which is usually fulfilled for 3D meshes.

- *Radial distance* R measures the distance of a surface point Q to the origin (centroid) and has taken place in many different shape descriptors [32, 30]. Although it is not an effective shape feature all by itself, when used jointly with other local surface features, it helps us to decouple the feature distribution at varying distances from the object's center of mass.

- *Radial direction* $\hat{\mathbf{R}}$ is a unit length vector $(\hat{R}_x, \hat{R}_y, \hat{R}_z)$ collinear with the ray traced from the origin to the surface point Q. This unit-norm vector is obviously scale-invariant. When we augment the $\hat{\mathbf{R}}$-vector with the radial distance R, the resulting 4-tuple $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ can serve as an alternative to the standard Cartesian coordinate representation of the surface point. However in this parameterization, distance and direction information are decoupled. We say that the feature R *radializes* the density of the feature $\hat{\mathbf{R}}$. Note also that the range of these features can be determined independently. In fact, the vector $\hat{\mathbf{R}}$ lies on the unit 2-sphere, and the scalar R lies on the interval $]0, r_{max}]$, where r_{max} depends on the size of the surface.

First-Order Features

First-order features require first-order differentiability, hence the existence of a tangent plane at each surface point. For 3D meshes, each interior point on a mesh triangle has obviously a tangent plane, which is basically the plane supporting the mesh triangle to which the point belongs. At vertices, even though the situation is more complex, one can compute a tangent plane by using the 1-ring of the vertex point [68].

- *Normal direction* $\hat{\mathbf{N}}$ is simply the unit normal vector at a surface point and represented as a 3-tuple $(\hat{N}_x, \hat{N}_y, \hat{N}_z)$. Similar to the radial direction $\hat{\mathbf{R}}$, the normal $\hat{\mathbf{N}}$ is scale-invariant.

- *Radial-normal alignment* A is the absolute cosine of the angle between the radial and normal directions and is computed as $A = |\langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \rangle| \in [0, 1]$. This feature measures crudely how the surface deviates locally from sphericity. For example, if the local surface approximates a spherical cap, then the radial and normal directions align, and the alignment A approaches unity.

- *Tangent plane distance* D stands for the absolute value of the distance between the tangent plane at a surface point and the origin. This scalar feature D is related to the radial distance R by $D = RA$. The joining of D with the normal direction $\hat{\mathbf{N}}$ provides a four-component vector $(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ that corresponds to the representation of the local tangent plane. As in the radial case, this representation also separates the distance and direction information associated with the tangent plane.

- In addition to the radial-normal alignment A, the interaction between the surface normal vector and the radial direction can be quantified by taking the cross product between $\hat{\mathbf{R}}$ and $\hat{\mathbf{N}}$. The *torque* feature $\mathbf{C} = \hat{\mathbf{R}} \times \hat{\mathbf{N}}$ can be considered as a local rotational force when $\hat{\mathbf{R}}$ is viewed as the position of a particle, which is under the influence of an external force $\hat{\mathbf{N}}$.

Second-Order Features

The second fundamental form II_Q contains useful differential geometrical information about a surface point Q. This form is defined as $II_Q(\mathbf{u}) = \langle d\mathbf{N}_Q(\mathbf{u}), \mathbf{u} \rangle$, where $\mathbf{u}$ is a 3-vector lying on the tangent plane and $d\mathbf{N}_Q$ is the differential of the normal field at the point Q [69]. The differential $d\mathbf{N}_Q$ is a linear map, measuring how the normals pull away within a neighborhood of Q, along a direction pointed by an arbitrary vector $\mathbf{u}$ on the tangent plane. Minimum and maximum eigenvalues of $d\mathbf{N}_Q$, called *principal curvatures* κ_1 and κ_2, are also the minimum and maximum of the second fundamental form II_Q. They measure the minimum and maximum rates of change at which the normal deviates from its original direction at Q. Given the principal curvatures, one can obtain a unique local characterization up to a scale [69]. By definition, II_Q requires second-order differentiability. This condition is never fulfilled for a 3D mesh, which is just piece-wise planar (hence at most first-order differentiable). Nevertheless, II_Q can be computed by fitting a twice-differentiable surface patch to the vertex point and invoking standard formulae from differential geometry [69], or by discrete approximation using the mesh triangles within the 1-ring of the vertex point [68, 70].

- *Shape index* SI, first proposed by Koenderink and van Doorn [57], provides a local categorization of the shape into primitive forms such as spherical cap and cup, dome,

rut, ridge, trough, or saddle (see Figure 2.3). In the present work, we consider the parameterization proposed in [71] given by

$$SI = \frac{1}{2} - \left(\frac{2}{\pi}\right) arctan\left(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}\right).$$

SI is confined within the range $[0, 1]$ and not defined when $\kappa_1 = \kappa_1 = 0$ (planar patch). Since the shape index SI is a function of the principal curvatures, it is considered as a second-order feature. It not only inherits the translation and rotation invariance of the principal curvatures, but also is a unitless quantity hence scale-invariant.

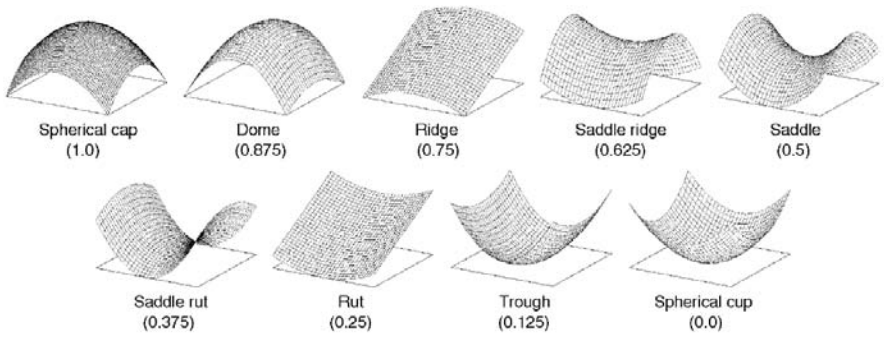

Figure 2.3: Shape index characterizes the local surface into a set of representatives.

Construction of a Multivariate Feature

Each of the above features reflects a certain incomplete aspect of the local shape. We can obtain a more thorough characterization of the surface point Q by constructing the multivariate feature $(R, \hat{\mathbf{R}}, \hat{\mathbf{N}}, SI)$ as explained below:

(1) The radial distance R restricts the point to the surface of a sphere around the object's center of mass.

(2) The radial direction $\hat{\mathbf{R}}$ spots the location of the point on the sphere.

(3) The normal direction $\hat{\mathbf{N}}$ associates a first-order differential structure, i.e., a tangent plane with the point.

(4) The shape index SI adds the more refined categorical surface information in terms of shape primitives.

The construction process is illustrated in Figure 2.4. In the density-based shape description framework, the pdf of these features, taken together in the form of a higher dimensional joint feature, would become a global descriptor, summarizing every single piece of local shape information up to second-order. However, this multivariate feature construction followed by pdf estimation is not without a caveat. Observe that the $(R, \hat{\mathbf{R}}, \hat{\mathbf{N}}, SI)$ is an 8-component feature with an intrinsic dimensionality of 6 as it takes values within $]0, r_{max}] \times \mathcal{S}^2 \times \mathcal{S}^2 \times [0, 1]$, where $\mathcal{S}^2$ denotes the unit 2-sphere. This fairly high dimensionality brings

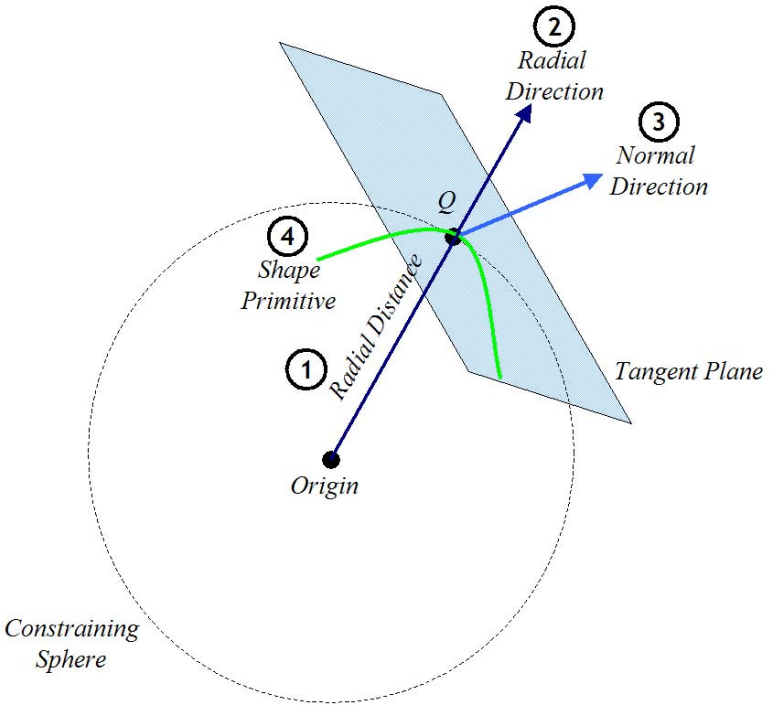

Figure 2.4: Construction of a multivariate feature

in concomitant problems of pdf estimation accuracy, high computation time and huge storage size as will be clarified in the upcoming sections.

Based on the above discussion, the features presented in this section can also be classified into two types: *primary* and *auxiliary*. The features involved in the full characterization up to second order as described above, namely, $R, \hat{\mathbf{R}}, \hat{\mathbf{N}}$ and SI are denominated as primary. We call the remaining tangent plane distance D, the radial-normal alignment A and the radial-normal torque $\mathbf{C}$ features as auxiliary, in the sense that they encode interactions between primary features. From a computational viewpoint, auxiliary features can be derived from primary ones. In Table 2.1, we summarize the properties of our local surface features.

Table 2.1: Classification and Invariance of Local Surface Features

	Classification		**Invariance**		
Feature	**Order**	**Type**	**Translation**	**Rotation**	**Scale**
Radial Distance	Zero	Primary	No	Yes	No
Radial Direction	Zero	Primary	No	No	Yes
Normal Direction	First	Primary	Yes	No	Yes
T-plane Distance	First	Auxiliary	No	Yes	No
Alignment	First	Auxiliary	No	Yes	Yes
Torque	First	Auxiliary	No	No	Yes
Shape Index	Second	Primary	Yes	Yes	Yes

From a practical viewpoint, by joining primary and auxiliary features, we can design the following multivariate pdf-based descriptors with manageable dimension in terms of pdf estimation accuracy, descriptor computation time and storage size:

- *Radial*-descriptor is the pdf of the Cartesian coordinate representation $(R, \hat{\mathbf{R}})$ of the surface point.
- *T-plane*-descriptor is the pdf of the $(D, \hat{\mathbf{N}})$-feature and aggregates the local tangent plane information.
- *Torque*-descriptor is the pdf of a radialized version of the torque feature, given by $(R, \mathbf{C})$.
- *Sec-Order*-descriptor is the pdf of the (R, A, SI)-feature, which radializes the auxiliary alignment information A together with the second-order feature SI.

In the experiments, we explore the discrimination ability of these *designed* features and their variants.

2.1.2 Feature Calculation

A triangular mesh $\mathcal{M}$ is given in terms of a union of K triangles $\mathcal{M} = \bigcup_{k=1}^{K} T_k$, where each triangle T_k is associated with a triplet of 3D points (A_k, B_k, C_k), represented by a triplet of 3-vectors $(\mathbf{a}_k, \mathbf{b}_k, \mathbf{c}_k)$. Regarding an m-dimensional local feature $S = (S_1, \ldots, S_m)$, we can obtain the observations by evaluating the value of S at the barycenter of each triangle or at each vertex. However, the mesh resolution might not be very fine and/or the triangles might have arbitrary shapes and sizes. Accordingly, the feature value at the barycenter or at the vertex may not be the most representative one. The shape of the triangle should be in some way taken into account in order to reflect the local feature characteristics more faithfully. The expected value of the local feature $E\{S|T\}$ over a generic triangle T is thought to be more reliable than the feature value sampled only at a single point, e.g, the barycenter or the vertex.

Consider T as an arbitrary triangle in 3D space with vertices A, B, and C represented by the vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ respectively (see Figure 2.5). By noting $\mathbf{e}_1 = \mathbf{b} - \mathbf{a}$ and $\mathbf{e}_2 = \mathbf{c} - \mathbf{a}$, we can obtain a parametric representation for an arbitrary point Q inside the triangle T as $\mathbf{p} = \mathbf{a} + x\mathbf{e}_1 + y\mathbf{e}_2$, where the two parameters x and y satisfy the constraints: $x, y \geq 0$ and $x + y \leq 1$. Assuming that the point Q is locally uniformly distributed inside the triangle T, the expected value of the ith component of S, denoted by $E\{S_i|T\}$, is given by

$$E\{S_i|T\} = \iint_{\Omega} S_i(x, y) f(x, y) dx dy, \quad i = 1, \ldots, m. \tag{2.1}$$

where $S_i(x, y)$ is the feature value at (x, y) and $f(x, y)$ is the uniform probability density function of the pair (x, y) over the domain $\Omega = \{(x, y) : x, y \geq 0, x + y \leq 1\}$. Accordingly, $f(x, y) = 2$ when $(x, y) \in \Omega$ or zero otherwise[1]. The integration is performed over the domain Ω. To approximate Eq. 2.1, we can apply Simpson's one-third numerical integration formula [72]. Avoiding the arbitrariness in vertex labeling by considering the three permutations of the labels A, B, and C, we get three approximations, which are in turn averaged to yield:

[1] Note that the area of the domain Ω is $1/2$, thus $f(x, y) = 2$ whenever $(x, y) \in \Omega$.

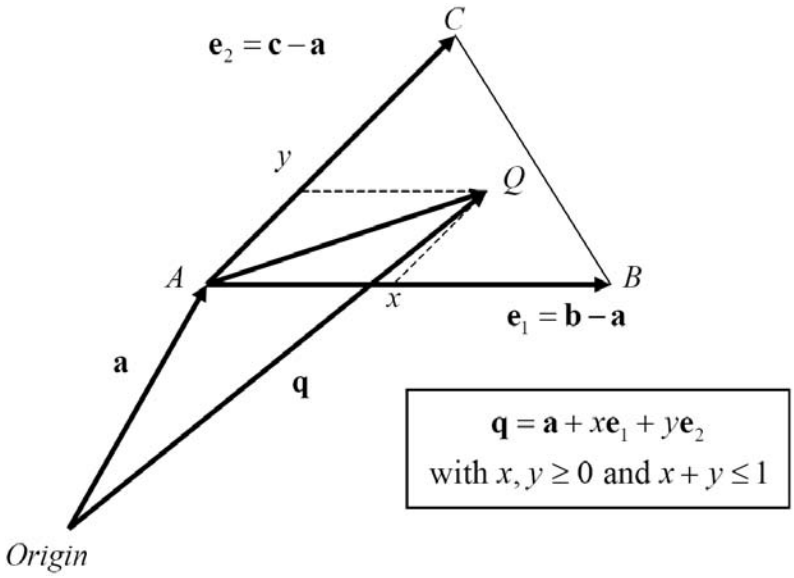

Figure 2.5: Parameterization of a point on a 3D triangle

$$
\begin{aligned}
E\{S_i|T\} \approx\ & (1/27)(S_i(\mathbf{a}) + S_i(\mathbf{b}) + S_i(\mathbf{c}))) \\
& +(4/27)(S_i((\mathbf{a}+\mathbf{b})/2) + S_i((\mathbf{a}+\mathbf{c})/2) + S_i((\mathbf{b}+\mathbf{c})/2)) \\
& +(4/27)(S_i((2\mathbf{a}+\mathbf{b}+\mathbf{c})/4) + S_i((\mathbf{a}+2\mathbf{b}+\mathbf{c})/4) + S_i((\mathbf{a}+\mathbf{b}+2\mathbf{c})/2)).
\end{aligned} \tag{2.2}
$$

Notice that Eq. 2.2 boils down to taking a weighted average of the feature values calculated at nine adequately chosen points on the triangle. For meshes with low resolution, this averaging has the effect of smoothing the observations so that the subsequent pdf estimation can be performed more accurately, thus resulting in more reliable descriptors. In Section 4.2.1, we support this claim by experimentation. As a side remark, we also point out the above averaging scheme can be applied to higher-order moments and cross-correlations of feature components.

It is also worth noting that the Simpson averaging scheme does not apply to the shape index SI. The computation of the latter feature involves curvature estimation, which can be carried on a per vertex basis as in [70]. In order to obtain realizations of SI per triangle, we take the average of the values at the three vertex points forming the triangle. In a similar way to Simpson averaging, this adds an implicit smoothing effect to the shape index calculation.

2.1.3 Target Selection

We define the target selection problem as sampling the range of the feature at which the pdf is evaluated. Since a density-based descriptor is merely a sampled version of a continuous pdf, we need to be efficient in choosing density evaluation points. First, we should not include very low density points and second, we should exploit any special structure associated with the feature range. The features presented in the previous section are either scalar or vector by construction. Sampling the real interval $\mathcal{I}$ of a scalar feature is relatively simple. For certain vector features, such as $\hat{\mathbf{R}}$ and $\hat{\mathbf{N}}$, on the other hand,

the unit-norm condition implies that the feature lies on the unit 2-sphere $\mathcal{S}^2$ and that the corresponding target points should satisfy this property. In the following, we discuss the sampling of $\mathcal{I}$ and $\mathcal{S}^2$ separately, providing two different methods for each.

Sampling for Scalar Features

The support $\mathcal{I} = (s_{min}, s_{max})$ of a scalar feature S is determined by the lower and upper percentiles $(\alpha_{min}, \alpha_{max})$ of the distribution of S, defined as

$$\begin{aligned} \alpha_{min} &\triangleq Pr\{S \leq s_{min}\} = \int_{-\infty}^{s_{min}} f_S(s)ds, \\ \alpha_{max} &\triangleq Pr\{S \geq s_{max}\} = \int_{s_{max}}^{\infty} f_S(s)ds, \end{aligned}$$

where $f_S(s)$ is the prior density of the feature S. We set both lower and upper percentiles to a unique value α. The prior density does not contain any object-specific information and is calculated by averaging over a set of representative objects $\mathcal{O} = \{O_u : u = 1, \ldots, |\mathcal{O}|\}$. Concretely, we have

$$f_S(s) = \sum_u f_S(s, O_u) = \sum_u f_S(s|O_u)Pr\{O_u\} = \frac{1}{|\mathcal{O}|}\sum_u f_S(s|O_u), \tag{2.3}$$

assuming that all objects are equiprobable, i.e., $Pr\{O_u\} = 1/|\mathcal{O}|$. Once the domain of the pdf is set, the $N_{\mathcal{I}}$ targets points remain to be determined. The two methods we consider are given below.

- In *uniform sampling*, we partition the interval into $N_{\mathcal{I}}$ equally spaced (uniform) sub-intervals and take the midpoints as targets.
- In *equal probability sampling*, we partition the interval into $N_{\mathcal{I}}$ equal probability regions and take the midpoints as targets. Similar to Max-Lloyd quantization [73], this scheme yields non-uniformly spaced sub-intervals and places more targets to regions of high prior density.

Note that clipping the tails of the distribution is meaningful only if the scalar feature has a magnitude interpretation (see Figure 2.6), according to which too small and/or too large values can be considered as outliers. In fact, for the alignment A and the shape index SI, which both lie on the unit-interval $[0, 1]$, the values near the boundaries are quite indicative of the local shape. For these, we simply take $N_{\mathcal{I}}$ equally spaced points within the unit interval.

Sampling for Unit-Norm Vector Features

For the radial direction $\hat{\mathbf{R}}$ and the normal direction $\hat{\mathbf{N}}$, we sample the unit 2-sphere $\mathcal{S}^2$ to obtain the targets. We propose again two different sampling methods:

- In *octahedron subdivision*, we consider an octahedron circumscribed by the unit sphere, subdivide into four each of its eight triangles, radially project the new triangles on the unit sphere, and iterate a factor of a times the subdivision process. The barycenters of the resulting triangles (after projecting back to the unit sphere) become the target set for direction components. This leads to an approximately

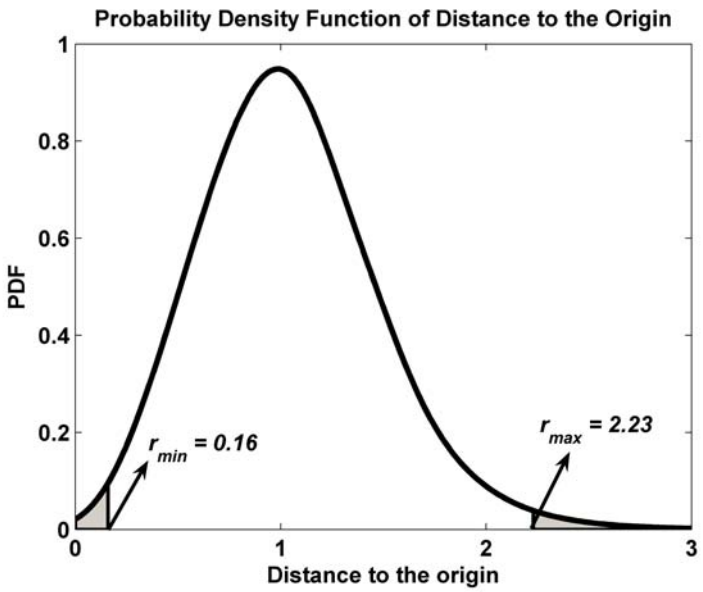

Figure 2.6: Probability density function of R displayed with the lower and upper 1% tails clipped

uniform partitioning of the sphere as shown in Figure 2.7. The recursion factor a determines the number of resulting points $N_{\mathcal{S}^2}$; e.g., for $a = 1$, we get $N_{\mathcal{S}^2} = 8 \times 4 = 32$ points; for $a = 2$, we get $N_{\mathcal{S}^2} = 8 \times 16 = 128$, and in general $N_{\mathcal{S}^2} = 2^{2a+3}$.

- In *spherical coordinate sampling*, we parameterize the unit sphere in terms of spherical coordinates (θ, ϕ). Recall that any point on the sphere can be expressed as a 3-tuple given by $(cos\theta sin\phi, sin\theta sin\phi, cos\phi)$ where $0 \leq \theta < 2\pi$ and $0 \leq \phi < \pi$. Uniformly sampling the θ- and ϕ-coordinates at N_θ and N_ϕ points, respectively, results in $N_{\mathcal{S}^2} = N_\theta \times N_\phi$ points. This method, however, does not provide a uniform partitioning of the sphere (see Figure 2.7).

Sampling for Arbitrary Vector Features

As pointed out earlier, we can design new features by joining the features presented in Section 2.1.1. In such cases, the target selection range occurs as the Cartesian product of the individual ranges of the features involved. $(R, \hat{\mathbf{R}})$ and $(D, \hat{\mathbf{N}})$ constitute two cases in point: their ranges both consist of the Cartesian product of an interval $\mathcal{I}$ on the real line with the unit 2-sphere $\mathcal{S}^2$. This decoupling allows us to sample $\mathcal{I}$ and $\mathcal{S}^2$ independently into $N_\mathcal{I}$ and $N_{\mathcal{S}^2}$ points respectively, and to take the Cartesian product of the two resulting sets to yield the target set $\overline{\mathcal{R}}_S$ of size $N = N_\mathcal{I} \times N_{\mathcal{S}^2}$. This idea extends naturally to the joining of L different features $S_l, l = 1, \ldots, L$ with individual target sets $\overline{\mathcal{R}}_{S_l}, l = 1, \ldots, L$, in which case the final target set is obtained as the L-fold Cartesian product of $\overline{\mathcal{R}}_{S_l}$'s.

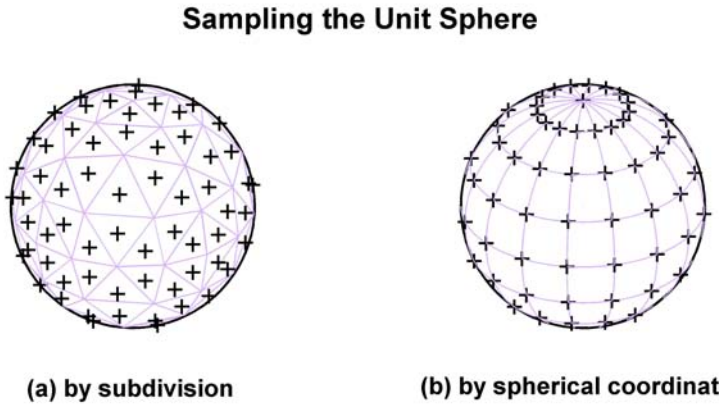

Figure 2.7: $N_{\mathcal{S}^2} = 128$ points on the unit-sphere obtained by the two methods mentioned in the text: (a) by subdividing an octahedron $a = 2$ times, (b) by uniformly sampling the spherical coordinates, $N_\theta = 16$ and $N_\phi = 8$

2.2 Kernel Density Estimation

Given a set of independent observations $\{s_k \in \mathbb{R}^m\}_{k=1}^K$ of a random variable (scalar or vector) $S \in \mathbb{R}^m$, the kernel approach to estimate the probability density value of S at a target $t \in \mathbb{R}^m$ is formulated in its most general form as

$$f_S(t) = \sum_{k=1}^{K} w_k \, |H_k|^{-1} \, \mathcal{K}\left(H_k^{-1}(t - s_k)\right). \tag{2.4}$$

where $\mathcal{K} : \mathbb{R}^m \to \mathbb{R}$ is a kernel function, H_k is an $m \times m$ matrix composed of a set of design parameters called *bandwidth* parameters (smoothing parameters or scale parameters) for the kth observation, and w_k is the positive scalar weight associated with the kth observation. The weights satisfy the constraint $\sum_k w_k = 1$ and, if all the observations are considered as equally likely, we have $w_k = 1/K, \forall k$. The contribution of each data point s_k to the density function f_S at a target point t is computed through the kernel function $\mathcal{K}$ rescaled by the matrix H_k and weighted by w_k. Thus KDE involves a data set $\{s_k\}_{k=1}^K$ with the associated set of weights $\{w_k\}_{k=1}^K$, the choice of a kernel function $\mathcal{K}$ and the setting of bandwidth parameters $\{H_k\}_{k=1}^K$.

Historically, the kernel estimate for a univariate pdf $f(s), s \in \mathbb{R}$ has been first introduced by Rosenblatt [74] as a numerical approximation to the derivative of the distribution function $F(s) = \int_{-\infty}^{s} f(u)du$. The method is also known as *Parzen window* after Parzen who has generalized the approach and proved consistency and asymptotic normality of the univariate kernel density estimator [75]. The multivariate generalization is due to Cacoullos [76]. Scott [77, 78, 7] and Silverman [79, 80, 81] have greatly contributed to the domain by dealing with issues such as the choice of the kernel function and the bandwidth parameters. Since then, the KDE approach has been a powerful non-parametric data analysis methodology finding several application niches in pattern recognition and computer vision

[13, 82, 83].

KDE can be viewed as a data smoothing technique overcoming certain limitations of the classical histogram method [7, 6]. In the histogram, one places a set of equal sized bins on the input domain and basically counts the number of observations falling inside each of them. This approach might become overly restrictive due to two reasons: (i) depending on the choice of origin, the appearance of the histogram differs significantly, (ii) all of the samples falling inside a bin contribute to pdf value equally, no matter how far they are from the bin center, leading to non-smooth estimates especially in high dimensions and when the number of observations is small. The KDE addresses both of these limitations [6]. First, by construction, there is no need to fix an origin in the input domain. Second and more importantly, the kernel $\mathcal{K}$ acts as a convolution filter weighting the observations s_k as a function of their distance to the density evaluation point t. In a nutshell, closer observations contribute more to the pdf estimate and far-away ones less. The smoothing behavior of the kernel estimator is controlled by the bandwidth parameter(s).

In Section 2.2.1, we put KDE in context and illustrate its discriminativeness for shape description. Note that in Appendix C.1, we derive the kernel estimator of a multivariate pdf following [13], paving the way for our discussion in Section 2.2.2 on the crucial issue of bandwidth selection. In Section 2.2.3, we discuss fast computational schemes to make efficient use of KDE for the 3D shape description problem.

2.2.1 KDE in Context

As pointed out in the previous paragraphs, KDE is a general purpose data smoothing technique. In the present section, we place this powerful tool in the context of 3D shape description and illustrate its discriminativeness using a constructed example. We need first to instantiate Eq. 2.4 with a particular choice of the kernel function. In the present work, we prefer the Gaussian kernel, in which case Eq. 2.4 becomes

$$f_S(t|O) = (2\pi)^{-m/2} \sum_{k=1}^{K} w_k |H_k|^{-1} exp\Big(-\frac{1}{2}(t-s_k)^T H_k^{-2}(t-s_k)\Big). \tag{2.5}$$

The reason for our choice on the Gaussian kernel is mainly computational. Direct evaluation of Eq. 2.5 on a target set $\{t_n\}_{n=1}^N$ using a set of observations $\{s_k\}_{k=1}^K$ has $O(KN)$-complexity, which might become infeasible or impractical for real-time tasks. The fast Gauss transform algorithm [8, 9] alleviates the computational burden by reducing the complexity down to $O(K+N)$ (see Section 2.2.3).

To put Eq. 2.5 in context, we make the following points for 3D shape description:

- *Observations* or *sources* $\{s_k\}_{k=1}^K$ are the feature values (or vectors) computed on the surface of an object O. They can be obtained from each of the mesh triangles, vertex points or by the averaging scheme described in Section 2.1.2.

- *Targets* $\{t_n\}_{n=1}^N$ are the pdf evaluation points so that $f_S(t_n|O)$-values constitute the descriptor vector $\mathbf{f}_{S|O} = [f_S(t_1|O), \ldots, f_S(t_N|O)]$.

- *Weights* $\{w_k\}_{k=1}^K$ stand for the strength or the intensity of each of the sources. For a 3D mesh, a weight w_k can be set to the relative area of the mesh triangle (with respect to the total surface area) over which the feature observation s_k is computed.

- *Bandwidth parameters* $\{H_k\}_{k=1}^{K}$ model the degree of uncertainty about the observations and control the smoothing bevahiour of the KDE. Appropriate bandwidth selection is a critical issue for all applications using the KDE scheme [84]. Multiple hypotheses can be made prior to the choice of the bandwidth parameters. Eq. 2.5 embodies the most general situation in which case the degree of uncertainty about each observation varies. It is also possible to impose a unique degree of uncertainty by assigning a fixed bandwidth matrix H to each of the observations in which case we have $H_k = H, \forall k$.

Discriminativeness of the KDE-based framework

We can analyze the discriminativeness of the KDE-based framework in two closely related but different contexts:

- In *recognition*, we are given a shape O' and wish to decide whether it is just *a perturbed version of* or is *completely different from* a certain shape O. In the former case, a good descriptor should not vary too much from shape O to O', in an appropriate sense of variation between descriptors. In the latter case, the descriptor must react more to differences. Put in other words, the amount of perturbation is one of the determining factors for the shape O' to be recognized as O or not.

- In the more complicated cases of *classification* and *retrieval*, we are faced with the problem of finding an equivalence relation on a given set of shapes. Depending on the application, we should assign *geometrically*, *topologically* and/or *semantically similar* shapes into the same equivalence class. All the shapes O' belonging to a certain equivalence class can be considered as random realizations of an ideal shape O, which may or may not exist physically but should fulfill the idea of the shape class under question. This platonic view of the affairs subsumes there exists a non-physical "pencil" and all the real physical "pencils" are just random realizations of it. In this setting, the variation among the descriptors of the shapes from the same equivalence class should always be uniformly lower than the variation among the descriptors of the shapes from different equivalence classes.

We will make use of the following simple example to illustrate the discriminativeness of the KDE-based scheme. Let $\mathcal{S} = \{s_k \in \mathbb{R}\}_{k=1}^{K}$ be a set of measurements about a scalar feature S and $\mathcal{S}' = \{s'_k \in \mathbb{R} : s'_k = s_k + \epsilon_k\}_{k=1}^{K}$ be a perturbed version of $\mathcal{S}$. We assume that the set $\mathcal{S}$ is non-random and constitutes an ideal set obtained from the surface of a generic shape O. The set $\mathcal{S}'$, on the other hand, is non-ideal and random. Randomness is incorporated through zero-mean independent identically distributed (i.i.d.) random variables ϵ_k's with variance σ^2. This additive perturbation model accounts for random geometrical variations of unconstrained amount, e.g., for the radial distance feature R. A small perturbation indicates small shape differences, while a larger one corresponds to more fundamental departures from the original shape. Regarding the set $\mathcal{S}'$, depending on the type of the context chosen for analysis, we can state the following:

- In *recognition*, $\mathcal{S}'$ arises from a "noisy" shape O'. The amount of noise, for which σ^2 is one possible measure, constitutes a determining factor for the shape O' to be O or not.

- In *retrieval* or *classification*, $\mathcal{S}'$ arises from a shape O', which is *similar* to the original shape O. Here, the amount of noise can be viewed as a decreasing function of similarity.

The shape information contained within the set $\mathcal{S}$ is described by the function $f_{S|O}$:

$$f_{S|O} \triangleq f_S(t|O) = \frac{1}{\sqrt{2\pi}Kh}\sum_{k=1}^{K} exp\left(-\frac{1}{2}\left(\frac{t-s_k}{h}\right)^2\right).$$

The descriptor function $f_{S|O'} \triangleq f_S(\cdot|O')$ for the shape O' can be defined likewise. Note that, while $f_{S|O}$ is non-random as it is a deterministic function of the deterministic set $\mathcal{S}$, $f_{S|O'}$ is random due to perturbation variables ϵ_k's. Here, to simplify the analysis, we have assumed that the number of observations K is the same for both sets, all the observations are equally weighted by $1/K$ and the scalar bandwidth parameter is fixed, i.e., $h_k = h, \forall k$. To quantify the amount of variation between $f_{S|O}$ and $f_{S|O'}$, we use the mean integrated absolute error ($MIAE$) defined as

$$\begin{aligned} MIAE(f_{S|O}, f_{S|O'}) &\triangleq E\left\{\int_{t\in\mathbb{R}} \left|f_S(t|O) - f_S(t|O')\right| dt\right\}, \\ &= \int_{t\in\mathbb{R}} E\left|f_S(t|O) - f_S(t|O')\right| dt, \end{aligned}$$

where $E\{\cdot\}$ is the expectation operator. It can be shown that, under the above assumptions and making use of the Hermite identity $exp(2ab - b^2) = \sum_{n=0}^{\infty} \frac{1}{n!} b^n H_n(a)$ (see Appendix C.2), an upper bound for $MIAE$ is given by

$$MIAE(f_{S|O}, f_{S|O'}) \leq C\sum_{n=1}^{\infty} \frac{1}{\sqrt{n!}} \frac{E|\epsilon|^n}{h^n}, \tag{2.6}$$

where $E|\epsilon|^n = E|\epsilon_k|^n, \forall k, n$ by i.i.d. assumption and C is a numerical constant. We can make the following comments based on Eq. 2.6:

- $MIAE$ increases as $E|\epsilon|^n$-terms increase. Recall that $E|\epsilon|^n$ quantifies the amount of geometrical variations exhibited by the random set $\mathcal{S}'$ with respect to the ideal deterministic set $\mathcal{S}$.

- $MIAE$ decreases as h^n-terms increase. This is a manifestation of the smoothing behavior of the KDE in our context. Eq. 2.5 tells that a large h flattens the descriptors so that variation among them gets small. The bound in Eq. 2.6 corroborates this fact.

- One can make $MIAE$ arbitrarily small or large by choosing the design parameter h accordingly. For instance, consider the hypothesis that "O' is O" (recognition) or "O' is in the same equivalence as O" (classification). If we have to retain the hypothesis, we should choose a large h so that the distance between descriptors becomes smaller than the rejection level. Or at the other extreme if the hypothesis is to be rejected, we should choose a small h so that the descriptor variation is large enough to reject the hypothesis.

The third point above is an important one as it elucidates the effect of the bandwidth parameter on the discriminativeness of the KDE-based scheme *a posteriori*, that is, based on whether the hypothesis is to be accepted or rejected. However, this *a posteriori* knowledge is the very purpose of recognition and classification: it cannot be used in advance. Nevertheless, the above analysis provides us with conceptual guidelines about the discriminativeness of the KDE-based scheme. To recapitulate, the bandwidth is a parameter of choice affecting the discriminativeness of the descriptors:

- Large bandwidths lead to smooth feature pdfs eliminating some shape details and favor the acceptance hypothesis by reducing descriptor variation.
- Small bandwidths preserve shape details more faithfully, lead to more discriminant descriptors and favor the rejection hypothesis.

How large the bandwidth parameter should be set is a matter of compromise between descriptor smoothness vs. discriminativeness, which is a hard question to answer *a priori*. One can view the bandwidth as a complex function of the feature dimension, the number of available observations, and in our context, how finely the shape database is categorized, making the analytical approaches to bandwidth selection very difficult. In the following section, we present several bandwidth parameter models and put the selection problem in context.

2.2.2 Bandwidth Selection

Looking at Eq. 2.5, we can see that the KDE scheme places a neighborhood in the form of a confidence ellipsoid of varying size and shape around each source s_k. As a target t is nearer to the center of this ellipsoid, the source s_k contributes more to the pdf value at t. The size and shape of the confidence ellipsoid is determined by the kernel function parameterized by the bandwidth H_k. As stated previously, the structure imposed on the bandwidth matrix models the uncertainty about the observations (both in kind and in degree) as well as the smoothness of the resulting descriptors. In the sequel, we first present the classical account on bandwidth parameter selection [84], and then proceed to discuss the selection problem in the context of density-based 3D shape descriptors.

Bandwidth Models and Standard Selectors

The KDE equation in Eq. 2.4 and its version with the Gaussian kernel in Eq. 2.5 embody the most general situation for the following reasons:

- The fact that H_k is a full matrix (i.e., every element can be non-zero) allows taking into account variances (diagonal entries) along each axis and covariances (off-diagonal entries) between different axes of the feature distribution. If the variance along an axis is large, the bandwidth along this axis should be large. A similar remark can be made for a strong pairwise correlation, which would lead to a higher entry at the corresponding off-diagonal. Non-zero off-diagonals lead to confidence ellipsoids, of general form, that need not to be aligned with feature axes.
- The fact that H_k varies for every source s_k takes into account variations regarding the sampling density. If the sampling density is low around a neighborhood of s_k (of a certain orientation), the corresponding bandwidth matrix H_k should admit higher

values as its entries in order to smooth the estimate at t. The shape and the size of the confidence ellipsoid vary for each observation.

In practice, H_k's are not known in advance and the accuracy of the KDE relies on how to choose the bandwidth matrix. We can make different assumptions about its structure:

(A1) $H_k = H = hI, \forall k = 1, \ldots, K$ (I is the $m \times m$ identity matrix). The bandwidth matrix *does not depend* on the source location, it is diagonal and controlled by a single parameter h. In this case, Eq. 2.4 gets the simplest form:

$$f_S(t) = \frac{1}{h^m} \sum_{k=1}^{K} w_k \mathcal{K}\left(\frac{t - s_k}{h}\right).$$

(A2) $H_k = H = diag(h_1, \ldots, h_m), \forall k = 1, \ldots, K$. The bandwidth matrix *does not depend* on the source location and has a diagonal form, determined by m parameters.

(A3) $H_k = H, \forall k = 1, \ldots, K$. The bandwidth matrix *does not depend* on the source location and is a full matrix determined by $m(m-1)/2$ parameters, since H is $m \times m$ symmetric.

(A4) $H_k = h_k I, k = 1, \ldots, K$. The bandwidth matrix *depends* on the source location, it is diagonal and proportional to a single parameter h_k. Since there are K sources, we are faced with the problem of choosing K parameters.

(A5) $H_k = diag(h_{k1}, \ldots, h_{km}), k = 1, \ldots, K$. The bandwidth matrix *depends* on the source location and is a diagonal matrix, determined by m parameters for each source s_k. Since there are K sources, we have to choose Km parameters.

(A6) $H_k, k = 1, \ldots, K$. The bandwidth matrix *depends* on the source location and is a full matrix, determined by $m(m-1)/2$ parameters. Since there are K sources, we have to choose $Km(m-1)/2$ parameters.

Each assumption above requires a data-driven bandwidth estimation procedure with varying complexity. Assumptions (A1) to (A3) lead to *fixed* bandwidth models and differ from the *variable* ones (A4) to (A6) in the sense that the bandwidth matrix is unique for a given set of observations, i.e., it does not depend on the source location. Variable bandwidth models have recently gained interest [83, 84]. They are more flexible since they make no assumptions about the shapes of the confidence ellipsoids (centered at observations). However, they are computationally very expensive to obtain and then to use; first because one should estimate as many bandwidth matrices as the number of observations, and second because, even if they are available in one way or another, there is no alternative, as efficient as the fast Gauss Transform (cf. Section 2.2.3), to the direct method for evaluating the pdf estimate at multiple target points. Computational requirements oblige us to focus on fixed bandwidth models.

Fixed bandwidth selectors split into three classes [6, 7]):

- Rule-of-thumb selectors,
- Plug-in selectors,
- Cross-validation selectors.

Rule-of-thumb and *plug-in* selectors both rely on minimizing the asymptotic mean integrated squared error (*AMISE*) [6, 7]) (cf. Appendix C.3). Under (A3) and when the observations are equally weighted, i.e., $w_k = 1/K, \forall k$, *AMISE* is given by

$$AMISE(H) = \frac{1}{4}C_1 \int \left(tr\left(H^T \mathcal{H}_f H\right)\right)^2 ds + C_2 \left(K|H|\right)^{-1}, \tag{2.7}$$

where C_1 and C_2 are kernel dependent constants, $\mathcal{H}_f$ is the Hessian matrix of second partial derivatives of the unknown pdf f_S. In theory, it is possible to minimize Eq. 2.7 with respect to H to obtain a bandwidth matrix, which is optimal in the *AMISE* sense. However, the expression one obtains depends on the unknown pdf. Rule-of-thumb selectors assume a reference density of known analytic form (such as the multivariate Gaussian), then calculate an analytical expression for the Hessian to obtain the optimal bandwidth for the reference density. In [7], it is demonstrated that in the case of multivariate Gaussian kernel and multivariate Gaussian reference pdf, under (A2), *AMISE*-optimal bandwidths are given by

$$h_j = \left(\frac{4}{(m+2)K}\right)^{1/(m+4)} \sigma_j, j = 1, \ldots, m, \tag{2.8}$$

where σ_j is the standard deviation of the reference density along the jth axis. In practice, one replaces σ_j with the sample standard deviation of the feature observations. Although not rigorously demonstrated elsewhere, an analogous expression for (A3) would be (as suggested in [6]):

$$H = \left(\frac{4}{(m+2)K}\right)^{1/(m+4)} \Sigma^{1/2}, \tag{2.9}$$

where Σ is the covariance matrix of the multivariate Gaussian reference. The expressions in Eqs. 2.8 and 2.9 are called Scott bandwidths. Rule-of-thumb selectors provide us with explicit, easily applicable formulae, which prove to be useful in practical real-time applications, where we cannot afford the computational cost of data-driven parameter estimation. Furthermore, when the true unknown pdf is not that much far from the multivariate Gaussian, say for instance, if it is unimodal, fairly symmetric and does not have tails that are too fat, Scott bandwidths are also statistically plausible [6]. A more complicated option is to employ a pilot estimate for the unknown pdf (or properly to say, for its Hessian) and to plug this in the minimizer of Eq. 2.7 to obtain an estimate more adapted to data than in Eqs. 2.8 and 2.9 [6]. Such approaches are called plug-in selectors, which can be viewed as data-driven generalizations of the rule-of-thumb selectors.

Cross-validation (*CV*) is a fairly general statistical learning method. In the KDE context, *CV* selectors are used in conjunction with the integrated squared error (*ISE*) to obtain a criterion, which does not involve the unknown density [78, 6]. Under (A1) and for equally weighted observations, the *ISE* criterion using *CV* can be written as

$$\begin{aligned} CV(h) =& \frac{1}{K^2 h} \sum_k \sum_{k'} \mathcal{K} * \mathcal{K} \left(\frac{s_k - s_{k'}}{h}\right) \\ & - \frac{2}{K(K-1)} \sum_k \sum_{k' \neq k} \mathcal{K}\left(\frac{s_k - s_{k'}}{h}\right), \end{aligned} \tag{2.10}$$

where $*$ stands for the convolution operation. The *CV* method fulfills also an optimality property: it can be shown that $ISE(\hat{h}_{CV}) \xrightarrow[K\to\infty]{} \min_h ISE(h)$ [6].

Plug-in and cross-validation methods are applicable in the multivariate case too. However, estimation of multiple bandwidth parameters is mathematically and computationally much more involved as compared to the scalar case. Except for some special cases as in the rule-of-thumb type of plug-in selectors, the multivariate problem has no direct univariate analogue. Furthermore, one should also keep in mind that there is no application-independent way of finding the optimal bandwidth parameter. The schemes mentioned above try to minimize *AMISE* (plug-in methods) or *ISE* (*CV* methods) to estimate optimal bandwidths. However optimality here depends on the criterion chosen and not on the closeness to a true underlying parameter. For instance, even asymptotically optimal criteria may show bad behavior in simulations [6]. There is no guarantee that these criteria would yield the best performance for the given application. To us, the best answer to this challenging question can be obtained via performance-driven experimentation.

Bandwidth Selection in Context

After a classical account of bandwidth selection, we now focus on strategies suitable for our application. Note that the above discussion dealt with setting the bandwidth for just one set of observations. In the context of density-based shape description for retrieval, on the other hand, since each shape induces a set of observations, we have to select as many bandwidth parameters as the number of shapes in the application database. We consider three levels of analysis at which the parameters in the bandwidth matrix H_k can be chosen. We show the experimental results of these options in Section 4.1.

(1) *Triangle-level*: This option allows a distinct bandwidth matrix for each triangle in the mesh (each inducing an observation s_k), hence it is a variable bandwidth model. As discussed above, this choice is very general since it does not make any assumptions about the shape of the kernel function and hence about the shape of the kth triangle. In general, finding a KDE bandwidth matrix specific to each observation is a difficult problem [85]. For the Gaussian kernel, however, the estimation of the bandwidth matrix H_k can be considered as the estimation of the feature covariance matrix over a given triangle. The moment formula in Eq. 2.1 and its numerical approximation in Eq. 2.2 can be easily adapted for moments of any order. For example, the (i,j)th component h_{ij} of H can be computed by

$$h_{ij}^2 = \iint_\Omega S_i(x,y)S_j(x,y)f(x,y)dxdy - \iint_\Omega S_i(x,y)f(x,y)dxdy \iint_\Omega S_j(x,y)f(x,y)dxdy, \quad i,j = 1,\ldots,m.$$

Note however, the generality of this variable bandwidth model might become overly restrictive as the approach relies too much on the triangulation of the mesh. The shapes of the confidence ellipsoids, hence the uncertainty about the observations are determined solely by the available triangulation. Consequently, the descriptors might overfit the observations and become too much discriminating.

(2) *Mesh-level*: The second option is to use a fixed bandwidth matrix for all triangles in a given mesh, but different bandwidths for different meshes. This choice implicitly assumes that all confidence ellipsoids have the same shape for a given set

of observations obtained from a mesh. In this case, the bandwidth matrix for a given feature can be obtained from its observations using Scott's rule-of-thumb [6]: $H_{\text{Scott}} = \left(\sum_k w_k^2\right)^{1/(m+4)} \hat{\Sigma}^{1/2}$, where m is the dimension of the feature, $\hat{\Sigma}$ is the estimate of the feature covariance matrix and w_k is the weight associated to each observation. Notice that this expression is a generalization of Eq. 2.9 for arbitrarily weighted observations, i.e., w_k's do not need to be $1/K$ (see Appendix C.3 for our derivation). Recall that Scott's rule-of-thumb is proven to provide the optimal bandwidth in terms of estimation error when the kernel function and the unknown density are both Gaussian. Although, there is no guarantee that feature distributions to be Gaussian, Scott's rule-of-thumb provides us with a simple working alternative.

(3) *Database-level*: In the last option, the bandwidth parameter is fixed for all triangles and meshes. Setting the bandwidth at database-level has the implicit effect of regularizing the resulting descriptors by the same amount. A pilot set of parameters is estimated from a representative subset of the database by averaging the Scott bandwidth matrices over the selected meshes. The bandwidth is then optimized based on its retrieval performance on a given training database, that is, by experimenting with a range of bandwidth values centered at the average Scott bandwidth (see Section 4.1). Notice, however, that this option might not be the best choice from a shape modeling perspective. The resulting descriptor might be inaccurate in representing the true feature pdf conditioned on the given shape. In this case, the bandwidth should be viewed as an application-driven hyperparameter of the algorithm. Imposing a fixed bandwidth for the whole database corresponds to a great simplification. Such simplifications are common in machine learning and are both experimentally and theoretically shown to increase the generalization performance of a pattern recognition algorithm [13]. In the present case, a different bandwidth for each mesh risks of overly adapting the descriptor to its corresponding mesh, thus unnecessarily emphasizing individual shape details. However, for discrimination in a relatively large database consisting of various inhomogeneous classes, we have to smooth out (or eliminate) the individual aspects and emphasize the shared ones.

2.2.3 Computational Considerations

The computational complexity of KDE using directly Eq. 2.4 is $O(KN)$ where K is the number of observations (the number of triangles in our case) and N is the number of density evaluation points, i.e., targets. For applications like content-based retrieval, the $O(KN)$-complexity is prohibitive. However, when the kernel function in Eq. 2.4 is chosen as Gaussian, we can use the fast Gauss transform (FGT) [8, 9] to reduce the computational complexity significantly. To give an example, on a Pentium 4 PC (2.4GHz CPU, 2 GB RAM) and for a mesh of 130,000 triangles, the direct evaluation of the $(R, \hat{\mathbf{R}})$-descriptor (1024-point pdf) takes 125 seconds. With FGT, on the other hand, the $(R, \hat{\mathbf{R}})$-descriptor computation takes only 2.5 secs. FGT is an approximation scheme enabling the calculation of large sums of Gaussians within reasonable accuracy and reducing the complexity down to $O(K + N)$. In our 3D shape description system, we have used an improved version of FGT implemented by C. Yang [9].

For the sake of completeness, here, we only provide the conceptual guidelines of the FGT algorithm (see [8, 9] for mathematical and implementation details). FGT is a special case of the more general fast multi-pole method [8], which trades off computational simplicity for

acceptable loss of accuracy. The basic idea is to cluster the source points and target points using appropriate data structures and to replace the large sums with smaller ones that are equivalent up to a given precision. The gain in complexity is achieved by avoiding the computation of every Gaussian at every evaluation point unlike the direct approach, which has $O(KN)$-complexity. Instead, truncated Hermite series (each of which is equivalent to an exponential in the sum) are constructed about a small number of source cluster-centers in $O(K)$ operations. The accuracy can be controlled by the truncation order. These series are then shifted to target cluster-centers, and evaluated at the K targets in $O(N)$ operations. Basically, since the two sets of operations are disjoint, the total complexity of FGT becomes $O(K+N)$.

2.3 Descriptor Manipulation Tools

2.3.1 Marginalization

Features can be selectively removed from the multivariate pdf-based descriptor by marginalization, that is, by integrating out feature variables. Specifically, to remove the component S_i from the pdf of some m-dimensional feature $S = (S_1, \ldots, S_i, \ldots, S_m)$, we use the following equation:

$$f_{S_{i'}|O} \triangleq f_{S_{i'}}(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_m|O), \tag{2.11}$$

$$= \int_{S_i} f_S(s_1, \ldots, s_i, \ldots, s_m|O)\, ds_i, \tag{2.12}$$

which gives the pdf of a "reduced" feature $S_{i'} \triangleq (S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_m)$. Reducing the descriptor $f_{S|O}$ to $f_{S_{i'}|O}$ saves us one dimension at the cost of losing any information brought by the component S_i. For instance, marginalizing the magnitude component R from the pdf of $(R, \hat{\mathbf{R}})$-feature vector, the size of the descriptor $\mathbf{f}_{(R,\hat{\mathbf{R}})|O}$ is N_R times smaller than that of $\mathbf{f}_{(R,\hat{\mathbf{R}})|O}$ since the target set for $(R, \hat{\mathbf{R}})$ contains $N_R \times N_{\hat{\mathbf{R}}}$ points. We can then hope to identify features that can be marginalized for the sake of dimensionality reduction, while monitoring the descriptor's discrimination ability. An obvious instance is the case of a redundant component in the directional parts $\hat{\mathbf{R}}$ and $\hat{\mathbf{N}}$ of the local features $(R, \hat{\mathbf{R}})$ and $(D, \hat{\mathbf{N}})$ respectively. For example, $\hat{\mathbf{R}}$ is unit norm with $\hat{R}_x^2 + \hat{R}_y^2 + \hat{R}_z^2 = 1$, hence given any two components, say $\hat{R}_x$ and $\hat{R}_y$, the third one $\hat{R}_z$ is completely determined up to the sign. Thus, it can be conjectured that $\hat{R}_z$ can be marginalized out without deteriorating performance. We show this experimentally in Section 4.5.1.

Marginalization is an appealing tool, which can be used for more interesting tasks than dimensionality reduction. In the introduction of this chapter and in Section 2.1.1, we have pointed out that our framework can produce a family of descriptors using different sets of multivariate features. The specificity of a feature increases as a function of the included components, hence, in principle, the corresponding pdf-based descriptor should become more discriminating as more components are added. The critical issue is to determine how much the feature is to be specific, or equivalently, which components are to be included in the final multivariate feature to obtain the best performance for a particular classification and/or retrieval task. This problem can be viewed as a particular instance of feature selection, for which many suboptimal heuristics have been developed [86]. Given L different features and $|\mathcal{O}|$ objects, a naive solution is to construct all $2^L - 1$ feature combinations and then to calculate $2^L - 1$ sets of $|\mathcal{O}|$ pdf-based descriptors for subsequent performance

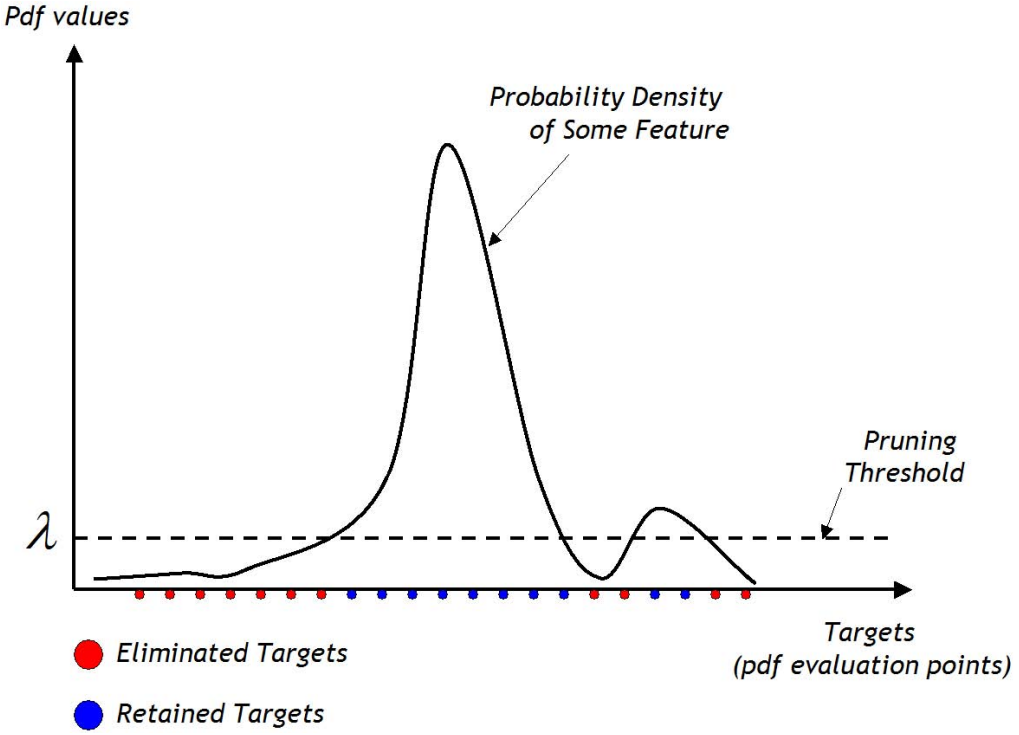

Figure 2.8: Illustration of probability density pruning on a univariate example

evaluation. Using marginalization, we can avoid this laborious descriptor computation phase by computing just one set of descriptors, without resorting to any heuristics. This one set of descriptors corresponds to the case in which all of the L features are included. We can then invoke the marginalization equation (Eq. 2.12) to obtain the descriptors corresponding to all subsets of the L features. Evidently, this procedure is exact (within pdf estimation accuracy) and way more efficient than the naive solution. We employ this particular facet of marginalization in Section 4.6.2.

2.3.2 Probability Density Pruning

Another approach to reduce a descriptor's dimensionality involves identifying insignificant targets (density evaluation points) by pruning the prior feature density, defined in Eq. 2.3. This idea is effected as follows. For a selected threshold λ, we define a new target set $\overline{\mathcal{R}}_S^\lambda = \{t_n \in \overline{\mathcal{R}}_S : f_S(t_n) > \lambda\}$ (see Figure 2.8). The reduced descriptor for some object O becomes then $\mathbf{f}_{S|O}^\lambda = [f_S(t_n|O)]_{t_n \in \overline{\mathcal{R}}_S^\lambda}$. In other words, we eliminate the targets at which prior density values are below the threshold λ. We conjecture that these targets are not significant in describing the shapes as they do not induce large pdf values on average, hence they can be pruned from the descriptors. Notice that heuristic selection methods are not practical when the descriptor vector size is in the order of thousands. Pruning by suppressing small pdf values, albeit not tantamount to feature selection, still serves the goal by reducing the descriptor size. Note also that once insignificant target points are eliminated by pruning, we can use the new target set $\overline{\mathcal{R}}_S^\lambda$ for KDE. $\overline{\mathcal{R}}_S^\lambda$ having smaller cardinality than the original $\overline{\mathcal{R}}_S$, we can also save from descriptor computation time.

2.4 An Invariant Similarity Measure for Pdfs

In this section, we deliver an important result on recovering invariance against certain types of object transformations at the matching stage. We start by recalling the rule relating two pdfs under transformations of random variables. Consider a generic random variable $S \in \mathcal{R}_S$ with pdf f_S. Let Γ be a bijection on $\mathcal{R}_S$. The pdf of S is related to the pdf of its transformed version $\Gamma(S)$ by

$$f_S(s) = f_{\Gamma(S)}(\Gamma(s))|J_\Gamma(s)|, \tag{2.13}$$

where J_Γ is the Jacobian of Γ. Recall that the Jacobian of a transformation Γ is the determinant of the matrix of its first partial derivatives.

The above identity may have several applications in our context. For instance, consider the situation where we have a 3D object O embedded in a certain reference frame $\mathbb{F}$ and we dispose of the pdf $f_{S|O} \triangleq f_S(\cdot|O)$ of a pose-dependent feature S evaluated on the surface of this object. The radial direction feature $\hat{\mathbf{R}}$ and the normal feature $\hat{\mathbf{N}}$ can be two such features. Suppose furthermore that we observe the same object O but in a different reference frame $\mathbb{F}'$. Let the transformation Γ be the mapping from $\mathbb{F}$ and $\mathbb{F}'$, that is, applying Γ on O in $\mathbb{F}$, we can obtain O in $\mathbb{F}'$. In this specific case, Γ is an orthogonal transformation, accounting for rotations, relabelings, and mirror reflections of the coordinate axes. Clearly, applying Γ on the object O does not change its intrinsic shape properties neither the shape information contained in the feature S. The sole effect of this transformation is to map S, evaluated in $\mathbb{F}$, to $\Gamma(S)$ evaluated in $\mathbb{F}'$. By Eq. 2.13 and the fact that $|J_\Gamma(s)| = |\Gamma| = 1, \forall s \in \mathcal{R}_S$, in this specific case, we can relate the pdfs $f_{S|O}$ and $f_{\Gamma(S)|O}$ of the feature S, evaluated when the object O is in $\mathbb{F}$ and $\mathbb{F}'$ respectively, by the following identity

$$f_S(s|O) = f_{\Gamma(S)}(\Gamma(s)|O), \forall s \in \mathcal{R}_S. \tag{2.14}$$

The problem of recovering the pose of an object O in $\mathbb{F}'$ with respect to $\mathbb{F}$ boils down to finding Γ that satisfies the identity in Eq. 2.14. In principle, we can solve this problem by invoking a distance measure between $f_{S|O}$ and $f_{\Gamma(S)|O}$ such as the L^1-distance and setting it to zero:

$$\int_{s \in \mathcal{R}_S} \left| f_S(s|O) - f_{\Gamma(S)}(\Gamma(s)|O) \right| ds = 0. \tag{2.15}$$

Whenever the pdfs $f_{S|O}$ and $f_{\Gamma(S)|O}$ are available, Eq. 2.15 can be solved with respect to Γ, either numerically or analytically.

Guided by the above discussion, we can define a similarity[2] measure between two different objects O and O' as follows:

$$dist(O, O') = \min_{\Gamma \in \mathbf{O}(3)} \int_{s \in \mathcal{R}_S} \left| f_S(s|O) - f_{\Gamma(S)}(\Gamma(s)|O') \right| ds, \tag{2.16}$$

where $\mathbf{O}(3)$ is the group of orthogonal transformations in $\mathbb{R}^3$. This measure is invariant against the extrinsic effects of rotations, relabelings and mirror reflections of the coordinate axes and captures solely intrinsic shape variations to the extent encoded by the feature pdfs.

Let us now explain how and in which situations we can implement this measure for density-based descriptors. Observe that Eq. 2.16 is quite implicit and assumes that the pdfs

[2] Actually a dissimilarity measure.

are analytically available and continuous. Recall, however, that a density-based descriptor is just a discretized version of the continuous pdf, i.e., it is a finite vector given by $\mathbf{f}_{S|O} = [f_S(t_1|O), \ldots, f_S(t_N|O)]$ for the object O and $\mathbf{f}_{S|O'} = [f_S(t_1|O'), \ldots, f_S(t_N|O')]$ for the object O'. Recall also that the set $\overline{\mathcal{R}}_S = \{t_n \in \mathcal{R}_S\}_{n=1}^N$ is the target set at which the pdf is evaluated. With this notation, the discretized version of the measure in Eq. 2.16 would be

$$dist(O, O') = \min_{\Gamma \in \mathbf{O}(3)} \left\| \mathbf{f}_{S|O} - \mathbf{f}_{\Gamma(S)|O'} \right\|_{L^1} \tag{2.17}$$

$$= \min_{\Gamma \in \mathbf{O}(3)} \sum_{n=1}^{N} \left| f_S(t_n|O) - f_{\Gamma(S)}(\Gamma(t_n)|O') \right|, \tag{2.18}$$

provided that $\overline{\mathcal{R}}_S$ is a uniform partitioning of $\mathcal{R}_S$ (cf. Section 2.1.3). There remains one important problem regarding the calculation of this measure. To perform the minimization in (2.17) with respect to Γ, we need the pdf values $f_{\Gamma(S)}(\Gamma(t_n)|O')$ but we only have $f_S(t_n|O')$. Hopefully, whenever $\overline{\mathcal{R}}_S$ is closed under the action of Γ (i.e., when $\forall t \in \overline{\mathcal{R}}_S, \Gamma(t) = t' \in \overline{\mathcal{R}}_S$), Γ maps the target t to t', which is also in the target set. By Eq. 2.14, we have $f_S(t|O') = f_{\Gamma(S)}(t'|O')$. Thus, given the descriptor $\mathbf{f}_{S|O'}$ and a certain Γ, we can recover $\mathbf{f}_{\Gamma(S)|O'}$ by just permuting the vector entries, without re-evaluating the descriptor for every admissible Γ. Evaluation of Eq. 2.17 becomes a matter of testing admissible transformations Γ (under the action of which the target set $\overline{\mathcal{R}}_S$ is closed) and picking the minimum value.

The set of admissible transformations Γ needs further attention as it determines the type of invariances we want to achieve at the matching stage. In Eqs. 2.16 and 2.17, this set is specified as the group of orthogonal transformations $\mathbf{O}(3)$, which includes rotations, relabelings and mirror reflections of the coordinate axes. From a practical viewpoint, the minimization involved in the discrete case of Eq. 2.17 can only be performed by an exhaustive search on the set of admissible Γ. Thus, the cardinality of the search space matters a lot in order to obtain a measure that can be computed within reasonable time. Accordingly, we focus on developing a similarity measure invariant to relabelings and mirror reflections (but not to rotations) due to following reasons:

- Rotations form a subgroup of $\mathbf{O}(3)$, and are called as special orthogonal transformations $\mathbf{SO}(3)$, which contains infinitely many Γ. Thus in the continuous case of Eq. 2.16, the minimization is to be performed within an infinite set. In the discrete case that we are particularly concerned with, the cardinality of the search space depends on the granularity of the target set, that is, how fine $\mathcal{R}_S$ is partitioned into $\overline{\mathcal{R}}_S$. The larger the target set is, the larger the set of admissible rotations becomes. On the other hand, the set of admissible axis relabelings and mirror reflections in $\mathbb{R}^3$, denoted by $\mathbf{PS}(3)$, has finite cardinality both in the continuous and discrete cases. A labeling means which one of the three coordinate axes is the x-axis, which one is the y-axis and which one is the z-axis. A mirror reflection changes their polarity. The three axes of the 3D coordinate system can be labeled in $3! = 6$ possible ways. Given a certain labeling, there are $2^3 = 8$ possible polarity assignments. Thus, there are $6 \times 8 = 48$ possible xyz-configurations. The passage from one axis configuration to another can be effected with a matrix Γ, which consists of the product of a permutation matrix (for relabelings) and a sign matrix (for reflections), as illustrated in Figure 2.9.

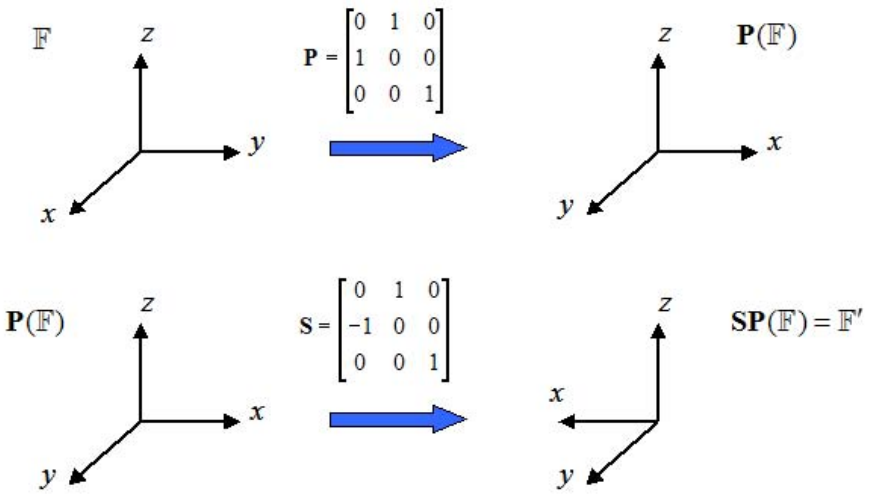

Figure 2.9: A permutation **P** followed by a polarity reversal **S** changes the axis configuration form $\mathbb{F}$ to $\mathbb{F}'$.

- In descriptor-based 3D retrieval, rotation errors constitute a lesser problem than mislabelings or wrong polarity assignments of the coordinate axes. In general, PCA-based object normalization methods correctly find the major axes of the 3D object but fail to unambiguously resolve the labels and/or the polarities of these axes (see Figure 2.10) [87]. After normalization, rotation errors remain in small amount so that the variation between descriptors becomes unimportant, or not that much significant as a mislabeling or a wrong polarity assignment would produce. Furthermore, smoothness reinforced by the KDE-based scheme can gloss over small rotation errors, as discussed in Section 2.2.1.

Figure 2.10: Three airplane models after pose normalization: major axes are correctly found but the front of the fuselage of the rightmost model is in the opposite direction.

The importance of target selection reappears since the target set $\overline{\mathcal{R}}_S$ should be closed under the action of $\Gamma \in \mathbf{PS}(3)$. Our working example can be pose-dependent features such as $\hat{\mathbf{R}}$ or $\hat{\mathbf{N}}$, taking values within the unit 2-sphere $\mathcal{S}^2$. In this case, the target set $\overline{\mathcal{R}}_{\hat{\mathbf{R}}}$

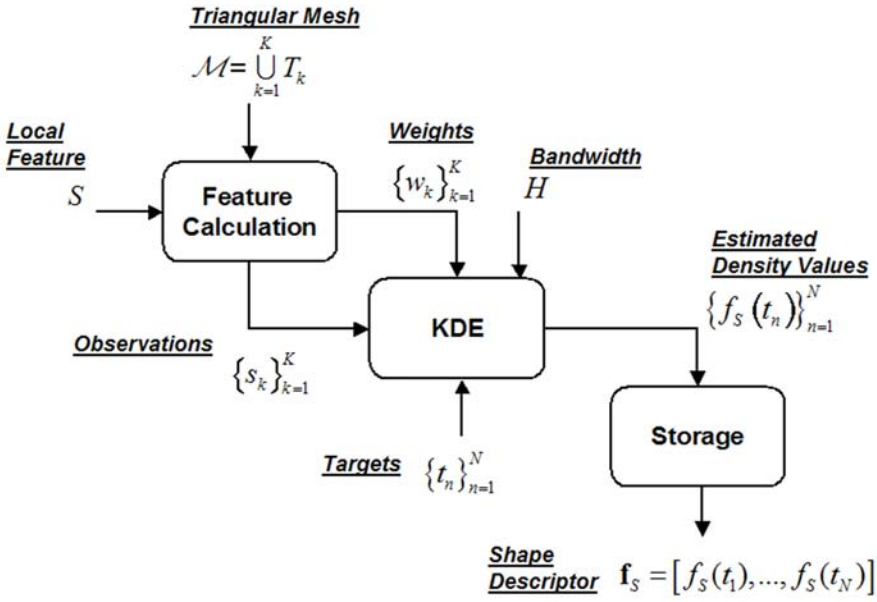

Figure 2.11: Flow diagram of density-based shape description algorithm

is a set of 3D points lying on $\mathcal{S}^2$. Recall that, in Section 2.1.3, we have presented two different schemes for sampling $\mathcal{S}^2$: (i) octahedron subdivision and (ii) spherical coordinate sampling. The target points in $\overline{\mathcal{R}}_{\hat{\mathbf{R}}}$ produced by spherical sampling are symmetric with respect to each of the xy-, xz- and yz-planes, thus $\overline{\mathcal{R}}_{\hat{\mathbf{R}}}$ is closed when Γ is solely a reflection. However, it is not closed under the action of a permutation since the configuration of the parallels and the meridians depends on the specific choice of the xyz-labels. When $\overline{\mathcal{R}}_{\hat{\mathbf{R}}}$ is obtained by octahedron subdivision, on the other hand, the closedness under permutations and reflections is ensured. Consider the regular octahedron with its center placed at the origin and its 6 vertices at $\{(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)\}$. First, the appearance of the octahedron does not depend on the axis labels: after a permutation, we recover the same vertex coordinates. Second, the octahedron is symmetric with respect to each of the xy-, xz- and yz-planes: after a reflection, we recover again the same coordinates. This nice property of the octahedron holds for its successive subdivisions of any order, making the resulting target set $\overline{\mathcal{R}}_{\hat{\mathbf{R}}}$ closed under the action of coupled permutations and reflections.

2.5 Summary of the Algorithm

We summarize below the proposed algorithm to obtain a density-based 3D shape descriptor:

1. For a chosen local feature S, specify a set of targets $t_n, n = 1, \ldots, N$.

2. Normalize the object O represented in terms of a 3D triangular mesh $\mathcal{M} = \bigcup_{k=1}^{K} T_k$ according to the invariance requirements of S.

3. For each mesh triangle T_k, calculate its feature value s_k using Eq. 2.2 and its weight w_k.

4. Set the bandwidth parameters H_k according to the strategy chosen among the three options described in Section 2.2.2.

5. For each target $t_n, n = 1, \ldots, N$, evaluate the feature pdf $f_S(t_n|O)$, using Eq. 2.5.

6. Store the resulting density values $f_S(t_n|O)$ in the shape descriptor $\mathbf{f}_{S|O} = [f_S(t_1|O), \ldots, f_S(t_N|O)]$.

7. If necessary, marginalize and/or prune the descriptor to reduce dimensionality, as described in Section 2.3.

Figure 2.11 depicts the flow diagram of the algorithm when the bandwidth parameters are set at database-level. Alternatively, in the triangle or mesh-level setting, a bandwidth matrix is to be computed for each triangle or for the entire mesh, respectively. Note that in Figure 2.11, we assume that the mesh $\mathcal{M}$ has already undergone a pose and/or scale normalization step depending on the missing invariance properties of the local feature S chosen.

Chapter 3

Statistical Similarity Learning

Ongoing research in 3D shape retrieval shows that no single descriptor is capable of providing satisfactory retrieval performance for a broad class of shape databases, regardless of the associated semantics [2, 23, 3]. Figure 3.1 displays the response of two different descriptors A and B to two different queries from the Princeton Shape Benchmark [23]. The first query is a biplane model and the second one is a chair model. In response to the biplane model, descriptor A returns correctly four biplanes in the first three and in the sixth matches, while the fourth and the fifth retrieved models are not biplanes, but still flying objects that can be considered as relevant. Descriptor B, on the other hand, returns models that are completely irrelevant to the biplane query (three shelf models, two coarse human models and a microscope!). For the chair query, the situation reverts to the advantage of Descriptor B, which can retrieve six chair models; while descriptor A, after first three correct matches, returns two tree models and a monument! Thus, the adequacy of the descriptors A and B depends on the nature of the query. Furthermore, these examples can be multiplied; for instance, there are cases where the sets of relevant matches for different descriptors are disjoint (see Figure 3.2). This suggests that a good retrieval machine should rely on a diverse set of descriptors for robust performance.

As the example in Figure 3.1 illustrates, experimental evidence motivates us to consider fusion schemes to learn similarities between pairs of objects on statistical grounds. In the present chapter, we tackle the similarity learning problem by a score fusion scheme, which minimizes a convex regularized version of the empirical risk associated with ranking instances. We follow the statistical learning framework developed in [10, 11] and provide an application to 3D shape retrieval. We observe that learning a linear scoring function can be cast into binary classification in the score difference domain, that is, finding a separating hyperplane. Given a query, the hyperplane parameters can be considered as optimal with respect to the empirical risk associated with ranking instances.

Our approach can be contrasted to the one in [88] where the ranking problem is formulated as classification of pairwise instances using difference vectors (between descriptors in our setting) inputted to an SVM-type algorithm. Our solution differs from that of [88] in that it operates on scores rather than on the specific representations associated with training instances. We prefer to work with scores due to following reasons:

- Even though most of the 3D shape descriptors proposed in the literature [2, 3] are vector-based, the computation of non-classical similarity measures involves a minimization over a set of plausible transformations in order to secure invariance against certain types of transformations (cf. Section 2.4). As exemplified in Section 2.4,

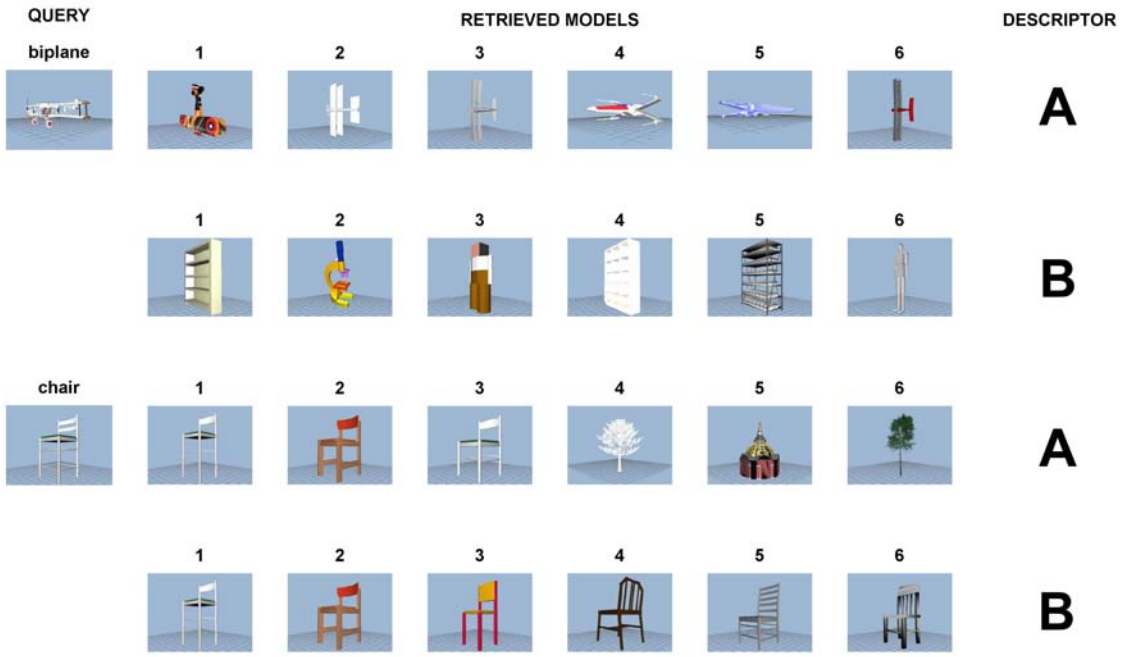

Figure 3.1: Response of two different descriptors A and B to two different queries *biplane* and *chair*

such procedures may require permutations of descriptor vector entries, in which case, working with descriptors rather than with scores would become very hard, if not impossible.

- On a more general vein, descriptors might not be vectors but graphs with specialized metrics for matching. Consequently, working with scores makes our approach a more general solution to statistical similarity learning.

- In the 3D shape retrieval domain, descriptor dimensionality can easily reach the order of thousands. Combining scores has this additional benefit of reducing data dimensionality in an informed way.

There is a considerable body of work concerning statistical ranking, which find application niches in other than the 3D shape retrieval, such as text-based information retrieval and collaborative filtering. Large margin approaches as in [88, 89] considered the problem as one of ordinal regression, in which case relevance labels are treated as categorical variables at ordinal scale. In these approaches, the aim is to find a set of rank boundaries in the feature domain. In [90], the authors adapted the on-line perceptron learning scheme to the ranking problem to cope with situations where training examples are not readily available prior to learning. On a different spirit, the works in [91, 92] treated the problem as combining preference judgments based on a variation of the AdaBoost algorithm for binary classification. Recently in [93], the authors proposed a computationally efficient approximation for learning a ranking function from structured order constraints on sets of training samples.

This chapter is organized as follows. In Section 3.1, we introduce the score fusion problem as well as the notation that we will use subsequently. In Section 3.2, following

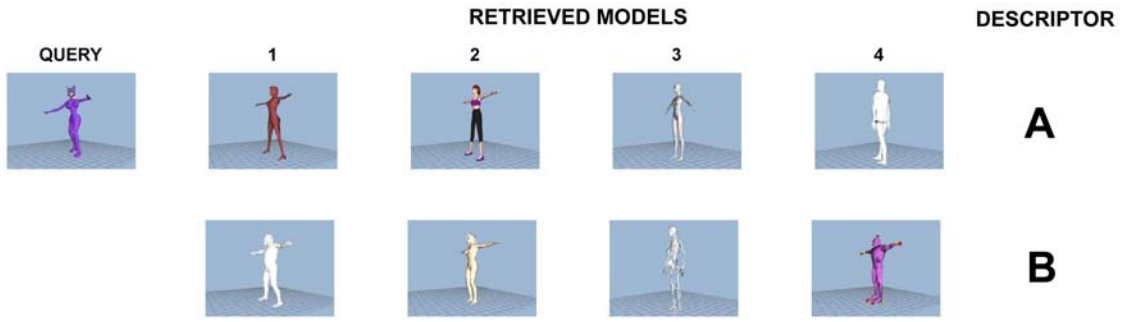

Figure 3.2: Response of two different descriptors A and B to the same query: both descriptors retrieve different plausible matches.

[11], we recapitulate the ranking risk minimization framework. In Section 3.3, we give an SVM-based solution to the score fusion problem. Finally, in Section 3.4, we propose different search protocols for the retrieval machine combining multiple sources of similarity information.

3.1 The Score Fusion Problem

In this section, we introduce the score fusion problem for 3D shape retrieval following the statistical ranking framework developed in [10, 11]. Consider the problem of ranking two generic database shapes x and x' based on their relevance to a query shape q. Suppose that we have access to K similarity values $sim_k(x,q) \triangleq s_k$ and $sim_k(x',q) \triangleq s'_k$ for each of the pairs (x,q) and (x',q) respectively, where $k = 1,\ldots,K$. In our context, each similarity value s_k arises from a different shape descriptor and reflects some, possibly different, geometrical and/or topological commonality of shape pairs (x,q) and (x,q'). For similar shape pairs, an ideal similarity value should be higher than it is for less similar ones. In retrieval problems, a shape x in the database that is more similar to the query q is expected to be ranked higher than another intrinsically less similar shape x'. These similarity values/scores can be written more compactly in the vector form as $\mathbf{s} = [s_1,\ldots,s_K]$ and $\mathbf{s}' = [s'_1,\ldots,s'_K]$ where $\mathbf{s},\mathbf{s}' \in \mathbb{R}^K$. Our objective is to build a scalar-valued scoring function φ of the form $\varphi(x,q) = \langle \mathbf{w},\mathbf{s}\rangle$, where $\mathbf{w} = [w_1,\ldots,w_K] \in \mathbb{R}^K$ is a vector, whose components are associated with the individual scores s_k. The scoring function φ should assign a higher score to the shape, which is more relevant to the query q, i.e., it should satisfy the following property:

$$\begin{array}{ll} \varphi(x,q) > \varphi(x',q) & \text{if } x \text{ is more relevant to } q \text{ than } x', \\ \varphi(x,q) < \varphi(x',q) & \text{otherwise,} \end{array}$$

where ties are arbitrarily broken. The relevance of the shapes x and x' to the query q can be encoded by indicator variables y and y' respectively. In this work, we assume crisp relevances $y = 1$ (*relevant*) and $y = -1$ (*not relevant*), in which case, the above property reads as:

$$\begin{array}{ll} \varphi(x,q) > \varphi(x',q) & \text{if } y - y' > 0, \\ \varphi(x,q) < \varphi(x',q) & \text{if } y - y' < 0. \end{array}$$

The function φ must subsume the similarity information residing in the individual scores s_k in order to emulate the ideal similarity notion between shapes, hence to achieve a better retrieval performance. Given the linear form $\varphi(x, q) = \langle \mathbf{w}, \mathbf{s} \rangle$, score fusion can be formulated as the problem of finding an optimal weight vector $\mathbf{w}$, according to some criterion, as we explain in the following section.

3.2 Ranking Risk Minimization

The criterion of interest is the so-called *ranking risk* defined as

$$R(\varphi; q) = \mathbb{P}\left((\varphi(x, q) - \varphi(x', q)) \cdot (y - y') < 0\right).$$

In other words, the ranking risk with respect to the query q is the probability that φ gives a score for x higher than it gives to x' while x' is relevant to q but x is not. Naturally, we want this probability to be as small as possible. Rewriting the misranking probability in terms of an expectation, we have

$$\mathbb{P}\left((\varphi(x, q) - \varphi(x', q)) \cdot (y - y') < 0\right) = \mathbb{E}\left\{\mathbb{I}\left\{(\varphi(x, q) - \varphi(x', q)) \cdot (y - y') < 0\right\}\right\},$$

where $\mathbb{I}\{\cdot\}$ is one if the predicate inside the braces is true and zero otherwise. Since we do not have access to the probability measure $\mathbb{P}$ on (x, y), we have to find an empirical estimate of the ranking risk based on a training set $\mathcal{X} = \{(x_n, y_n)\}_{n=1}^{N}$. A natural estimate for this expectation can be obtained by averaging over the pairwise instances of the training set $\mathcal{X}$ [10, 11], i.e.,

$$\hat{R}(\varphi; q) = \frac{2}{N(N-1)} \sum_{m<n} \mathbb{I}\left\{(\varphi(x_m, q) - \varphi(x_n, q)) \cdot (y_m - y_n) < 0\right\}. \tag{3.1}$$

The properties of this empirical quantity, known as a U-statistic in the statistics literature, has been investigated in the context of statistical ranking [10, 11]. This particular type of risk functional uses the 0-1 loss to assess the cost of a misranked pair of objects, i.e., if $\varphi(x_m, q) < \varphi(x_n, q)$ and $y_m > y_n$, the scoring function $\varphi(\cdot, q)$ (wrongly) assigns a higher score to x_n than to x_m while x_m is relevant to the query q but x_n is not. Thus the scoring function has made an error in ranking x_n and x_m with respect to the query q. Such misrankings are naturally undesirable and our task is to find a scoring function (or more appropriately its parameters $\mathbf{w}$) so that the number of misranked pairs is as small as possible.

3.3 SVM Formulation

We can identify the empirical ranking risk in Eq. 3.1 as an empirical classification error. Let $z \triangleq (y - y')/2$, taking values within $\{-1, 0, 1\}$. We observe then the following:

$$z = \begin{cases} 1 & x \text{ should be ranked higher than } x', \\ -1 & x \text{ should be ranked lower than } x'. \end{cases}$$

When $z = 0$, i.e., if shapes x and x' are both relevant ($y = y' = 1$) or both *not* relevant ($y = y' = -1$), we have no particular preference in ranking them with respect to each other (we can decide arbitrarily). Corresponding to each non-zero z, we can define *a score*

difference vector $\mathbf{v}$, which is given simply by $\mathbf{v} \triangleq \mathbf{s} - \mathbf{s}'$, the difference between the score vectors $\mathbf{s}$ and $\mathbf{s}'$ of the shapes x and x' respectively. With this new notation and writing the scoring function φ explicitly in terms of its parameters $\mathbf{w}$, Eq. 3.1 now reads as

$$\hat{R}(\mathbf{w}; q) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}\left\{ z_t \left\langle \mathbf{w}, \mathbf{v}_t \right\rangle < 0 \right\}, \tag{3.2}$$

where $(\mathbf{v}_t, z_t)$ corresponds to pairs of shapes x_m and x_n whose respective relevance labels y_m and y_n are different (z_t is either 1 or -1), and T is the total number of such pairs. Thus, we have converted the empirical ranking risk written in terms of *score* vectors $\mathbf{s}$ and *relevance* indicators y (Eq. 3.1) into an empirical classification error in terms of *score difference* vectors $\mathbf{v}$ and *rank* indicators z (Eq. 3.2). In both cases, the sought after parameter vector $\mathbf{w}$ is the same. Replacing the 0-1 loss in Eq. 3.2 with a convex loss function and adding a regularization term on some norm of $\mathbf{w}$, one obtains a tractable convex optimization problem in $\mathbf{w}$ [14, 94]. In particular, using the *hinge* loss as the convex loss and the L^2-norm as the regularization term leads to the well-known C-SVM problem, formulated as

$$\underset{\mathbf{w} \in \mathbb{R}^K}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{t=1}^{T} \xi_t, \tag{3.3}$$

$$\text{subject to} \quad z_t \left\langle \mathbf{w}, \mathbf{v}_t \right\rangle \geq 1 - \xi_t, \forall t = 1, \ldots, T \tag{3.4}$$

$$\xi_t \geq 0, \forall t = 1, \ldots, T. \tag{3.5}$$

where $\xi_t, t = 1, \ldots, T$ are the slack variables and C is a constant trading off the classifier complexity vs. the training error [94]. In summary, the problem of finding the parameter vector $\mathbf{w}$ of the linear scoring function φ is the same as the C-SVM problem in the domain of score difference vectors. For a suitably chosen C, the solution of the above convex minimization is

$$\hat{\mathbf{w}} = \frac{1}{T'} \sum_{t'=1}^{T'} \alpha_{t'} z_{t'} \mathbf{v}_{t'}, \tag{3.6}$$

where $\alpha_{t'}$ are the non-zero Lagrange multipliers (corresponding to the constraints (3.4)) [14, 94]. The index t' runs over such non-zero multipliers and the corresponding score difference vectors $\mathbf{v}_{t'}$ are the so-called *support vectors*. The key point here is that *the weight vector learned by SVM in the score difference domain can directly be used to evaluate the scoring function at the matching stage.*

We can now summarize the training algorithm to learn the parameter $\mathbf{w}$ of the scoring function φ:

Given Database shapes $x_n, n = 1, \ldots, N$
A query q
Relevance labels y_n
K different shape description schemes

Calculate Calculate a score vector $\mathbf{s}_n \in \mathbb{R}^K$ for each (x_n, q)-pair.

Identify The pairs of labels (y_m, y_n) such that $y_m - y_n \neq 0, \forall m, n = 1, \ldots, N$.
Let t run over such (m, n).

Construct The score difference vectors $\mathbf{v}_t$ and their rank indicators z_t.

Run The C-SVM algorithm to learn the weight vector $\mathbf{w} \in \mathbb{R}^K$, using the set $\{(\mathbf{v}_t, z_t)\}_{t=1}^T \subset \mathbb{R}^K \times \{-1, 1\}$.

3.4 Applications

In this section, we illustrate our score fusion scheme for similarity learning in two different retrieval protocols: (i) *bimodal search* and (ii) *two-round search*.

3.4.1 Bimodal Search

In this scenario, the user provides a textual description associated with the query shape. The keyword can be selected from one of the predefined class concepts. We call this protocol as *bimodal* since the query is formulated in terms of two information modalities, a 3D shape and a concept keyword. This protocol necessitates an off-line stage during which the weight vectors associated with each class concept are learned. Note that the criterion that we have elaborated on in the previous sections is *per-query* and should be extended to a *per-concept* risk $\hat{R}(\mathbf{w}, \mathcal{C})$, where $\mathcal{C}$ stands for the working concept. This can be done straight-forwardly by averaging per-query risks associated with a given concept, that is,

$$\hat{R}(\mathbf{w}; \mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{q \in \mathcal{C}} \hat{R}(\mathbf{w}; q), \tag{3.7}$$

where $|\mathcal{C}|$ is the number of training shapes belonging to $\mathcal{C}$. However, since the minimization should be performed in the score difference domain, the problem turns out to be a very large-scale one even for moderately sized classes. Given a training database $\mathcal{D}$ of size $|\mathcal{D}|$, the number of score difference instances per concept is $|\mathcal{C}| \times (|\mathcal{C}| - 1) \times (|\mathcal{D}| - |\mathcal{C}|)$, e.g., for $|\mathcal{D}| = 1000$ and for $|\mathcal{C}| = 10$, the number of training instances becomes ~ 90000, in which case we incur to memory problems using standard SVM packages [95]. In order to maintain the generality and practical usability of our approach in this protocol, we develop two heuristics:

(1) **Average per-query weight vector.** The weight vector $\hat{\mathbf{w}}_{\mathcal{C}}$ for a given concept class is computed as the average of the per-query weight vectors corresponding to the training models within that class, that is,

$$\hat{\mathbf{w}}_{\mathcal{C}} = \frac{1}{|\mathcal{C}|} \sum_{q \in \mathcal{C}} \hat{\mathbf{w}}_q \tag{3.8}$$

where $\hat{\mathbf{w}}_q$ is the weight vector for query q, obtained by minimizing the convex regularized version of Eq. 3.2. We denote this heuristic by *AVE-W*.

(2) **Per-class risk minimization using per-query support vectors.** In this second heuristic, we exploit the sparsity of the SVM solution, which means that the per-query weight vector given in Eq. 3.6 is the weighted sum of usually a much smaller number of training instances than the size of the whole training set. It is a well known fact that, for a given problem, the SVM solution remains the same when only the support vectors are provided for minimization [14, 94]. By this token, support vectors can also be called as "difficult instances" of the training problem. They reflect the distribution of training data in an economical way, as they are fewer in number than the whole training instances. Accordingly, the learning of a per-concept weight

vector can be carried in two stages. First, we identify the support vectors of per-query problems by per-query minimization. Then, we pool all the support vectors corresponding to a given concept and perform the minimization using this newly formed set to learn the per-concept weight vector. We repeat this procedure as many times as the number of predefined shape concepts. We denote this heuristic by *PCMIN-W*.

3.4.2 Two-round Search

This protocol requires user intervention during the querying process. In the first round, the retrieval machine returns a ranked list of shapes using a simple scoring scheme, e.g., the sum of the available raw scores. After the first found, we can invoke the score fusion scheme in two different ways:

(1) **On-line.** The user marks M models among the returned ones, as either *relevant* ($y = 1$) or *non-relevant* ($y = -1$) with respect to his/her query. In the second round, the retrieval machine returns a refined ranked list using the scoring function $\varphi_{\mathbf{w}}(x, q) = \langle \mathbf{w}, \mathbf{s} \rangle$. The weight vector $\mathbf{w}$ is learned *on-line* using the M marked shapes as training instances. In order not to demand too much from the user, M should not be large and is typically limited to a few first instances. For example, when $M = 8$ and the number of positive M^+ and negative instances M^- are equal ($M^+ = M^- = 4$), the total number of training score difference vectors is just 16. Consequently, on-line learning is computationally feasible.

(2) **Off-line.** In this variant of the two-round search, all the shapes in the database have their individual weight vectors stored, which have already been learned *off-line*. The individual weight vectors can be obtained as in the bimodal protocol. At querying time, the user is asked to mark just the first relevant item in the displayed page of the results. The second round evaluates the scoring function $\varphi_{\mathbf{w}}(x, q) = \langle \mathbf{w}, \mathbf{s} \rangle$ using the weight vector corresponding to the marked shape. Clearly, this protocol does not perform any on-line learning and constitutes a less demanding option than the former in terms of user interaction needed, as the user is asked to mark just one item.

The two variants of the two-round protocol presented here constitute particular forms of relevance feedback and prove to be more suitable applications for our similarity learning scheme than bimodal search, as the experiments in Section 4.10 demonstrate.

Chapter 4

Experiments

In this chapter, we concretize the density-based shape description framework and the score fusion based similarity learning approach in experimental terms. Our analysis extends over ten different sections clustered into the following groups:

- **Parameter Selection and Robustness Experiments.** Bandwidth parameter selection constitutes the major issue in all applications based on KDE. As a starting point, in Section 4.1, we experiment with possible bandwidth selection strategies introduced for 3D shape description (cf. Section 2.2.2), since the results of all the subsequent analyses rely upon the bandwidth parameter. In Section 4.2, we illustrate the robustness properties of the density-based framework with an emphasis on the effect of the bandwidth in conjunction with the discussion in Section 2.2.1 on KDE in context.

- **Efficiency Experiments.** We treat the efficiency problem as a matter of descriptor parsimony. Target selection experiments in Section 4.3 deal with feature domain sampling, as well as with descriptor size. In Section 4.5, we experiment with our marginalization and probability density pruning tools for descriptor dimensionality reduction. We also compare our findings with those of the classical PCA approach.

- **Effectiveness Experiments.** Effectiveness refers to the discrimination ability of the descriptors in a retrieval application. In Section 4.4, we show how the invariant matching scheme introduced in Section 2.4 outperforms standard similarity measures. Sections 4.6 and 4.7 invoke feature-level and score-level information fusion to further improve the retrieval performance of the density-based framework.

- **Comparison Experiments.** Quite many shape descriptors in the literature have analogies to cases of the density-based framework in the local surface features they use. Our approach differs from these, which we denominate collectively as *histogram-based*, in the use of KDE, in which case multivariate extensions come naturally and prove to be more descriptive for the retrieval task as shown in Section 4.8. We also compare our scheme against its state-of-the-art competitors. In another comparison study in Section 4.9, we analyze the performance variation of our descriptors across databases differing in classification properties, class cardinalities and mesh qualities. These databases are (see also Section 1.2):

  – *Princeton Shape Benchmark* (PSB) [23]. 1814 general-purpose low-quality 3D models split into two subsets: Training (907 models in 90 classes) and Test

(907 models in 92 classes). Classification is induced by functionality as well as by form. In general, the meshes in PSB have low resolution, they are non-regular, non-smooth and disturbed with degeneracies such as non-manifold, non-connected triangles of varying size and shape. These models are usually called as "triangular soups".

  - *Sculpteur* (SCU) [24, 25]. 513 high-quality 3D models in 53 classes consisting largely of archaeological models. The meshes in SCU are regular, smooth and highly detailed in terms of resolution.
  - *SHREC Watertight* (SHREC-W) [26]. 400 high-quality 3D models in 20 classes. The specialty of this database is that the classification assumes topological equivalences in addition to geometrical, constituting a challenging test environment for geometry-induced description methods. The meshes in SHREC-W are regular and smooth.
  - *Purdue Engineering Shape Benchmark* (ESB) [27]. 865 3D models of engineering parts in 45 classes. The meshes in ESB are regular but in general non-smooth due to the general platonic solid form of the engineering parts.

- **Statistical Score Fusion Experiments.** This last cluster addresses the similarity learning algorithm for retrieval (Section 4.10). We test the bimodal and two-round search protocols introduced in Chapter 3 and show that more effective 3D retrieval is possible when a moderate amount of supervision is incorporated into the querying process.

We mostly experiment with PSB and to a lesser extent with SCU. Nevertheless, the extensive Section 4.9 reflects the virtues of our approach on SHREC-W and ESB. Before proceeding with the forthcoming sections, we need to make a clarification on object normalization. The local surface features presented in Section 2.1.1 have different invariance properties. Consequently, whenever a certain feature is invariant to one of the so-called extrinsic effects such as translation, rotation, mirror reflection and isotropic rescaling, the 3D object should be normalized prior to descriptor extraction. In our experiments, we had to carry out the following normalization steps:

- **Translation.** The object's center of mass is considered as the origin of the 3D coordinate system. We calculate the center of mass as the area-weighted average of the triangle barycenters.

- **Rotation.** To find the three axes of the coordinate system, we use Vranic's "continuous" PCA approach, where the covariance matrix of the surface points is evaluated by integrating over triangles instead of mere area-weighted averaging [17]. The major axes are then found, as is the standard practice, by an eigendecomposition of the estimated covariance matrix. The x, y, z labels of the axes are assigned in the decreasing order of the eigenvalues, while the polarities are estimated by Vranic's moments-based approach [17]. Note that, with our invariant matching scheme (cf. Section 2.4), we do not need the last two types of normalization, concerning coordinate axes relabelings and mirror reflections.

- **Isotropic Rescaling.** We calculate a scale factor as the area-weighted average of the surface point-to-origin distances. Dividing the surface point coordinates by this factor yields scale normalization.

We hope the reader will enjoy this experimental journey.

Table 4.1: DCG (%) for Possible Bandwidth Selection Strategies on PSB Training Set

	Descriptor		
Bandwidth Setting	*Radial*	*T-plane*	*Torque*
Triangle-level	35.2	-	-
Mesh-level	51.1	51.4	49.9
Database-level	57.0	59.8	55.6

4.1 Bandwidth Selection

Discrimination ability of the density-based approach critically depends upon the judicious setting of the bandwidth parameters. In this first experimental section, we explore the impact of the bandwidth selection strategy on the retrieval performance for the following multi-dimensional local features: *Radial* $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$, *T-plane* $(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ and *Torque* (R, C_x, C_y, C_z).

4.1.1 Levels of Analysis for Bandwidth Selection

Recall that in Section 2.2.2, we have introduced three levels of analysis for bandwidth selection:

(1) *Triangle-level:* The bandwidth parameters are set using the Simpson averaging scheme described in Section 2.1.2. This computationally expensive setting is only tested for the *Radial* descriptor, and is implemented using the KDE toolbox developed by A. Ihler [96] since the available FGT implementation does not allow a different bandwidth per triangle [9]. The KDE toolbox makes use of kd-trees and reduces the computational burden considerably, though not to the extent achieved by FGT.

(2) *Mesh-level:* The bandwidth matrix is distinct for each mesh and is computed using the Scott's rule-of-thumb.

(3) *Database-level:* The bandwidth matrix is the same for all meshes and is computed as the average Scott bandwidth over the meshes.

We have used only diagonal bandwidth matrices $H = diag\,(h_1, \ldots, h_m)$, after observing that, in general, off-diagonal terms are negligible (close to zero) as compared to the diagonal ones: they have no considerable effect on KDE, thus on descriptor entries. The following retrieval results have all been obtained using the L^1-measure. Table 4.1 compares the DCG scores obtained with *Radial*, *T-plane*, and *Torque* descriptors on PSB Training Set. Figure 4.1 shows the precision-recall curves corresponding to mesh and database-level settings for *Radial* and *T-plane*-descriptors. From Table 4.1 and Figure 4.1, we clearly observe that setting the bandwidth H at database-level is a better option than triangle and mesh-level settings.

4.1.2 Sensitivity Results

In Table 4.2, we provide the average Scott bandwidth values obtained from PSB Training meshes for *Radial*, *T-plane*, and *Torque* features. We have also analyzed the DCG

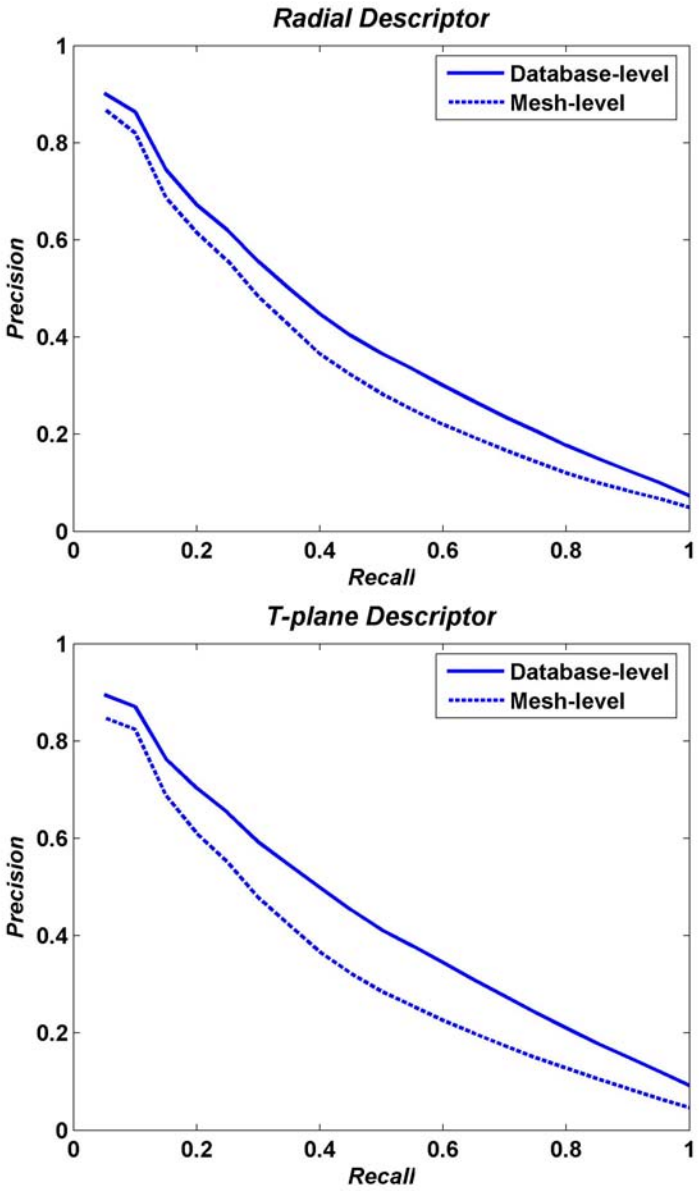

Figure 4.1: Precision-Recall curves with a bandwidth selection made at mesh-level vs. database-level for the *Radial* (top) and *T-plane* (bottom) descriptors on PSB Training Set

performance sensitivity as a function of the database-level bandwidth parameters. Given a database-level Scott bandwidth matrix H, we have carried this analysis by testing the performance of the descriptors obtained using perturbed bandwidth matrices of the form cH where c is a scalar multiplicative perturbation factor. In Figure 4.2, we provide the DCG profile as a function of $c \in [0.1, 2.0]$ with steps of 0.1 for the *Radial* descriptor. From this profile, we can infer the following:

- Within the $c \in [0.9, 1.3]$ interval, the profile remains flat at around DCG=57%. This shows that, under the database-level assumption, averaging Scott bandwidths is a sensible option.

Table 4.2: Average Scott Bandwidths Obtained from the PSB Training Set: $h_i, i = 1, \ldots, 4$ corresponds to the ith component of the feature.

	Bandwidth Parameters			
Descriptor	h_1	h_2	h_3	h_4
Radial	0.199	0.338	0.226	0.141
T-plane	0.202	0.239	0.270	0.285
Torque	0.199	0.153	0.236	0.236

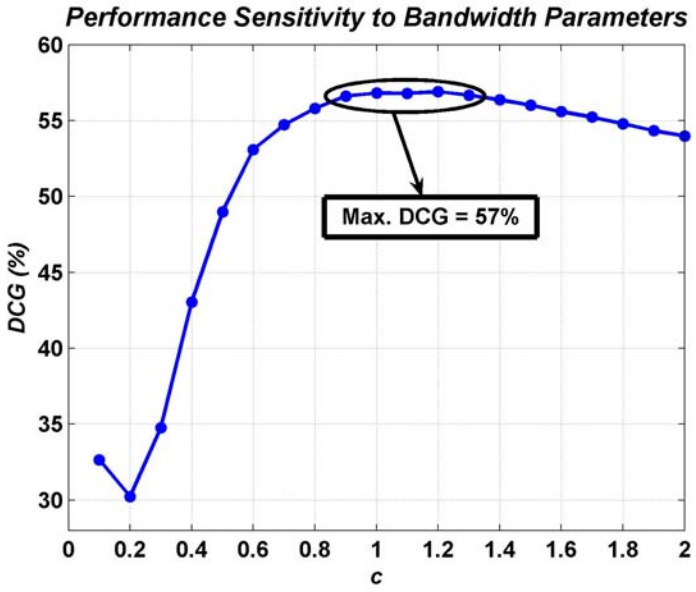

Figure 4.2: DCG Performance Sensitivity of the *Radial* descriptor to bandwidth parameters (see text)

- The performance degrades for small ($c \in [0.1, 0.9]$) and large ($c \in [1.3, 2.0]$) bandwidth values. Small bandwidth values correspond to *under-smoothed* pdf estimates while large values correspond to *over-smoothed* estimates (cf. Section 2.2.1).
- The performance degradation for under-smoothed estimates is much steeper than the over-smoothed ones. We interpret this as an experimental evidence for the regularization behavior of the KDE-based approach. Under-smoothed estimates reflect specific details about the features, hence descriptors become too much object-specific. Over-smoothed estimates, on the other hand, eliminate such details and emphasize commonalities between features/shapes. We deduce that the performance is less sensitive to large bandwidth values than to smaller ones.

The above analysis changes the bandwidth parameters only along a line in the parameter space. One might wonder about the situation when each average Scott bandwidth value is perturbed by its own factor. For each of the four bandwidth values of the *Radial* descriptor, we have sampled 100 uniformly distributed values within the flat performance interval $[0.9h_i, 1.3h_i]$. Using the obtained 100 bandwidth four-tuples, we have recomputed 100 descriptor sets and tested their performance. The average DCG statistic over these 100

descriptor sets has been $56.8 \pm 0.1\%$, the maximum and the minimum values being 57% and 56.5%. We deduce that, since the DCG performance persists with very low variation, the Scott bandwidth averaged at database-level constitutes a viable bandwidth selection option.

4.2 Robustness Results

4.2.1 Effect of Feature Calculation

The feature observations can be obtained from each mesh triangle, vertex or using the Simpson averaging scheme as presented in Section 2.1.2. In this section, we experiment with these feature calculation options. For this analysis, we have evaluated the retrieval performance of the *Radial* feature on PSB Training Set and SCU. As mentioned in Section 1.2, PSB meshes contain, in general, a smaller number of triangles of arbitrary size and shape, in contrast to highly detailed and regular SCU meshes. In Table 4.3, we provide the DCG performance of the *Radial* descriptor for each of the feature calculation schemes. These results show that Simpson averaging is more effective than the remaining vertex or centroid-based options for PSB, while for SCU, all schemes lead to similar performance. Consequently, we recommend Simpson averaging especially for coarse meshes.

Table 4.3: DCG (%) of *Radial* descriptor using Different Feature Calculation Schemes on PSB Training Set and SCU

	Databases	
Feature Calculation	PSB Training	SCU
Vertex	56.0	71.3
Centroid	55.6	71.2
Simpson	57.0	71.3

4.2.2 Robustness against Low Mesh Resolution

3D models of a certain shape may occur in different levels of detail, as in Figure 4.3, where we display meshes representing a unit sphere at various resolutions, from fine to coarse. Although after a certain level, the mesh might become semantically different from its original version, mesh resolution should not be taken as an intrinsic shape property. Accordingly, we expect that a shape descriptor exhibits small variations against changes in the level of detail. To analyze the behavior of the density-based descriptors under such circumstances, we have taken the highest resolution sphere (with 65024 faces) shown in Figure 4.3 and simplified it successively using the quadric edge collapse decimation procedure built in the MeshLab software [97]. Each decimation halves the number of faces. After simplifying the meshes successively, we have calculated the pdf-based descriptors corresponding to *Radial Distance* and *Normal Direction* features. Figure 4.4 shows the L^1 descriptor variation (with respect to the original mesh) as a function of decreasing resolution for both of the features. For the *Radial Distance*, we have considered three different choices for the scalar bandwidth parameter $h = 0.01$, $h = 0.1$ and $h = 0.5$. For the *Normal* feature, we have taken bandwidth matrices determined by a single scalar parameter of the form $H = hI$ again with the above choices for h (I is the 3×3 identity

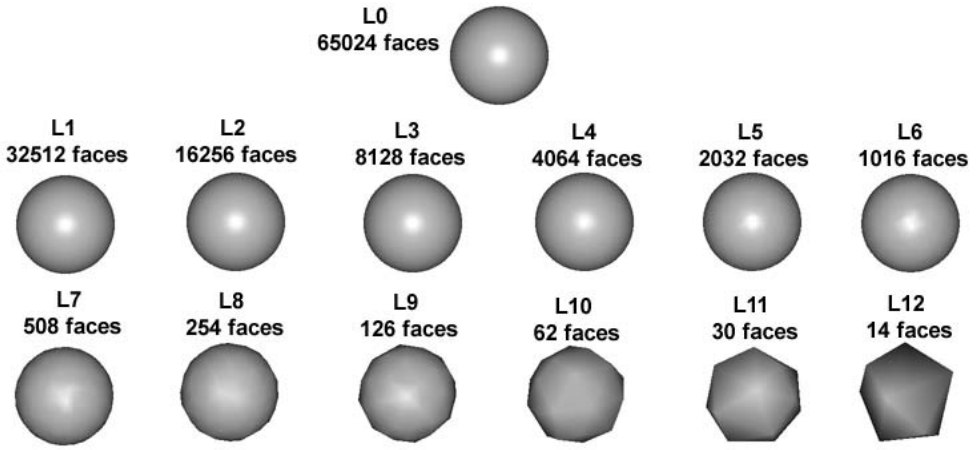

Figure 4.3: A 3D sphere at decreasing levels of mesh resolution

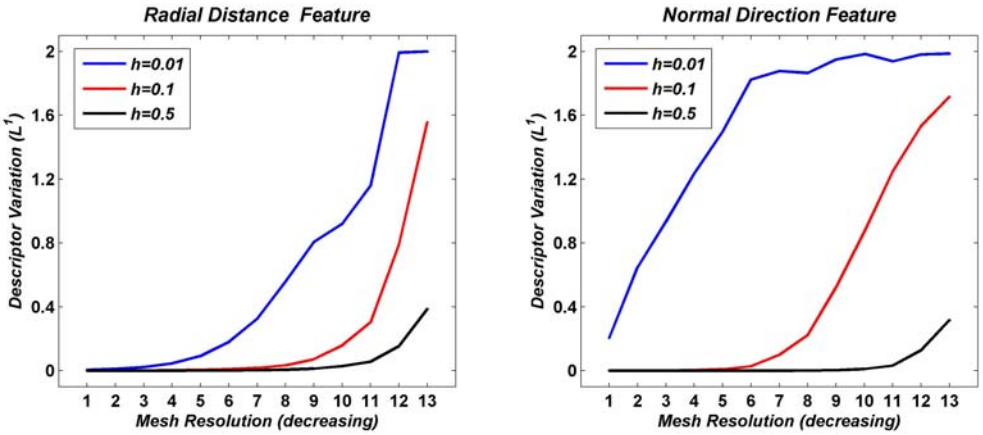

Figure 4.4: L^1 descriptor variation as a function of decreasing mesh resolution for three different smoothing options

matrix). Note that the exact values of h are designated arbitrarily, as what matters here is rather their relative magnitudes, which correspond to different levels of descriptor regularization: a small amount of smoothing ($h = 0.01$), a moderate amount of smoothing ($h = 0.1$) and a large amount of smoothing ($h = 0.5$). We interpret the descriptor variation curves in Figure 4.4 as follows:

- $h = 0.01$: We see that *Radial Distance* descriptors exhibit noticeable variations after only 4 levels of mesh simplification. A small h yields under-smoothed pdf estimates, which rely too much on the available feature observations. Thus, descriptors become too specific to the mesh upon which they are estimated and are affected by spurious resolution information as if it were an intrinsic shape property. *Normal*-descriptors are even more sensitive to the change in level of detail. Clearly, a small of amount of smoothing is not sufficient to discard spurious variations between descriptors.

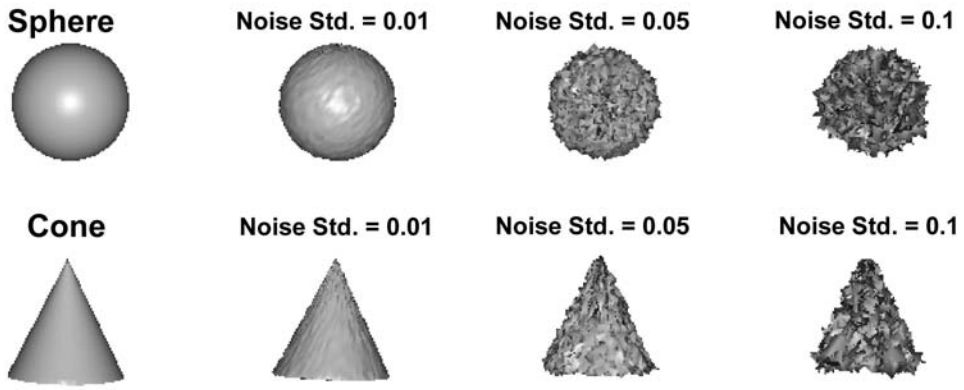

Figure 4.5: Sphere and cone models contaminated with various amounts of isotropic Gaussian noise

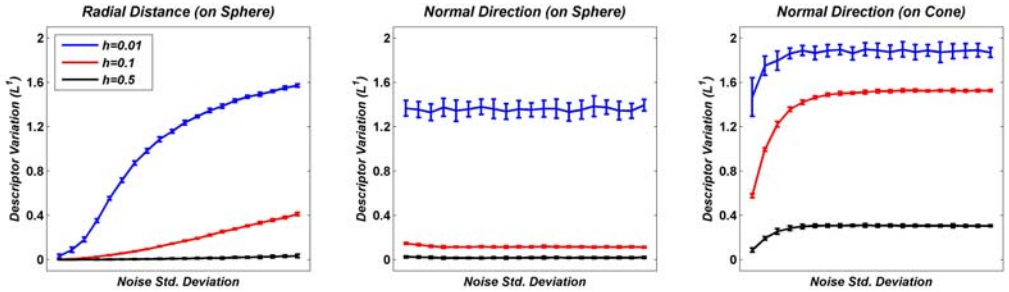

Figure 4.6: L^1 descriptor variation as a function of increasing noise level for three different smoothing options (the bar at each data point indicates standard deviation across different realizations of the additive noise)

- $h = 0.1$: Descriptors start to exhibit variations after 9 levels for the *Radial Distance* and after 7 levels for the *Normal*. Interestingly, these are also the first levels at which we can visually notice that the sphere is slightly altered.
- $h = 0.5$: Descriptor variation persists at negligible values until 11 levels, after which the object looks more like a polyhedron, which is less plausible to be in the same shape class as the sphere. In conclusion, this over-smoothing option is suitable whenever objects are represented by low resolution meshes.

4.2.3 Robustness against Noise

Figure 4.5 displays sphere and cone models contaminated with various amounts of isotropic Gaussian noise ($\sigma = 0.01$, 0.05, and 0.1). Observe that even $\sigma = 0.05$ has a destructive effect on the visual appearance of the shape. In Figure 4.6, we provide variation profiles as a function of the noise standard deviation for *Radial Distance* and *Normal* descriptors, again with the same choices for the bandwidth parameter value h. In much the same way as in the previous section, descriptor variations can be rendered negligible by increasing the smoothing level.

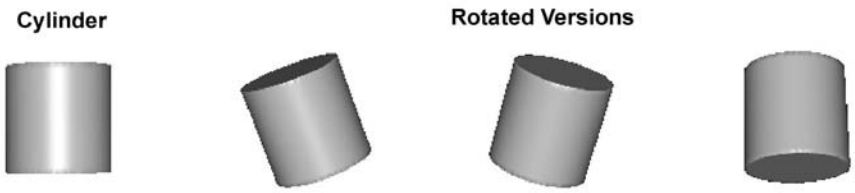

Figure 4.7: A cylinder model and its slightly rotated versions

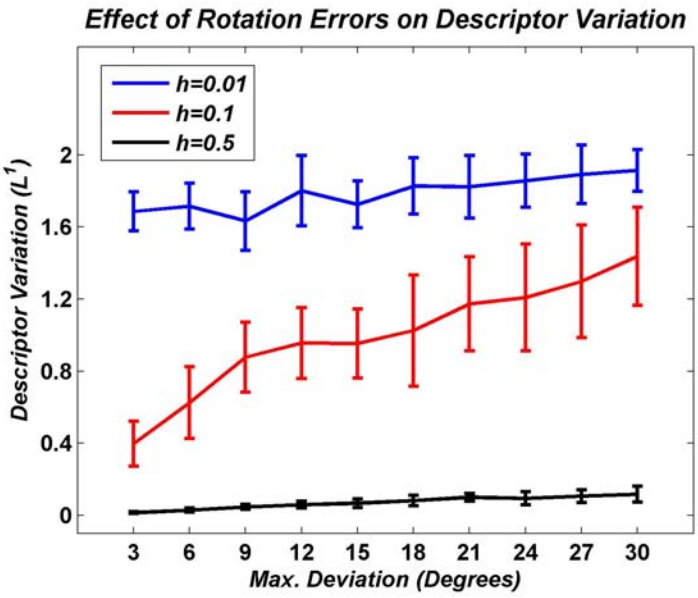

Figure 4.8: L^1 descriptor variation as a function of increasing angular deviation level for three different smoothing options (the bar at each data point indicates standard deviation across different realizations of the angular deviation)

We note that the smoothing effect illustrated by these descriptor variation curves should be interpreted with caution. In the present analysis, we know, in advance, that all mesh occurrences are either a low resolution or a noisy version of an original ideal shape. Consequently, the aim has been to show that the KDE-based scheme is able to make descriptor variations arbitrarily small by increasing the bandwidth parameter. However, *when we are concerned with a discrimination task in a large database*, arbitrarily large smoothing is not a good option as, on one hand, we want small variations between descriptors belonging to the same class, and on the other hand, we want large variations between descriptors belonging to different classes. The issue is to find the proper amount of smoothing for a satisfactory performance, as in the bandwidth selection experiments reported in Section 4.1.

4.2.4 Robustness against Pose Normalization Errors

The smoothing behavior of KDE may become profitable also for small pose deviations, arising due to the imperfections of the PCA-based normalization. We desire that descriptors based on pose-variant features, such as the *Normal*, be robust against perturbations

as in Figure 4.7, which illustrates the case of an up-right cylinder and its slightly deviated versions. To make our point, we have generated randomly rotated versions of a cylinder at increasing levels of angular deviation and evaluated the descriptor variation as in the previous two sections. We have again used $h = 0.01$, 0.1, and 0.5 values for the bandwidth parameter. Variation profiles for the *Normal* descriptor are displayed in Figure 4.8. We see that, even for deviations as large as 30°, it is possible to maintain a small descriptor variation when $h = 0.5$.

4.3 Target Selection

4.3.1 Effect of Sampling Schemes

In Table 4.4, we provide the DCG and NN scores corresponding to all combinations of the sampling schemes presented in Section 2.1.3. The running examples are the *Radial* and the *T-plane* descriptors. First, we would like to remind a subtle issue in target selection, which is closely related to the discretization of the continuous similarity measure as we have mentioned in Section 1.4. For the uniform sampling of the feature domain, the step size factor Δt_n in Eq. 1.1 becomes constant and then irrelevant. When sampling a scalar component, say R, with equal probability intervals (same area under the pdf curve) and/or when sampling a unit-norm vector component, say $\hat{\mathbf{R}}$, with equal spherical coordinate steps, the discretization step size Δt_n should be taken into account in order not to incur into any performance degradation. If this is the case, the performances of equal probability and uniform samplings of the scalar components become virtually equal (within 1%). The same observation also holds for sampling the unit sphere by octahedron subdivision and by spherical coordinates. Table 4.4 shows that all sampling schemes result in equivalent performances. We have also investigated the effect of changing the clipping level α (cf. Section 2.1.3) to determine the range of the scalar components. Our experiments with clipping levels 0.5%, 1%, 2%, and 5% yielded comparable results.

4.3.2 Effect of Descriptor Size

We have also analyzed the effect of changing the descriptor size $N = N_s \times N_v$ between 128 and 8192 under uniform sampling (i.e., uniform scalar set and octahedron subdivision for unit-norm vector set) using the L^1-metric. N_s stands for the size of the scalar set and N_v for the size of the unit-norm vector set. We have tested the combinations of $N_s \in \{4, 8, 16\}$ and $N_v \in \{32, 128, 512\}$ values. Table 4.5 reveals that to maintain adequate DCG performance, N_s should not be less than 8 and that N_v should be at least 128. We also remark that, for the combination $(N_s, N_v) = (8, 128)$, on PSB, preprocessing, feature calculation and density estimation stages take 0.4 second CPU time on the average using a Pentium M 1.86 GHz processor, 1 GB RAM.

4.4 Similarity Measures

The choice of the similarity measure associated with a certain descriptor drastically affects the retrieval performance. In Table 4.6, we provide the DCG scores on various databases using standard metrics defined in the Appendix B, namely, L^p, Kullback-Leibler (KL), Chi-Square (χ^2), and Bhattacharyya (B) distances. Note that, for L^p-distances, descriptors are rescaled to have unit L^p-norm. For KL, χ^2, and B, on the other hand, we have

Table 4.4: DCG (%) and NN (%) Performances using Different Sampling Schemes for Target Selection, scalar set size $N_s = 8$ and unit-norm vector set size $N_v = 128$, **1**: scalar set: *uniform*, unit-norm vector set: *by subdivision*, **2**: scalar set: *uniform*, unit-norm vector set: *by spherical coordinates*, **3**: scalar set: *equal probability*, unit-norm vector set: *by subdivision*, **4**: scalar set: *equal probability*, unit-norm vector set: *by spherical coordinates*

	Radial				*T-plane*			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
DCG	57.0	56.8	56.0	56.3	59.8	60.5	59.5	60.1
NN	57.3	55.9	55.7	55.1	58.7	60.8	59.8	60.5

Table 4.5: DCG (%) for Various Target Set Sizes

	Radial			*T-plane*		
	$N_v = 512$	$N_v = 128$	$N_v = 32$	$N_v = 512$	$N_v = 128$	$N_v = 32$
$N_s = 16$	57.2	56.6	52.2	60.7	60.6	58.3
$N_s = 8$	57.0	57.0	52.0	60.5	59.8	58.1
$N_s = 4$	55.8	55.4	49.9	57.3	57.1	53.4

Table 4.6: DCG (%) Values on PSB and SCU using Standard Similarity Measures

Database	**Descriptor**	$L^{0.6}$	L^1	L^2	L^∞	KL	χ^2	B
PSB Training	*Radial*	57.0	57.0	54.7	44.4	54.4	57.0	56.7
	T-plane	62.0	59.8	55.3	47.1	58.2	61.1	59.4
PSB Test	*Radial*	55.0	54.9	52.0	43.6	53.4	54.7	54.9
	T-plane	58.9	57.8	54.5	47.0	55.8	58.9	58.6
SCU	*Radial*	70.9	71.3	70.7	62.8	70.1	71.3	71.0
	T-plane	72.8	72.0	69.5	63.1	70.1	71.6	71.3

rescaled the descriptors to unit L^1-norm because these measures are defined originally for histograms and/or pdfs. We have observed that for all metrics, appropriate normalization invariably improves the performance. For both the *Radial* and the *T-plane* descriptors, the measures L^1, χ^2, and B yield the best results while L^∞ has the poorest performance. Intrigued by the lower performance of L^2 with respect to L^1, we have also explored the variability of DCG as a function of the p parameter of the L^p-metric within the interval $p \in]0, 2]$. As illustrated in Figure 4.9, the performance degrades significantly for $p > 1$, and for $0 < p \leq 1$, DCG peaks around $p = 0.6$, even outperforming χ^2 and B, the best measures of the previous experiment (see Table 4.7). However, the performance differential is minute. We advocate the use of the L^1-metric due to its computational simplicity and its satisfactory retrieval performance. We remark also that the performance ordering of the considered metrics remains more or less the same for all databases tested as can be seen in Table 4.6.

Another interesting result is the performance improvement gained by using the invariant version of the L^1-metric, introduced in Section 2.4 (see Table 4.7). Recall that this invariant metric can be implemented by taking the minimum distance after holding one descriptor fixed and permuting the entries of the other over a set of possible configurations. The set

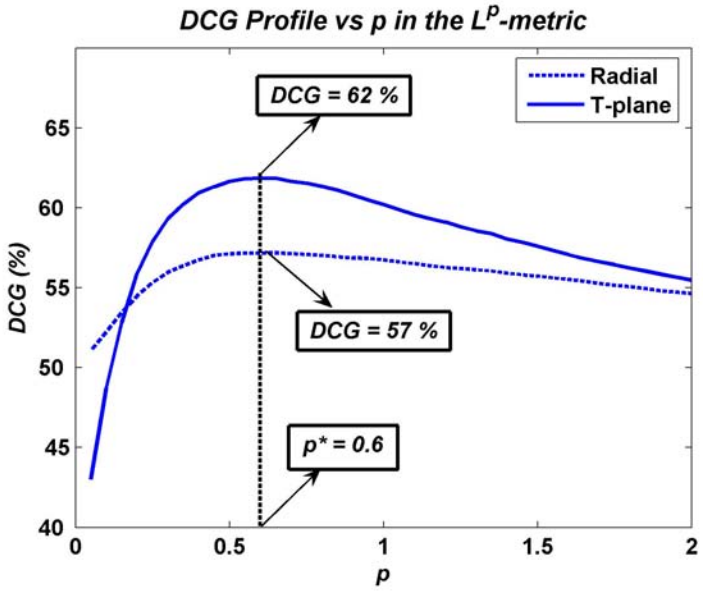

Figure 4.9: DCG Performance vs. the p parameter in the L^p-metric

Table 4.7: DCG (%) of L^1 vs. *Invariant*-L^1 on Different Databases

Database	**Descriptor**	L^1	*Invariant*-L^1
PSB Training	*Radial*	57.0	61.2
	T-plane	59.8	64.9
PSB Test	*Radial*	54.9	57.9
	T-plane	57.8	61.4
SCU	*Radial*	71.3	74.4
	T-plane	72.0	76.2
SHREC-W	*Radial*	74.4	78.0
	T-plane	80.3	82.6
ESB	*Radial*	68.8	70.1
	T-plane	73.7	75.4

of 48 possible permutations are given by coordinate axis relabelings and mirror reflections. Taking the minimum renders the measure invariant to pose normalization deficiencies. From Table 4.7, we observe that, invariably for all databases, this invariant scheme provides significant improvements over the standard L^1-measure at the expense of increased yet affordable computational cost.

4.5 Dimensionality Reduction

We now report the outcome of the dimensionality reduction experiments via dedicated marginalization and probability density pruning tools presented in Section 2.3 as well as the more classical principal component analysis (PCA). In order to quantify our findings, we define a DCG-efficiency measure ϵ as the ratio of DCG after reducing dimensionality

Table 4.8: Retrieval Performance After Marginalization

		Databases					
		PSB Training		PSB Test		SCU	
Desc.	Size	NN	DCG	NN	DCG	NN	DCG
$(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$	1024	57.8	57.0	54.6	54.9	75.0	71.3
$(R, \hat{R}_x, \hat{R}_y)$	512	56.6	57.1	53.5	54.5	74.7	70.5
$(R, \hat{R}_x, \hat{R}_z)$	512	58.4	57.0	52.7	54.3	76.0	71.4
$(R, \hat{R}_y, \hat{R}_z)$	512	55.6	56.2	53.7	54.2	73.9	70.8
$(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$	1024	59.9	59.8	59.1	57.8	74.5	72.0
$(D, \hat{N}_x, \hat{N}_y)$	512	59.2	60.0	55.7	57.3	71.7	69.6
$(D, \hat{N}_x, \hat{N}_z)$	512	60.9	60.5	57.0	58.0	74.3	72.7
$(D, \hat{N}_y, \hat{N}_z)$	512	59.3	59.7	56.7	57.7	73.1	72.0

to the baseline DCG (i.e., the DCG of the "non-reduced" descriptor), concretely, $\epsilon = \text{DCG}_{reduced}/\text{DCG}_{full}$.

4.5.1 Marginalization Results

Table 4.8 summarizes the effect of marginalizing one component from the *Radial* and *T-plane* descriptors. We see that, one of the unit-norm vector components, i.e., $\hat{R}_x, \hat{R}_y$ or $\hat{R}_z$ for the *Radial* descriptor, and $\hat{N}_x, \hat{N}_y$ or $\hat{N}_z$ for the *T-plane* descriptor, can be sacrificed. We do not incur into any significant loss in marginalizing one of these components (in some cases, we even observe an increase in performance) and the descriptor size is halved. This should not be a surprise: as we have pointed out in Section 2.3.1, a directional component is completely determined up to a sign given the other two. It is also worth noting that, with these results at our disposal, we can directly estimate the pdf of the most informative components and reduce the computational overhead beforehand. Note also that the above observations hold invariably for both PSB and SCU.

4.5.2 Probability Density Pruning Results

We experiment with the probability density pruning technique by varying the threshold parameter λ (cf. Section 2.3.2). Clearly, as we increase λ, we can obtain more reduction in descriptor size. The task here is to achieve this at no or little performance loss. From the DCG profiles in Figure 4.10, we observe that, for *Radial* and *Sec-Order* descriptors, a reduction rate at around 35% can be obtained with nearly 100% DCG efficiency. The performance of the *T-plane* descriptor, on the other hand, is much more sensitive to pruning. We want to point out that for density pruning, the reduction in size changes monotonically as a function of the performance loss incurred. Furthermore, insignificant targets can be eliminated off from density estimation stage, reducing the computational overhead beforehand.

4.5.3 PCA Results

A more traditional approach to reduce dimensionality is to project the descriptor vector onto a linear subspace using standard methods such as principal components analysis

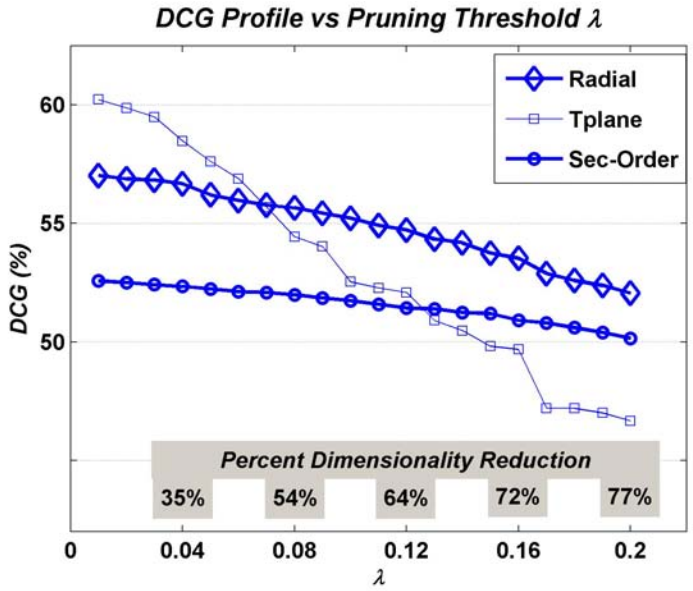

Figure 4.10: DCG profile as a function of the pruning threshold parameter λ on PSB Training Set

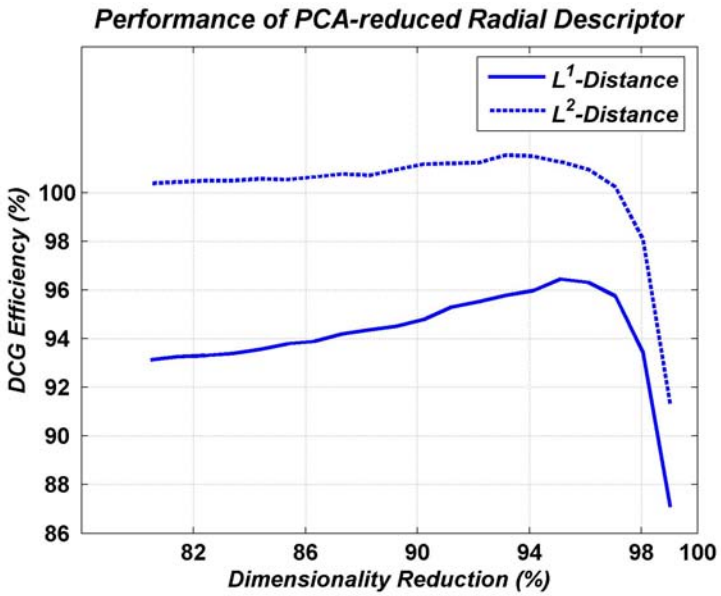

Figure 4.11: DCG efficiency profile of the *Radial* descriptor after PCA-based dimensionality reduction on PSB Training Set

(PCA) or independent component analysis (ICA) [13]. The PCA of our *Radial* descriptor on PSB Training Set reveals that the total variance in the first 270 components reaches 99%, suggesting that significant dimensionality reduction can be achieved. We have experimented by changing the projection dimension from 200 down to 10 (with increments of 10) and performed retrieval experiments using L^1 and L^2-distances. In Figure 4.11, DCG effi-

ciency profiles for both L^1 and L^2-distances are shown. Note that L^1 and L^2 performances of the full descriptor (size = 1024) are 57% and 54.8% respectively in terms of DCG. In Figure 4.11, we see that, for the L^1-distance, although dimensionality reduction at the maximum possible DCG efficiency ($\approx 96\%$) is significant (reduced descriptor size is 50), we can never achieve 100% DCG efficiency . This contrasts the cases of marginalization and probability density pruning where lossless dimensionality reduction is possible. On the other hand in the same figure, we observe that, when the L^2-metric is used, we can improve the DCG with respect to the full descriptor even after dimensionality reduction (efficiency above 100%). This should not be a big surprise as PCA is basically an L^2 method. Unfortunately, the L^2-metric is not our preferred one as it is always inferior to L^1 in terms of DCG performance (cf. Section 4.4). Nevertheless, reduction rates achieved by PCA are quite impressive. In applications where a DCG loss at the order of 2-5% is affordable, PCA may become interesting. We have also experimented with the ICA method but we have not observed any improvement in comparison to PCA.

In conclusion, we recommend the following procedure to reduce the dimensionality of a density-based descriptor:

(i) Use marginalization to discover redundant feature components whose removal do not lead to a performance loss.

(ii) Apply then probability density pruning to the marginalized pdf to reduce the descriptor size further, by adjusting the λ parameter at a rate where no performance loss is incurred.

4.6 Feature-Level Fusion

Feature-level fusion refers to joining simple scalar features to obtain more specific multivariate ones for subsequent density-based descriptor extraction. In principle, as we add more components, the discriminativeness of the multivariate feature increases. Accordingly, it is essential to discover which subset(s) of the whole feature set gives the best performance for a given data set with a certain classification semantics. Recall that, in Section 2.1.1, we have introduced several scalar and multivariate features to locally characterize a 3D surface. In this section, we take the individual components of the multivariate features introduced in Section 2.1.1 as scalars. Afterwards, we pool them with the remaining scalar features to obtain an exhaustive set given by

$$\mathcal{FS} = \left\{ R, \hat{R}_x, \hat{R}_y, \hat{R}_z, D, \hat{N}_x, \hat{N}_y, \hat{N}_z, C_x, C_y, C_z, A, SI \right\}.$$

In the following section, we first test the impact of feature-level combinations by synthesizing a few example subsets from $\mathcal{FS}$. Then, in Section 4.6.2, we invoke the marginalization tool to explore a non-redundant subset of $\mathcal{FS}$ in a complete manner.

4.6.1 A Few Examples

Obviously, the KDE scheme can be applied to any joining of the 13 features in $\mathcal{FS}$. The whole space of combinations contains $\sum_{i=1}^{13} \binom{13}{i} = 8191$ possibilities. However, estimation accuracy and computational/memory constraints limit the maximum number of features that can be joined. In fact, for dimensions exceeding five, even with a sparse sampling of the multidimensional feature range, we obtain an impractical descriptor size of the order

of 10^4. Hence, we should limit ourselves to three-, four-, or five-tuple feature combinations in which case the cardinality of the combination space reduces to $\binom{13}{3}+\binom{13}{4}+\binom{13}{5}=2288$. Even without an exhaustive search, a small subset of judiciously chosen combinations can indicate the effect of feature-level fusion in retrieval performance.

In Table 4.9, we give the results of a small subset of 13 combinations, where the shaded cells in any row show the features taking role in that specific combination. Consider, for example, row **1**: the shaded cells pick the *Radial Distance* R and the *Radial Direction* $\hat{\mathbf{R}}$ so that this row corresponds to the $(R,\hat{R}_x,\hat{R}_y,\hat{R}_z)$-combination. On the rightmost columns, we provide the size of the multivariate density descriptor, the NN and DCG scores respectively for the three data sets (PSB Training, PSB Test and SCU). Recall that the size of a descriptor equals the number of density evaluation points, i.e., the targets (see Section 2.1.3). In the experiments, as a general rule, for each scalar feature we have chosen 8 equally spaced points within its domain (9 for the shape index SI, in order to cover the shape primitives such as spherical cap and cup, ridge, trough, saddle, etc., see [71, 57] for details). For unit-norm directional features such as *Radial Direction* $\hat{\mathbf{R}}$ or *Normal* $\hat{\mathbf{N}}$, we have sampled 128 points on the unit sphere (by subdividing twice the triangles of an octahedron into four triangles). When a directional feature occurs without its third coordinate, as for example $(\hat{R}_x,\hat{R}_y)$, we choose 64 points. An exception to this case is $(R,\hat{R}_x,\hat{R}_y,\hat{N}_x,\hat{N}_y)$ where we have chosen 16 points for both $(\hat{R}_x,\hat{R}_y)$ and $(\hat{N}_x,\hat{N}_y)$ pairs to maintain a reasonable descriptor size. Finally, for the *Torque* feature $\mathbf{C}$, we have sampled 320 points within the unit sphere.

We can infer the following based on Table 4.9:

- For the three datasets, the $(R,\hat{R}_x,\hat{R}_y,\hat{N}_x,\hat{N}_y)$ (row **4** in Table 4.9) and $(\hat{R}_z,D,\hat{N}_x,\hat{N}_y)$ (row **10**) configurations, of sizes 2048 and 4096 respectively, are the two descriptors with top performance.

- The better performance of $(\hat{R}_z,D,\hat{N}_x,\hat{N}_y)$ against other 4-tuple combinations suggests that mixing pieces of radial and normal information results in a better descriptor than any other using solely the radial information $(R,\hat{R}_x,\hat{R}_y,\hat{R}_z)$ or the tangent plane information $(D,\hat{N}_x,\hat{N}_y,\hat{N}_z)$.

- Adding the alignment information A into a configuration significantly increases the discrimination performance, at the cost of increased descriptor size. For instance on PSB Training Set, the $(R,\hat{R}_x,\hat{R}_y,\hat{R}_z)$-configuration has a DCG performance of 57% (row **1**), while its augmented version (row **2**) has a DCG of 61.7%.

- The (R,A,SI)-descriptor, which radializes the alignment information coupled with the shape index, has the poorest performance with respect to the remainder. However, the performance differential is less important for SCU than for PSB. We can explain this fact by the high resolution of SCU meshes in comparison to PSB, which makes SI computation more reliable for SCU meshes.

Table 4.9: Retrieval Performance for Feature-level Combinations (gray cells denote the selected features for the 13 descriptors)

Multivariate Density Descriptor	Features													Databases					
														PSB Train.		PSB Test		Sculpteur	
	R	$\hat{R}_x$	$\hat{R}_y$	$\hat{R}_z$	D	$\hat{N}_x$	$\hat{N}_y$	$\hat{N}_z$	C_x	C_y	C_z	A	SI	Size	NN (%)	DCG (%)	NN (%)	DCG (%)	NN (%)	DCG (%)
1														1024	57.8	57.0	54.6	54.9	75	71.3
2														8192	63.9	61.7	62.1	59.7	76.6	74.0
3														4096	60.9	60.0	58.7	57.1	76.2	73.5
4														2048	62.4	61.0	61.6	60.1	78.8	75.4
5														4096	58.4	57.6	57.9	57.2	77.6	73.4
6														1024	59.9	59.8	59.1	57.8	74.5	72.0
7														8192	63.5	61.3	59.0	58.1	77.8	74.0
8														4096	61.0	60.5	55.0	57.6	74.3	72.9
9														4096	60.8	61.9	58.7	59.5	75.8	74.9
10														4096	64.1	62.2	60.1	59.7	76.4	72.8
11														2560	51.0	55.6	51.0	54.8	73.5	73.4
12														4096	61.4	60.5	56.8	57.8	78.2	74.7
13														576	42.9	52.6	40.5	49.1	69.8	71.0

Table 4.10: Performance of the Three Starting Feature Candidates for Marginalization Experiments on PSB Training Set

Descriptor	Size	NN	DCG
$(R, \hat{R}_x, \hat{R}_y, \hat{R}_z, \hat{N}_x, \hat{N}_y, \hat{N}_z, SI)$	73728	53.4	54.4
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, SI)$	10240	57.3	58.4
$(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$	10240	63.1	62.1

4.6.2 Marginalization Revisited

The aim of the previous section has been to explore performance variation among moderately sized descriptors, which can be computed fast enough for a real-time application. In the present section, putting such computational and memory constraints aside, we carry out a performance-driven combinatorial analysis of the feature set $\mathcal{FS}$. We make use of marginalization to explore a non-redundant subset of $\mathcal{FS}$.

In Section 2.3.1, marginalization has been introduced as a tool to obtain a family of nested descriptors, starting from the discretized pdf of an exhaustive feature combination. An obvious candidate for this multivariate feature is obtained by joining all the members of the set $\mathcal{FS}$ into a 13-dimensional vector. However, we will not start with this huge feature vector due to two reasons: (i) KDE in 13-dimensional space is just infeasible, (ii) the set $\mathcal{FS}$ contains redundant members in the sense that some of them can be computationally derived from the others. Consequently, in a first attempt, we will evaluate the performance of the following multivariate features:

- $(R, \hat{\mathbf{R}}, \hat{\mathbf{N}}, SI) = (R, \hat{R}_x, \hat{R}_y, \hat{R}_z, \hat{N}_x, \hat{N}_y, \hat{N}_z, SI)$. Recall that our construction in Section 2.1.1 synthesized this feature, which combines all the available local surface information up to second-order. It makes use of solely primary features (cf. Table 2.1), so it is non-redundant. One important problem associated with this feature is its dimensionality: beyond six dimensions, we can no longer be sure about the accuracy of KDE due to the curse of dimensionality [7, 13]. Another problem is the associated descriptor size.

- $(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, SI)$. This feature is a subset of the above one, in which case we have removed one component from each of the *Radial Direction* $\hat{\mathbf{R}} = (\hat{R}_x, \hat{R}_y, \hat{R}_z)$ and the *Normal* $\hat{\mathbf{N}} = (\hat{N}_x, \hat{N}_y, \hat{N}_z)$ features. Marginalization results in Section 4.5.1 have demonstrated that one such component can be safely removed, without risking a performance loss in practice, due to unit-norm condition on $\hat{\mathbf{R}}$ and $\hat{\mathbf{N}}$. Here, we have arbitrarily chosen to delete the z-component.

- $(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$. This feature is similar to the previous one except the SI feature, which has been replaced by the alignment A. We have decided to consider this third candidate to start the marginalization analysis by observing the poor performance of the primary second-order SI feature against the auxiliary first-order radial-normal alignment A. Furthermore, the alignment A has the additional effect of restoring the information lost due to the removal of $\hat{R}_z$ and $\hat{N}_z$ components.

In Table 4.10, we display the DCG performance of the above descriptors on PSB Training Set and their associated descriptor sizes. We can make the following comments:

- We see that the exhaustive feature $(R, \hat{\mathbf{R}}, \hat{\mathbf{N}}, SI)$ exhibits the poorest performance among the three options tested. This a priori unexpected result can be explained mainly by the potential inaccuracy of KDE in dimensions beyond six [7].
- Another possible reason of the poor performance of $(R, \hat{\mathbf{R}}, \hat{\mathbf{N}}, SI)$ might be the fact that such a specific feature induces an overly specific descriptor so that very individual shape details are preserved, preventing common shapes to cluster in the descriptor space.
- Comparing the performances of the two remaining two features, we uncover a fact that we could have intuitively anticipated: $(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$ performs better than $(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, SI)$, as the alignment A has proved to be very successful in boosting the performance of a lower-dimensional combination (cf. previous section).

Once we have determined a good starting point for subsequent feature space exploration, we can proceed with combinatorial performance assessment using marginalization. We emphasize that this analysis is exact within pdf estimation accuracy and does not involve any heuristics such as sequential floating search [98, 86]. As the full $(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$-descriptor is available by the above analysis, we can remove one component at a time to evaluate the performance of all $2^6 - 2 = 62$ remaining subsets of this 6-component feature, without explicit descriptor computation. The DCG values of all the 62 sets of descriptors are tabulated in the Appendix D. The best performing descriptors at each stage of marginalization are tabulated in Table 4.11. In this table, we also display the performance of concatenated marginal pdfs to further illustrate the benefits of feature fusion. A few comments follow:

- The components are removed in the following order (from left to right):

$$\hat{N}_y, \hat{R}_y, R, A, \hat{R}_x.$$

- If we interpret these results as an indication about the informativeness and/or discriminativeness of individual components, we have the following order relation:

$$\hat{N}_x \succ \hat{R}_x \succ A \succ R \succ \hat{R}_y \succ \hat{N}_y.$$

- A thorough investigation of the tables in Appendix D reveals the following interesting fact: if we were to resort to a heuristic telling us to remove the component whose exclusion gives the greatest performance loss at each stage, we would have obtained the same sets of best descriptors depicted in Table 4.11. That is, after a feature is removed at a certain stage, no descriptor instance using that feature reappears as the best performing one at later stages. This suggests that the above discriminativeness ordering is reliable.
- At each stage, the performance of joined features is superior to their concatenated marginal pdfs, putting again into evidence the effectiveness of feature-level fusion.

4.7 Basic Score Fusion

There is a limit for feature-level combinations with a dimension greater than five: (i) the quality of density estimation degrades due to the curse of dimensionality [13], and (ii)

Table 4.11: Best Performing Descriptors at Each Stage of Marginalization

Components		Joint-PDF		Concat. Marginals	
Retained	**Removed**	**DCG**	**Size**	**DCG**	**Size**
$R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A$	-	62.1	10240	57.7	45
$R, \hat{R}_x, \hat{R}_y, \hat{N}_x, A$	$\hat{N}_y$	62.6	5120	57.1	37
$R, \hat{R}_x, \hat{N}_x, A$	$\hat{R}_y$	63.4	2560	57.1	29
$\hat{R}_x, \hat{N}_x, A$	R	61.5	320	54.8	21
$\hat{R}_x, \hat{N}_x$	A	58.1	64	51.7	16
$\hat{N}_x$	$\hat{R}_x$	44.8	8	44.8	8

descriptors become prohibitively large and cannot be computed fast enough for on-line applications. Score fusion at descriptor-level provides a working alternative to extract most of the information brought by different features. However, in its basic form, it cannot make use of feature correlations. Accordingly, it constitutes a "suboptimal" heuristic way to fuse shape similarity information.

In basic score fusion, we simply sum the L^1-distances between descriptors computed using various feature combinations considered in Section 4.6.1. Note that summing the L^1-distances coming from several descriptors can be effected by concatenating the involved descriptors to obtain a larger one and then calculating a single distance. In this view, basic L^1 score fusion is the same as descriptor-level information fusion.

In the following experiments, we consider pairwise and triple concatenations of the 13 descriptors (each indicated by its boldface number) presented in Section 4.6.1. The performances of various descriptor combinations are visualized in Figure 4.12 (pairwise) and Figure 4.13 (triple). In Figure 4.12, the horizontal axis stands for descriptors **1**, **2**, **3**, and **5** of Table 4.9. Each bar series illustrates DCG performance improvements after pairwise combination with descriptors **6**, **7**, **8**, **9**, **10**, **11**, and **13** (displayed on the top of the bar) one at a time. For instance, at descriptor **1**, the bar with the index **6** on the top corresponds to the pairwise combination **1**+**6**, i.e., the concatenation of the $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$-descriptor with $(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$-descriptor. The gray bars indicate the maximum of DCG performances of the individual descriptors involved in that specific combination, e.g., $max(\text{DCG}_1, \text{DCG}_6)$. The white portion of the bar on top of the gray one shows the DCG improvement due to the combination. Figure 4.13 illustrates these gains for triple combinations, that is, descriptor vectors resulting from three concatenated descriptors. Accordingly, in Figure 4.13, the horizontal axis has labels of pairwise combinations, such as **1**+**6**, **1**+**7**,..., **5**+**9**, **5**+**10**. The darker portion of the bar again indicates the maximum individual performances of descriptors before combination, and the light portion indicates the corresponding DCG improvement after triple combination. In these experiments, the third descriptor is always either the (R, C_x, C_y, C_z)-descriptor (**11**) or (R, A, SI)-descriptor (**13**), as they bring in different shape information than the pairwise combinations displayed at the horizontal axis of Figure 4.13.

We have intuitively chosen the combinations so that their corresponding feature sets are as disjoint as possible. For instance, we did not consider concatenating the pdf of $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ (descriptor **1**) with that of $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z, A)$ (descriptor **2**) as the latter already contains the information brought by the former. We would like to underline the following points regarding the results of basic score fusion experiments:

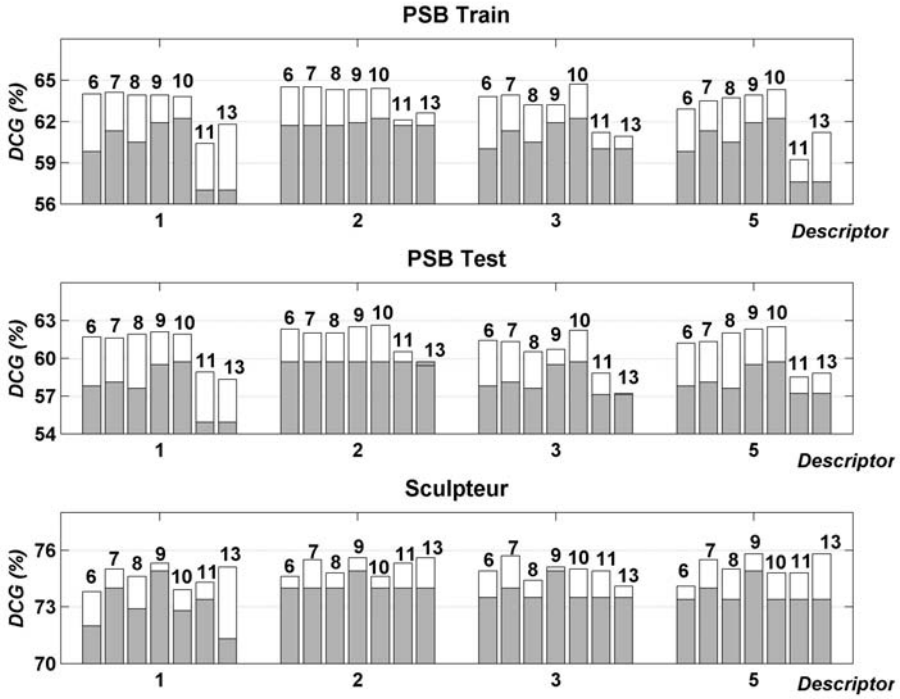

Figure 4.12: DCG improvements with pairwise descriptor combinations. Gray portion of the bar: maximum DCG before combination; white portion of the bar: DCG increase after combination.

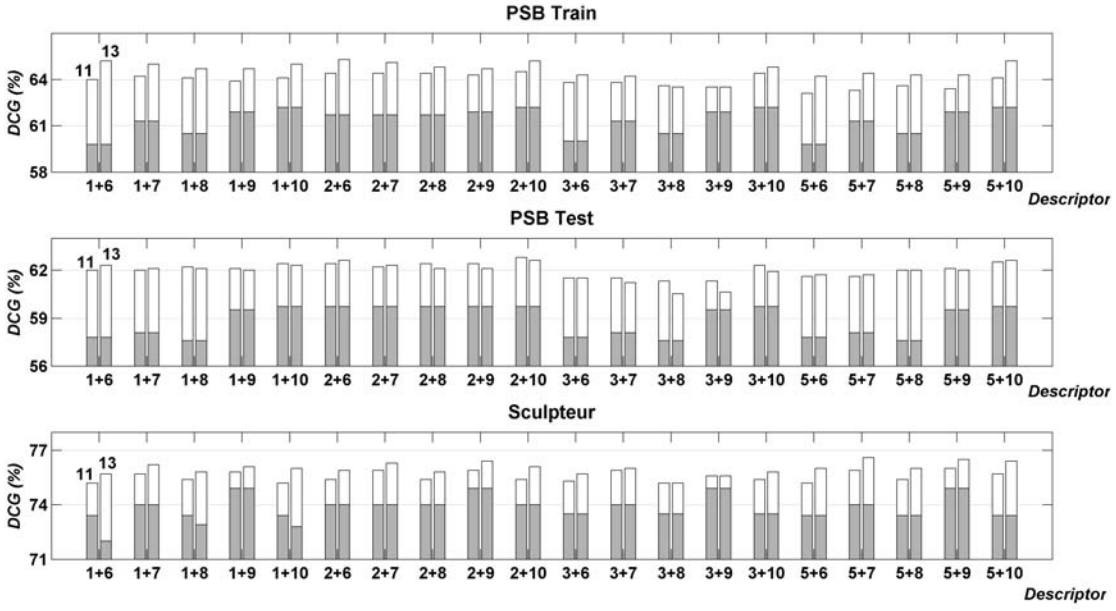

Figure 4.13: DCG improvements with triple descriptor combinations. Gray portion of the bar: maximum DCG before combination; white portion of the bar: DCG increase after combination.

- As expected, descriptor combinations become effective only when the involved features bring in a different kind of shape information. A case in point involves $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ (descriptor **1** in Table 4.9) and $(D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ (descriptor **6** in Table 4.9). For PSB Training Set, individual DCGs of these descriptors are 57% and 59.8% respectively (see Table 4.9). After the combination, the DCG performance boosts to 64% achieving a DCG improvement of 7% for PSB Training Set (6.7% for PSB Test Set and 2.5% for SCU), as shown in Figure 4.12. Furthermore, in this case, the descriptor size is 2048 (2×1024) and quite reasonable in comparison to other options.

- When we consider the triple combinations, in the case where (R, A, SI)-descriptor (descriptor **13** in Table 4.9) is concatenated to the previous descriptor combination **1**+**6**, DCG score has a further but smaller jump to 65.2% for PSB Training Set, to 62.3% for PSB Test Set, and to 75.7% for SCU (Figure 4.13). Presently, this combination constitutes the most interesting option as it has the smallest size among other triple combinations tested ($1024 + 1024 + 576 = 2624$ in total).

- These experiments demonstrate the potential of score fusion to improve retrieval performance, even in its basic form where no statistical learning is involved. This fact provides further motivation for statistical similarity learning. We deliver the results of statistical learning-based score fusion in the forthcoming Section 4.10.

4.8 Comparison to Other Methods

In this section, we illustrate the effectiveness of our shape description scheme in comparison to other state-of-the-art methods. First in Section 4.8.1, we compare certain instances of the density-based framework to their histogram-based peers. Then in Section 4.8.2, we provide a more general comparison.

4.8.1 Comparison with Histogram-Based Peers

One of the motivations of the present work is to show that a considerable improvement in the retrieval performance can be obtained by more rigorous and accurate computation of local feature distributions as compared to more practical ad-hoc histogram approaches. The term “histogram-based descriptor" stands for any count-and-accumulate type of procedure. Accordingly, in terms of the local surface features employed, many shape descriptors in the literature [35, 36, 32, 30, 87] correspond to an instance of the density-based framework. Our descriptors differ from their peers in one or more of the following aspects:

- **Feature Calculation.** Features are evaluated using the Simpson averaging scheme presented in Section 2.1.2. This provides smoother, hence more reliable observations for subsequent density or histogram estimation.

- **Multivariate Features.** By joining scalar features to obtain multivariate ones, we render the resulting descriptor more effective. This fact has already been demonstrated in Section 4.6.2. Here, we provide additional evidence for this claim.

- **KDE.** The most important distinction between KDE and histogram approach is that the former leads to smoother estimates of the pdf by judiciously chosen bandwidth parameters. KDE-based descriptors emphasize common intrinsic aspects of

the shapes in a better way than histogram-based ones, and eliminate extrinsic effects such as differing mesh resolutions and small pose deviations (cf. Section 4.2).

Radial vs. Cord and Angle Histograms

An interesting case in point is cord and angle histograms (*CAH*) [30]. The features in *CAH* are identical to the individual scalar components $R, \hat{R}_x, \hat{R}_y$, and $\hat{R}_z$ of our *Radial* feature up to a parameterization. In [30], the authors consider the length of a cord (corresponding to R) and the two angles between a cord and the first two principal directions (corresponding to $\hat{R}_x$ and $\hat{R}_y$). Notice that in our parameterization of $\hat{\mathbf{R}}$, we consider the Cartesian coordinates rather than the angles. In order to compare *CAH* with our *Radial* descriptor, we have implemented it by also including the histogram of the angle with the third principal direction. The resulting *CAH* descriptor is thus the concatenation of one cord length and three angle histograms. Each histogram consisting of 64 bins leads to a descriptor of size $N = 4 \times 64 = 256$. We have compared our approach to *CAH* in two levels. First, likewise to *CAH*, we have estimated only scalar pdfs of the individual components of our *Radial* feature and concatenated them. In Table 4.12, we denote this particular form of the descriptor by *Radial*-CM to remind that it consists of concatenated marginal pdfs. Then, to illustrate the true power of our scheme, we have used the joint pdf of the multivariate *Radial* feature $(R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$. We denote this instance of the descriptor by *Radial*-J.

Radial-CM differs from *CAH* in three aspects: first, it uses a different parameterization of the angle (direction) components; second, the local feature values are calculated by Eq. 2.2 instead of using mere barycentric sampling; third, it employs KDE instead of histogram. In Table 4.12, we provide the retrieval statistics of *CAH*, *Radial*-CM and *Radial*-J descriptors on PSB and SCU. In Figures 4.14, 4.15 and 4.16 we display the corresponding precision-recall curves. Observe that for all databases, the performance of *Radial*-CM is clearly superior to its histogram-based equivalent *CAH*. *Radial*-J improves the results even further.

Note that the performance improvement using our scheme is less impressive over SCU than over PSB. This can be explained by the fact that SCU meshes are much denser than PSB meshes in the number of triangles. As the number of observations K increases, the accuracies of the histogram method and KDE become comparable and both methods result in similar descriptors. This also indicates that the KDE methodology is especially appropriate for coarser mesh resolutions as in PSB.

Normal vs. Extended Gaussian Image

A second instance of our framework outperforming its competitor is the case of the *EGI* descriptor [35, 23, 3], which can be obtained by binning the surface normals. The density of the *Normal* feature $\hat{\mathbf{N}}$ is similar to *EGI*. There can be different implementation choices for binning surface normals, e.g., by mapping the normal of a certain mesh triangle to the closest bin over the unit sphere and augmenting that bin by the relative area of the triangle. In the present work, similar to [87], we preferred the following implementation for the *EGI* descriptor. First, 128 unit norm vectors $\hat{\mathbf{n}}_{bin,j}, j = 1, \ldots, 128$ are obtained as histogram bin centers by octahedron subdivision, as described in Section 2.1.3. Then, the contribution of each triangle $T_k, k = 1, \ldots, K$ with normal vector $\hat{\mathbf{n}}_k$ to the nth bin center is computed as $w_k \, |\langle \hat{\mathbf{n}}_k, \hat{\mathbf{n}}_{bin,j} \rangle|$ if $|\langle \hat{\mathbf{n}}_k, \hat{\mathbf{n}}_{bin,j} \rangle| \geq 0.7$ or otherwise as zero (recall that w_k is the relative area of the kth triangle). The *Normal* descriptor of the same size, i.e., 128, achieves a superior DCG of 51% as compared to the DCG score of 43.8% for *EGI*

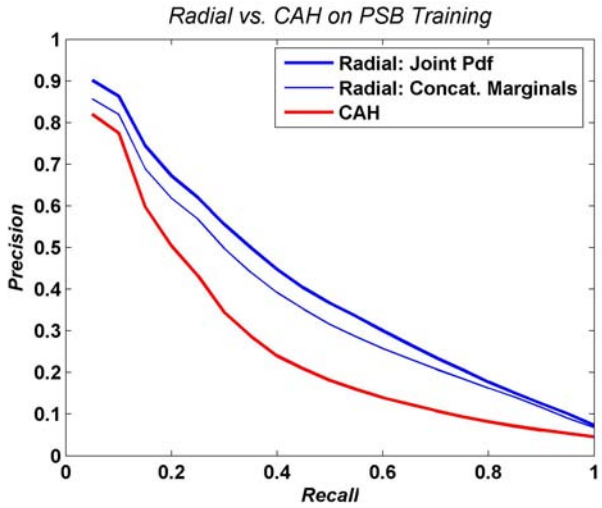

Figure 4.14: Precision-recall curve: *Radial* vs. *CAH* on PSB Training Set

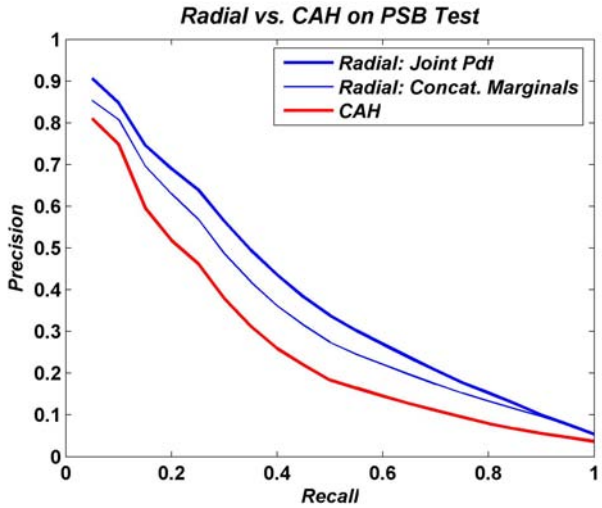

Figure 4.15: Precision-recall curve: *Radial* vs. *CAH* on PSB Test Set

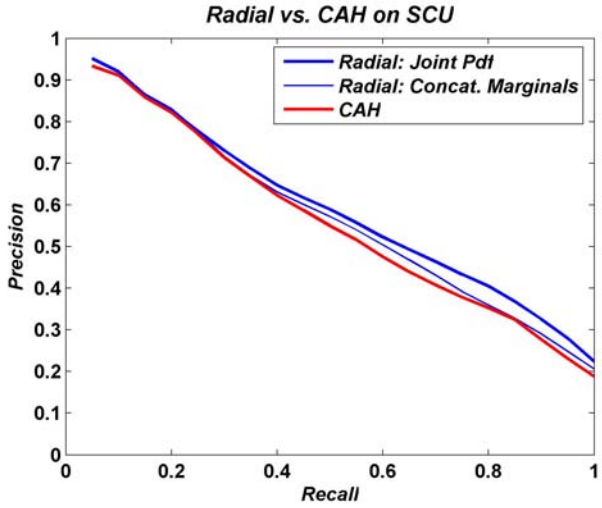

Figure 4.16: Precision-recall curve: *Radial* vs. *CAH* on SCU

Table 4.12: Retrieval Statistics (%): Density-Based Descriptors vs. Their Histogram-Based Peers

Database	Descriptor	NN	FT	ST	E	DCG
PSB Training	*Radial*-J	57.8	28.8	37.8	21.5	57.0
	Radial-CM	48.5	24.7	34.7	20.1	53.4
	CAH	36.4	15.7	22.5	13.0	43.8
	Normal	50.3	24.7	33.9	19.1	53.1
	EGI	33.8	16.1	23.8	14.2	44.3
	T-plane	59.9	31.9	42.8	24.3	59.8
	3DHT	60.0	31.4	40.6	23.2	58.7
PSB Test	*Radial*-J	54.6	26.9	35.2	20.7	54.9
	Radial-M	45.6	22.5	31.9	18.9	51
	CAH	33.2	15.9	22.9	13.7	43.3
	Normal	45.1	23.4	31.9	18.7	51
	EGI	31.1	16.5	24.5	14.5	43.8
	T-plane	59.1	30.3	39.9	22.9	57.8
	3DHT	58.8	31.1	39.6	23.0	57.7
SCU	*Radial*-J	75.0	45.8	57.2	32.3	71.3
	Radial-M	73.1	42.7	55.6	31.6	70.0
	CAH	67.8	42.7	53.6	30.9	68.1
	Normal	64.1	37.2	47.4	26.5	63.4
	EGI	48.9	25.2	34.9	20.3	53.5
	T-plane	74.5	48.4	58.5	33.0	72.0
	3DHT	77.8	48.5	60.3	33.6	72.7

on PSB Test Set (see Table 4.12). For SCU, the DCG performance differential is even more pronounced (DCG = 63.4% for the *Normal*, DCG = 53.5% for *EGI*). Precision-recall curves in Figures 4.17, 4.18 and 4.19 corroborate these facts.

T-plane vs. 3D Hough Transform

A third instance of comparison can be considered between our *T-plane* descriptor and the *3DHT* descriptor [87] since both of them use local tangent plane parameterization. The procedure to obtain the *3DHT* descriptor is carried out as follows. We first recall that *3DHT* is a histogram constructed by accumulating mesh surface points over planes in 3D space. Each histogram bin corresponds to a plane $\mathcal{P}_{ij}$ parameterized by its normal distance $d_i, i = 1, \ldots, N_{\mathcal{I}}$ to the origin and its normal direction $\hat{\mathbf{n}}_{bin,j}, j = 1, \ldots, N_{\mathcal{S}^2}$. Clearly, there can be $N_{\mathcal{I}} \times N_{\mathcal{S}^2}$ such planes and the resulting descriptor is of size $N = N_{\mathcal{I}} \times N_{\mathcal{S}^2}$. We can obtain such a family of planes exactly as described in Section 2.1.3. In our experiments, we have used $N_{\mathcal{I}} = 8$ distance bins sampled within the range $[0, 2]$ and $N_{\mathcal{S}^2} = 128$ uniformly sampled normal directions. This results in a *3DHT* descriptor of size $N = 1024$. To construct the Hough array, one first takes a plane with normal direction $\hat{\mathbf{n}}_{bin,j}, j = 1, \ldots, N_{\mathcal{S}^2}$ at each triangle barycenter $\mathbf{m}_k, k = 1, \ldots, K$ and then calculates the normal distance of the plane to the origin by $|\langle \mathbf{m}_k, \hat{\mathbf{n}}_{bin,j} \rangle|$. The resulting value is quantized to the closest $d_i, i = 1, \ldots, N_{\mathcal{I}}$ and then the bin corresponding to the plane $\mathcal{P}_{ij}$ is augmented by $w_k \, |\langle \hat{\mathbf{n}}_k, \hat{\mathbf{n}}_{bin,j} \rangle|$ if $|\langle \hat{\mathbf{n}}_k, \hat{\mathbf{n}}_{bin,j} \rangle| \geq 0.7$ (the value of 0.7 is suggested by

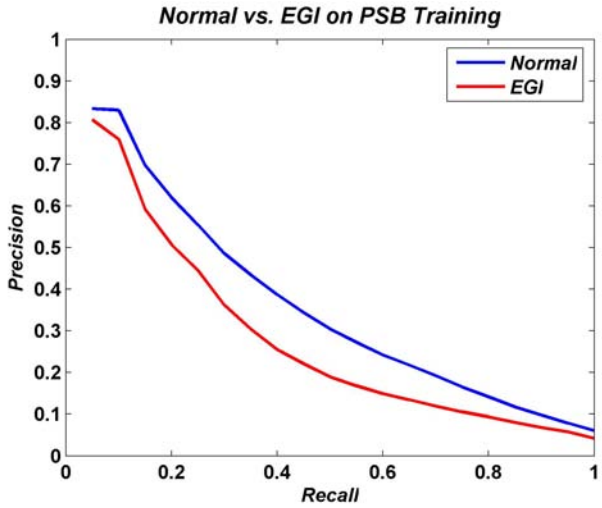

Figure 4.17: Precision-recall curve: *Normal* vs. *EGI* on PSB Training Set

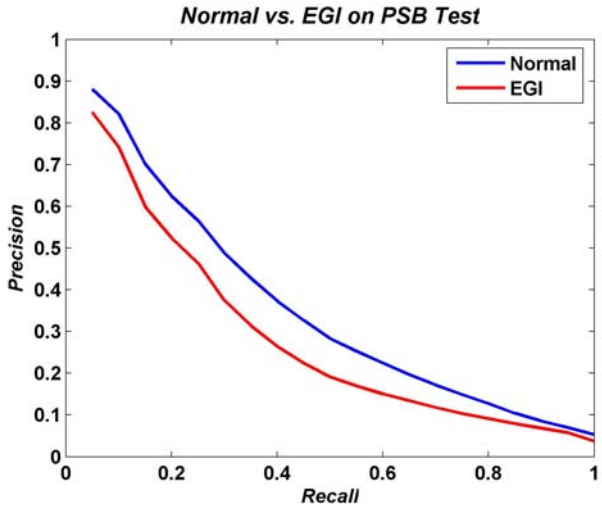

Figure 4.18: Precision-recall curve: *Normal* vs. *EGI* on PSB Test Set

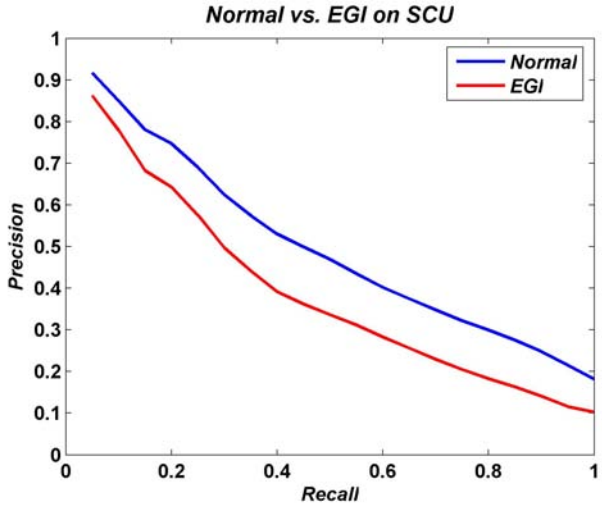

Figure 4.19: Precision-recall curve: *Normal* vs. *EGI* on SCU

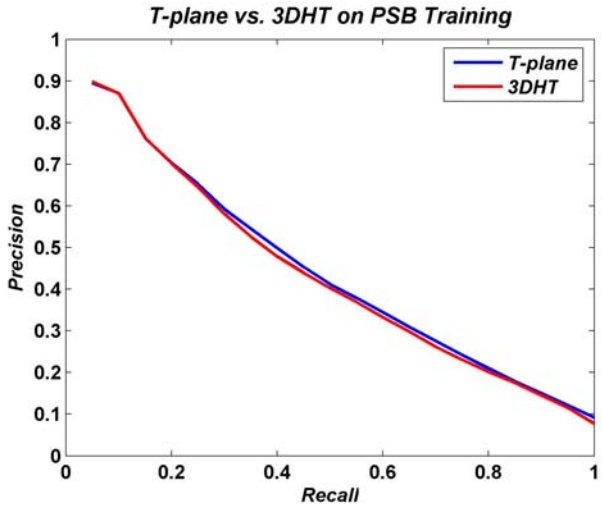

Figure 4.20: Precision-recall curve: *T-plane* vs. *3DHT* on PSB Training Set

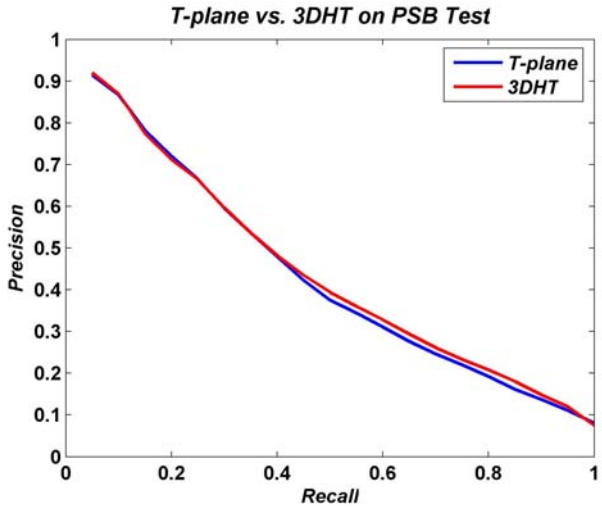

Figure 4.21: Precision-recall curve: *T-plane* vs. *3DHT* on PSB Test Set

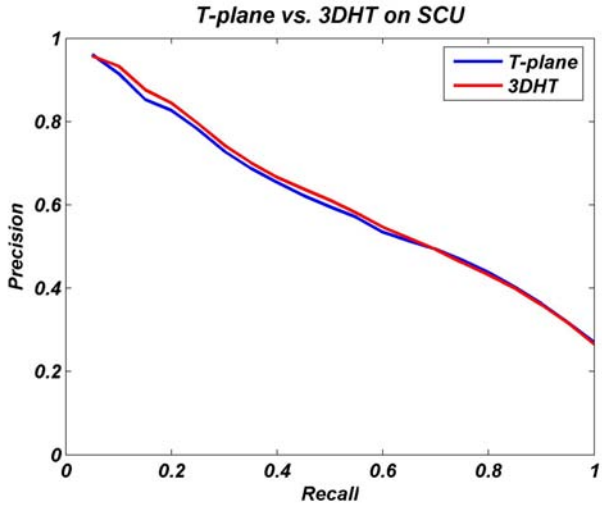

Figure 4.22: Precision-recall curve: *T-plane* vs. *3DHT* on SCU

Zaharia and Preteux [87] and we have also verified its performance-wise optimality). In Figures 4.20, 4.21 and 4.22, we compare the *T-plane* and the *3DHT* descriptors in terms of precision-recall curves. In Table 4.12, we provide the retrieval statistics. On PSB Training Set, the *T-plane* descriptor is better than *3DHT* by 1.1% DCG; on PSB Test, by just one tenth DCG percent point. On SCU, on the other hand, the 3DHT has a DCG of 72.7% against 72.0% of the *T-plane*. Since there is no convincing evidence (as compared to the previous cases) that one method outperforms the other uniformly on the available databases, we can state that the *T-plane* descriptor and its histogram-based peer *3DHT* are performance-wise equivalent.

4.8.2 General Comparison

The results in Sections 4.6 and 4.7 established that our scheme is most effective when we consider the score fusion of the *Radial*, *T-plane* and *Sec-Order* descriptors. Accordingly, we compare the retrieval performance obtained by the fusion of these descriptors against other well-known 3D shape descriptors on PSB Test Set. Apart from *3DHT* and *CAH* descriptors that we have implemented, we have taken the performance scores either from their original works or other comparison studies on PSB Test Set [23, 4].

- *Depth-Buffer Images (DBI).* This descriptor takes its roots from 2D image analysis and describes the object by six depth buffer images captured from orthogonal parallel projections [17, 2]. These binary images are then represented as concatenated vectors of Fourier coefficients of the lowest frequencies. *DBI* have been the best performing descriptor on the Konstanz database (cf. Appendix A) [2]. Here, we have taken the performance scores reported in [4].

- *Light Field Descriptor (LFD).* Like *DBI*, *LFD* is considered as a 2D method as it represents an object as a collection of 2D images rendered from the 20 vertices of a dodecahedron inscribed in a sphere enclosing the object [50]. The images can then be represented as Fourier coefficients or 2D Zernike moments as the authors suggest. In order to evaluate the dissimilarity between two objects, the corresponding *LFD*s, which consist of 20 light field images each, are aligned over all possible rotations and pairings of vertices, then the minimum distance value is taken. The performance scores reported here are from [23].

- *Radialized Extent Function (REXT).* This descriptor consists of a collection of spherical harmonics-transformed version of shape functions giving the maximal distance from the center of mass as a function of spherical coordinates (θ, ϕ) and radius. See Section 1.3.2 and [43, 17] for more details.

- *Global Spherical Harmonics Descriptor (GSHD).* Similar to *REXT*, *GSHD* results from the decomposition of a sphere into concentric shells of different radii [45]. The intersection of the shape with each of the concentric shells defines a spherical image, which is subsequently transformed to spherical harmonics coefficients. *GSHD* is rendered rotation-invariant by summing the magnitude of the harmonics coefficients within a given frequency shell [16]. See Section 1.3.2 and [45, 16] for further details.

- *Spherical Wavelet Descriptor (SWD).* This is a very recently introduced descriptor [46], overcoming the limitations of representing a spherical function in terms of spherical azimuth and elevation angles (θ, ϕ). The spherical functions are of general type, as

in *REXT* and *GHSD*. Parameterization in terms of (θ, ϕ) leads to over-accumulation of features near the poles and to sparsity near the equators. Notice that this is the same type of irregular (or non-uniform) sampling problem that we have dealt with in target selection (see Section 2.1.3). In [46], Laga et al. map the spherical domain of the function to a (unfolded) flat octahedron, which has regular 2D support. This sphere-to-octahedron mapping overcomes the irregular sampling problem and transforms the function defined on the sphere into an image defined on a regular 2D grid. Afterwards, the descriptor is obtained by applying classical 2D wavelet transform to the 2D image and then collecting a few first level's transform coefficients into a vector. Although the term "spherical wavelet" is slightly misleading as there is no genuinely spherical wavelets (e.g., second-generation wavelets [99]) involved in this procedure, *SWD* constitutes the first instance where the wavelet transform is used to describe a 3D shape.

- *Shape Distributions (D2).* This descriptor is simply a histogram of the Euclidean distance between two 3D points on the object's surface [32]. See Section 1.3.1 and [32] for details.

- *Generalized Shape Distributions (GSD).* This descriptor is introduced as a generalization to *D2*-shape distribution [33]. Accordingly, *GSD* is a 3D histogram counting the number of specific local shape pairs at certain distances: two dimensions of the histogram account for pairs of shape signatures, which are simply k-means quantized versions of spin images computed on the mesh, while the third dimension records the Euclidean distances of the local shape pairs (at which the signatures are calculated).

- *Shape Histograms (SECSHELL).* A shape histogram is defined by partitioning the 3D space into concentric shells and sectors around the center of mass of a 3D model [34]. The histogram is constructed by accumulating the surface points in the bins (in the form of shells, sectors, or both) based on a nearest-neighbor rule.

- *Voxel Descriptor (VOXEL).* This descriptor consists of merely a binary rasterization of the object boundary into a voxel grid [23].

- *3D Hough Transform Descriptor (3DHT).* See Section 1.3.1 and [38, 37] for details.

- *Extended Gaussian Image (EGI).* See Section 1.3.1 and [35, 36] for details.

- *Cord and Angle Histograms (CAH).* See Section 1.3.1 and [31] for details.

The specific instances where we compare our framework against the above descriptors are:

- *Radial+T-plane using the Invariant-L^1-metric.* This instance uses the sum of the invariant-L^1 distance values corresponding to the pdfs of $(R, \hat{\mathbf{R}})$ and $(D, \hat{\mathbf{N}})$-features. Recall that the mentioned invariance is against coordinate axis relabelings and mirror reflections, as described in Section 2.4 and experimented with in Section 4.4.

- *Radial+T-plane+Sec-Order using the L^1-metric.* This instance uses the sum of the plain L^1-distance values corresponding to the pdfs of $(R, \hat{\mathbf{R}})$, $(D, \hat{\mathbf{N}})$ and (R, A, SI)-features.

Table 4.13: Retrieval Statistics (%): State-of-the-Art 3D Shape Descriptors on PSB Test Set

Descriptor	NN	FT	ST	DCG	NDCG
DBI	66.5	40.3	51.2	66.3	17.1
Radial+T-plane (*Inv-*L^1)	68.6	39.3	50.0	65.9	16.4
SW	46.9	31.4	39.7	65.4	15.5
LFD	65.7	38.0	48.7	64.3	13.6
Radial+T-plane+Sec-Order (L^1)	65.3	35.2	45.2	62.5	10.4
REXT	60.2	32.7	43.2	60.1	6.2
GSHD	55.6	30.9	41.1	58.4	3.2
3DHT	58.8	31.1	39.6	57.7	1.9
SECSHELL	54.6	26.7	35	54.5	-3.7
VOXEL	54.0	26.7	35.3	54.3	-4.1
GSD	43.4	21.5	29.5	49.3	-12.9
EGI	37.7	19.7	27.7	47.2	-16.6
D2	31.1	15.8	23.5	43.4	-23.3
CAH	33.2	15.9	22.9	43.3	-23.5

In Table 4.13, we tabulate the full retrieval statistics corresponding to all of the descriptors in decreasing NDCG values. In Figure 4.23, we depict a visual performance landscape in terms of a DCG vs. NN scatter plot. A few comments are in order regarding these results:

- The best two descriptors are *DBI* and *Radial+T-plane* (*Inv-*L^1). While DCG of our descriptor is slightly worse than *DBI* by 0.4%, its NN score is better by 2.1%.
- The instances of the density-based framework are placed in the top league together with *DBI*, *LFD* and *SWD*.
- *DBI* and *LFD* take their roots from 2D ideas as they describe the shape in terms of 2D projections of the 3D shape. *SWD* also employs the wavelet transform in 2D although it starts from 3D shape information. Interestingly, except than our descriptors, 2D methods perform better than those using directly 3D shape information.
- Figure 4.23 reveals that performance DCG and NN scores are generally correlated, that is, if DCG is high, then NN is also high, and vice versa. The sole exception is *SWD* whose NN performance is relatively poor as compared to its DCG score.
- *Radial+T-plane+Sec-Order* marks the performance boundary between 2D methods and the remaining 3D descriptors.
- The clear performance gap between our descriptors and others such as *REXT*, *GSHD* and *3DHT* indicates that the density-based framework is currently the most successful way to process the 3D shape information for retrieval.

Finally, we note that the density-based framework coupled with FGT is computationally very efficient. On a standard laptop PC (1.86 GHz CPU and 1 GB RAM), the *Radial*, *T-plane* and *Sec-Order* descriptors can be computed under one second on average (over 907 PSB meshes). Furthermore, even without any dimensionality reduction and compression,

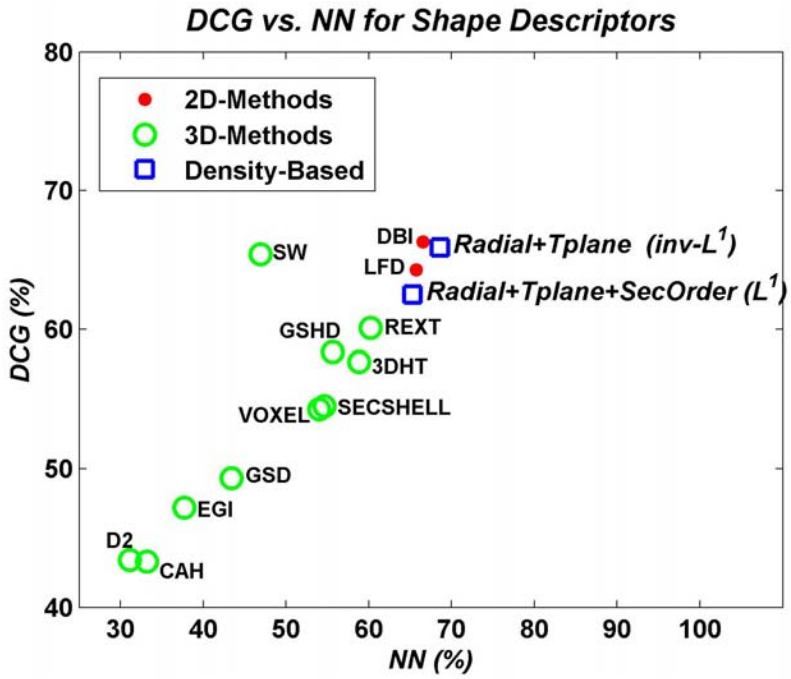

Figure 4.23: Performance landscape of 3D shape descriptors: DCG vs. NN on PSB Test Set

the storage of these descriptors coded with 16-bit double precision costs only 12 KB on average per object.

We close this section by noting that a more impressive retrieval performance on this data set has been reported using Funkhouser and Shilane's priority-driven search (*PDS*) method [4], which belongs to the paradigm of matching by feature correspondences. Given a query object and a database of target objects, all represented by sets of local 3D shape features, the *PDS* algorithm evaluates how well any subset of query features match target features. However, as the authors report in [4], this algorithm takes considerably more computation time to preprocess the database (4-5 minutes per object), more memory per object (100 KB per target object), and more time to find matches than 3D shape descriptors. In this computationally and memory-wise demanding scheme, there is no underlying global and compact shape description. DCG of *PDS* is 75.9%, 10% better than *DBI* and *Radial+T-plane* (*Inv-*L^1), indicating that, performance-wise, there is more way to go for descriptor-based 3D shape retrieval systems.

4.9 Performance Variation across Databases

In this section, we address the following types of questions concerning the retrieval performance of the density-based framework over different 3D object databases:

- Given a database, which local feature is the most effective?
- Given a feature, which database is the most challenging for the retrieval task?
- In which database, the benefits of score fusion become more important?

- Does the invariant version of the L^1-metric lead to significant improvements regardless of the nature of the database?

We illustrate the effectiveness of each feature on each database, as well as the improvements due to score fusion and to the use of the invariant metric, using "DCG-eye diagrams". These plots will help us make useful inferences at a glance. For instance in Figures 4.24 and 4.25, at each (*Database*, *Feature*)-pair, we have placed a filled blue circle (hence, the name "eye"), whose radius is proportional to the DCG of the feature pdf on that database. We have also put the exact DCG score at the center. The outer circle stands for the ideal unit DCG. Clearly, a small white space between the filled circle and the outer one indicates a good performance.

As in the previous section, we consider the *Radial*, *Tplane* and *Sec-Order* features, whose pdfs summarize respectively zero, first and second-order local surface information. Figures 4.24 and 4.25 tell us the following:

- For all databases, the *T-plane* feature performs better than the *Radial* feature, indicating that first-order surface information is more descriptive than the zero-order.

- The behavior of the *Sec-Order* feature is interesting: it performs the worst in PSB and the best in SHREC-W, when the L^1-distance is used. On a more general basis, the performance of this feature is somewhat correlated with the shape characteristics of the databases. PSB and SHREC-W represent two extremes in terms of mesh regularity and smoothness. SCU and SHREC-W have smooth and manifold meshes with second-order differential structure locally present at every mesh point. On the other hand, the meshes in PSB (referred to as triangular soups) are low resolution, non-manifold and cannot always be given a natural orientation; ESB consists of engineering parts, which contain many joints and flat patches. Consequently, for meshes belonging to these two databases, the shape index feature is either not defined everywhere or unreliable. The curvature estimation being reliable only when there is second order differential structure and data is smooth, we observe that the *Sec-Order* feature has unstable performance across different databases.

- Figure 4.25, which depicts the results of the *Invariant-L^1*-distance for the *Radial* and *T-plane* features, is compatible with Figure 4.24, except that, in SHREC-W, the performance of the *Sec-Order*-feature is no longer better than the *T-plane*. As we have already demonstrated in Section 4.4 (see Table 4.6), the invariant scheme always improves the performance for all databases and features[1].

Regarding the effectiveness of score fusion, we have tested two possible combinations: *Radial*+*T-plane* and *Radial*+*T-plane*+*Sec-Order*, using first the plain L^1-metric (Figure 4.26) and then its invariant version (Figure 4.27). Note that, in these figures, these combinations are denoted using the initials of the involved features as $R+T$ and $R+T+S$ to avoid clutter. On the effect of score fusion over different databases, we can state the following:

- In any event, the combination of the *Radial* and *T-plane* scores improves the results with respect to using solely the *Radial* score or the *T-plane* score, as established in Section 4.7.

[1] Observe that the invariant scheme does not apply to *Sec-Order* as this feature is pose-invariant by definition.

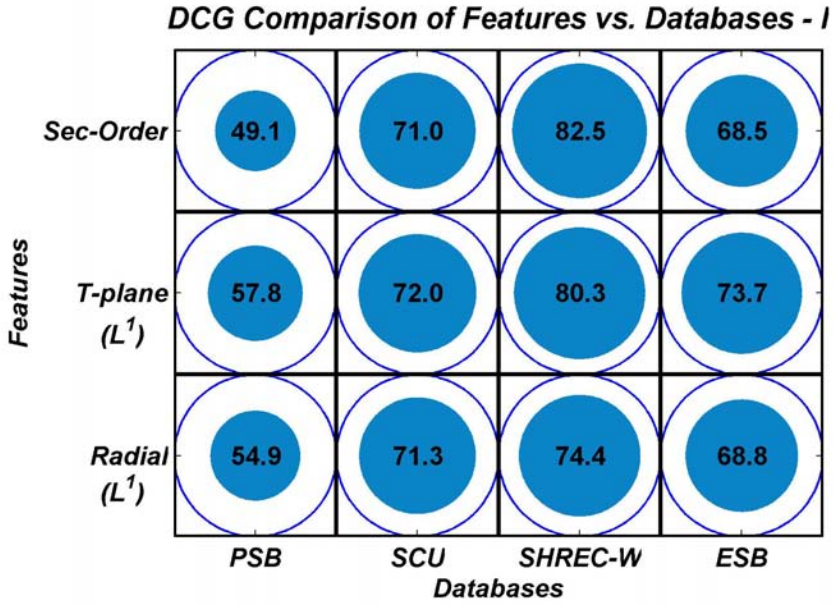

Figure 4.24: DCG-eye diagram for *Features* vs. *Databases* (L^1-metric)

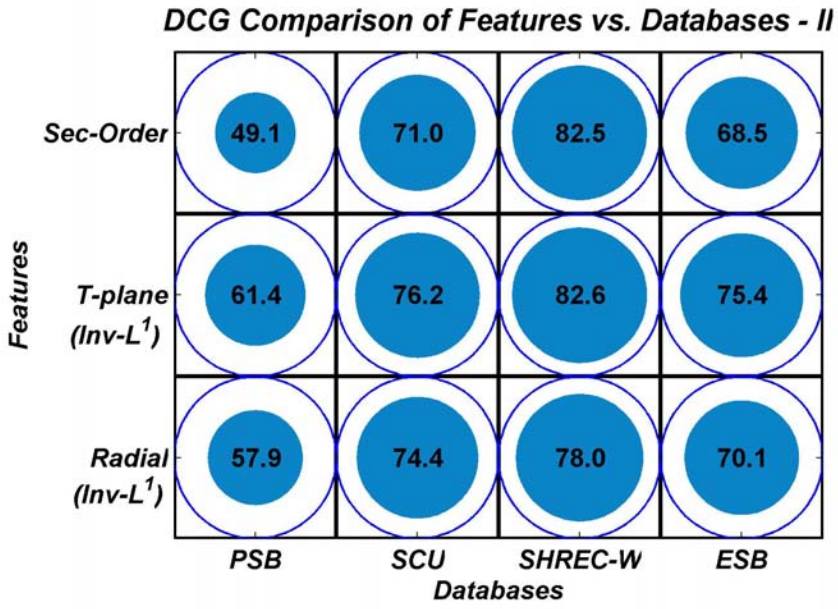

Figure 4.25: DCG-eye diagram for *Features* vs. *Databases* (*Invariant-*L^1-metric). Note that the *Invariant-*L^1-metric improves the average DCG performance over databases by at least 1.7 points for the *T-plane* and by 3 points for the *Radial* features.

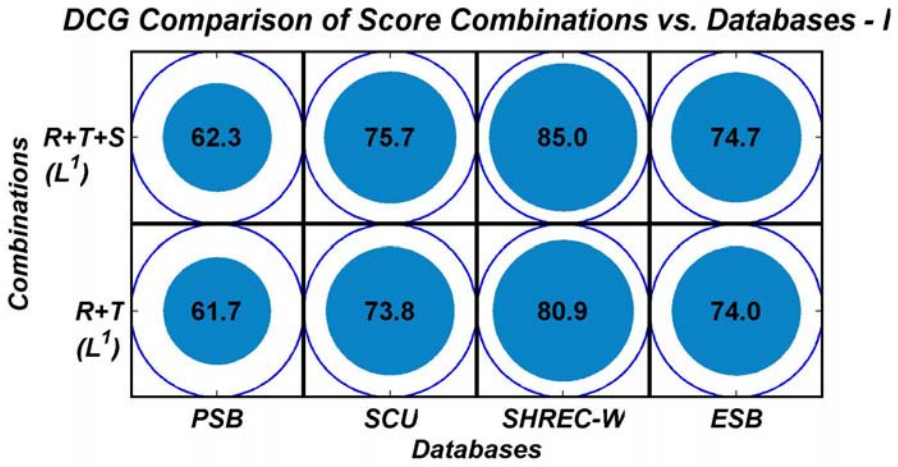

Figure 4.26: DCG-eye diagram for *Score Combinations* vs. *Databases* (L^1-metric)

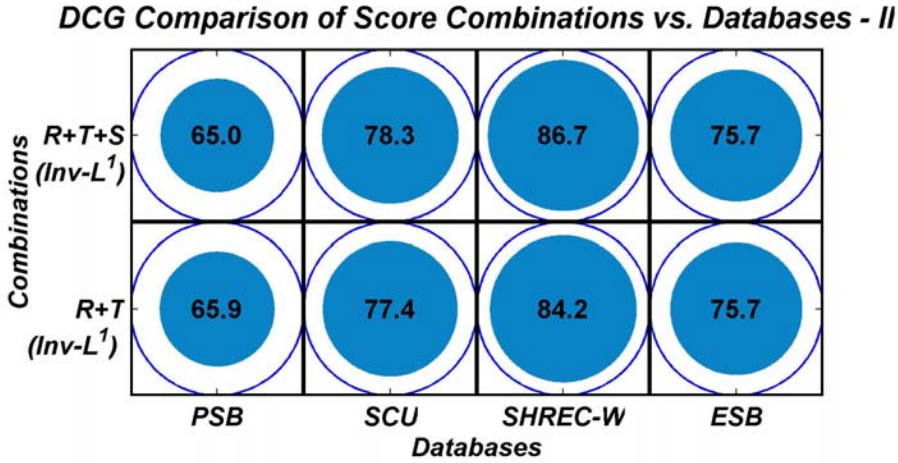

Figure 4.27: DCG-eye diagram for *Score Combinations* vs. *Databases* (*Invariant*-L^1-metric)

Table 4.14: Retrieval Statistics of the Density-Based Framework (%)

Database	NN	FT	ST	E	DCG
PSB	68.6	39.3	50.0	28.6	65.9
SCU	80.3	55.9	67.0	37.5	78.3
SHREC-W	94.5	62.7	73.5	53.1	86.7
ESB	83.9	45.9	58.6	38.2	75.7

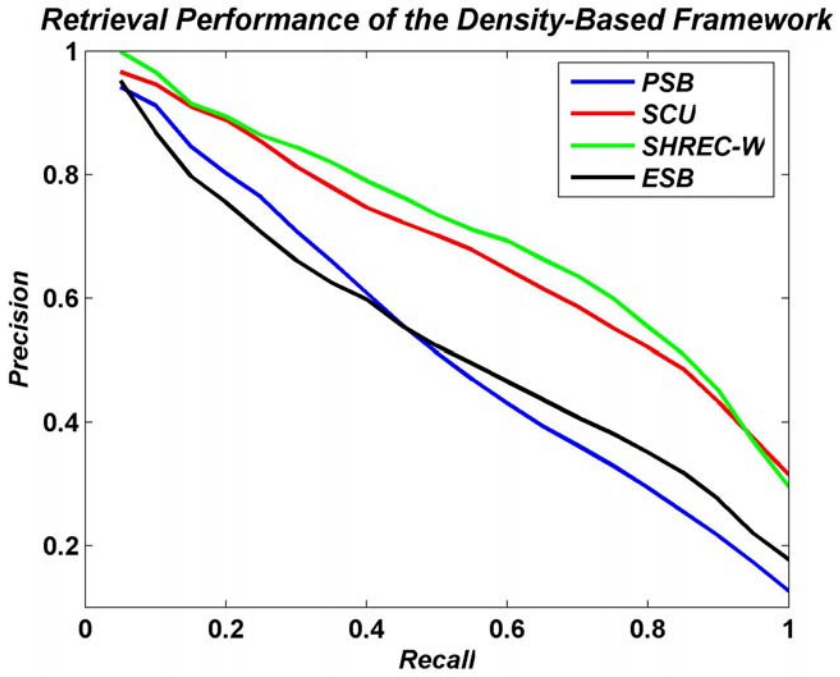

Figure 4.28: Precision-recall curves for the best the density-based descriptors on different databases

- Adding the *Sec-Order* score is significantly effective only for SHREC-W. For other databases, under the plain L^1-metric, we obtain a just noticeable improvement by including the *Sec-Order* score. Under the *Invariant-*L^1-metric, it actually worsens the performance for PSB and does not change anything for ESB. This last fact is compatible again with our comment above on mesh characteristics.
- Finally, we restate the most effective feature combinations and metrics for a given database below:
  - PSB: *Radial*+*T-plane* scores using *Invariant-*L^1,
  - SCU: *Radial*+*T-plane*+*Sec-Order* scores using *Invariant-*L^1,
  - SHREC-W: *Radial*+*T-plane*+*Sec-Order* scores using *Invariant-*L^1,
  - ESB: *Radial*+*T-plane* scores using *Invariant-*L^1.

The full retrieval statistics of these combinations are tabulated in Table 4.14. The corresponding precision-recall curves are shown in Figure 4.28. We underline the following points:

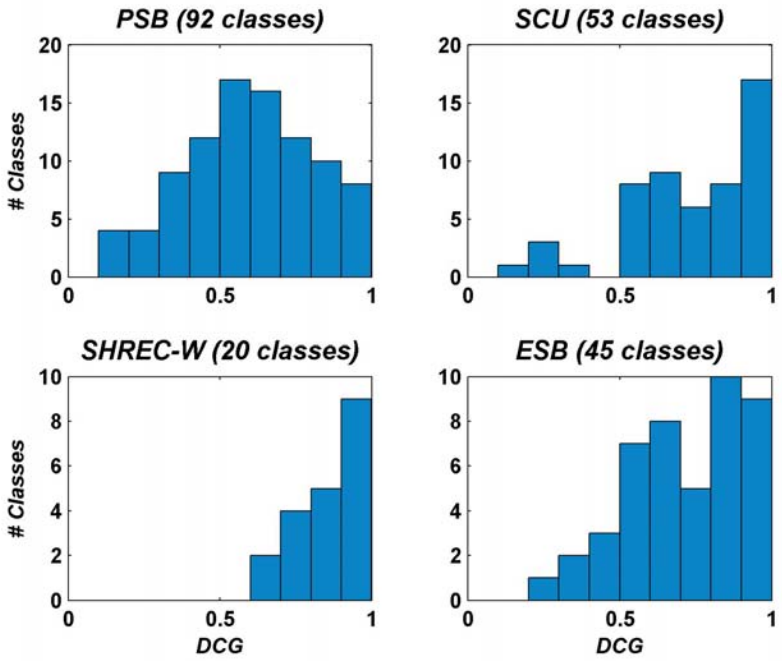

Figure 4.29: Histograms of class-wise DCGs for different databases

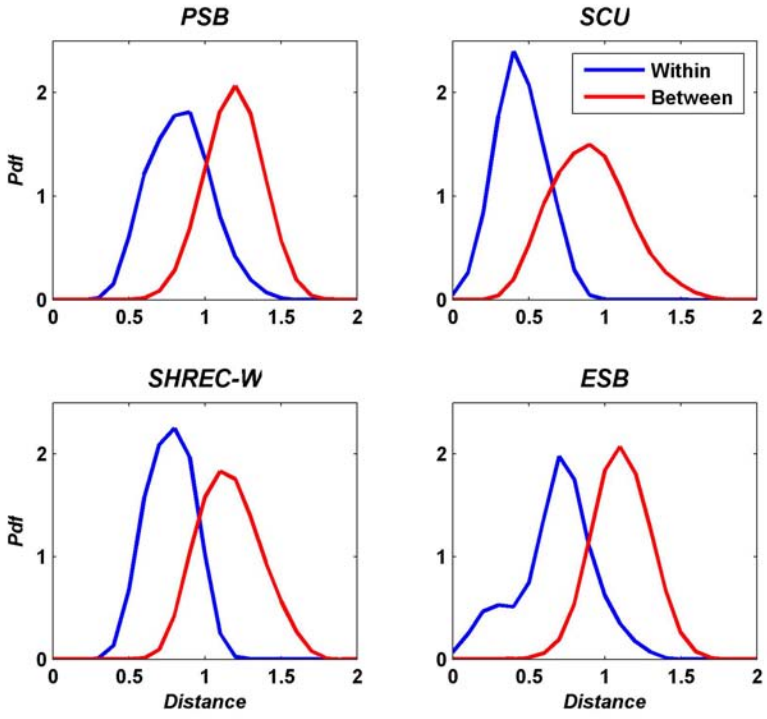

Figure 4.30: Distributions of within and between-class distances for different databases

- DCG-wise, the performance order of the density-based framework over these databases is SHREC-W $\succ$ SCU $\succ$ ESB $\succ$ PSB. This order is also evidenced in Figure 4.29 where we display the histogram of per-class DCGs for all databases: the variation in per-class DCGs follows the same order. Furthermore, as depicted in Figure 4.30, within-class distances in SHREC-W and SCU are lower on the average and more localized than in PSB and SCU.

- PSB is the most challenging database among the ones tested. The number of objects (907) and classes (92) are higher than the remainder. More importantly, in this database, classification is induced mainly by functionality-driven semantics. Since the form does not always follow the functionality, for discrimination, relying solely on global shape description has its limitations. For more intuitive and useful retrieval, we need user intervention to incorporate functionality-driven semantics to the search process, as will be explored in the final experimental Section 4.10. Nevertheless, on PSB, the density-based framework is at the top of the state-of-the-art 3D shape descriptors.

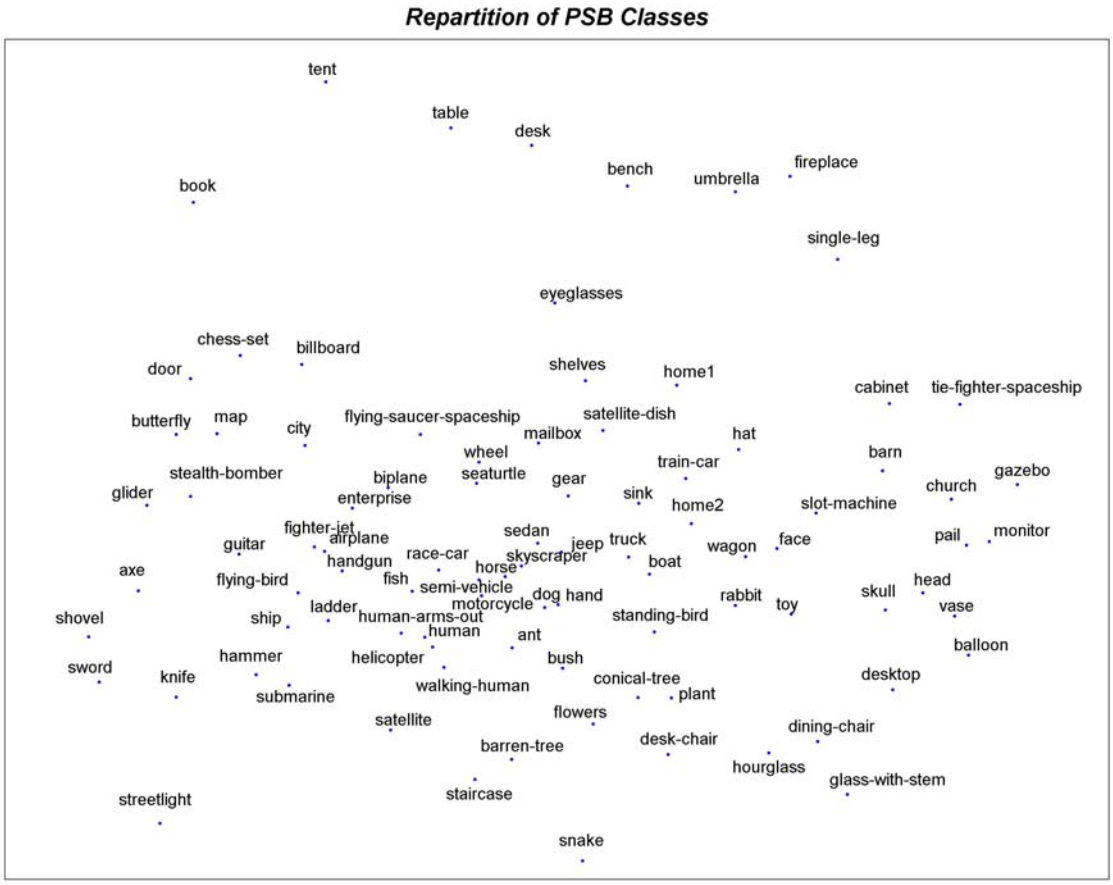

Figure 4.31: MDS map for PSB Test Set classes

In Figure 4.31, we depict the repartition of PSB classes obtained by multidimensional scaling (MDS) [13], applied on the average between-class distances given by our descriptors. MDS is a method to map pairwise dissimilarities to a lower dimensional

display where the distance order between pairs of entities is preserved as much as possible. The MDS mapping of PSB classes is open to many interesting observations showing that the density-based descriptors capture functionality-driven semantics to some extent. We itemize below some of the visible regroupments:

  - In the south-west part of the map, we see a clustering of *sharp* object classes, such as *axe*, *knife*, *shovel* and *sword*, used for cutting.
  - In the north part, we observe a super-class of *furniture* objects such as *table*, *desk* and *bench*.
  - In the south part towards the middle, classes such as *bush*, *plant*, *conical-tree*, *barren-tree* and *flowers* are clustered.
  - In the middle part towards the west, although within a clutter of irrelevant classes, we can detect a regroupment of *flying objects* such as *airplane*, *biplane*, *fighter-jet*, *enterprise-spaceship*, *stealth-bomber* and *flying-bird*.

- We are satisfied by the performance on SCU (DCG=78.3%), which consists of 513 objects in 53 classes. The best DCG performance reported on an earlier release of this database, containing 266 objects in 27 classes, was 88% due to the augmented Reeb graph approach [48].

- The least challenging database seems to be SHREC-W. Although it contains a small number of shape classes relative to the remainder, the fact that its classification is induced by topological equivalences makes it, a priori, a difficult database to be described by geometry-based methods like ours. It is interesting to observe that a geometry-based method performs so well on SHREC-W, especially on the type of classes shown in Figure 4.32. We also note that, in the Watertight track of the SHREC'07 event, our shape description scheme outperformed other geometry-based methods and had overall the second rank [26].

- The performance on ESB as measured by precision-recall is less impressive than the performance reflected by the DCG score (75.7%). In fact, the performance for $Recall < 0.4$ is worse than the more difficult PSB where our method has a DCG of 65.9%. It seems that, on ESB, the density-based framework does not always find the correct matches in the very beginning of the list but is eventually capable of retrieving them, thus lifting the DCG to a satisfactory value.

The main conclusion that we draw from this analysis is that the performance of the density-based framework generalizes well for databases with different application domains, semantics and shape properties.

4.10 Statistical Learning-Based Score Fusion

In the previous sections, we have investigated the discriminative power of the density-based 3D shape description in an unsupervised mode. Now we want to explore the contribution of a certain amount of supervision to the search process. The schemes tested in this section were described in Chapter 3.

In the following two sections, we will quantify the performance of the statistical learning-based score fusion that we interpret as an instance of supervised similarity learning. We consider the two applications introduced in Section 3.4, *bimodal* and *two-round* searches,

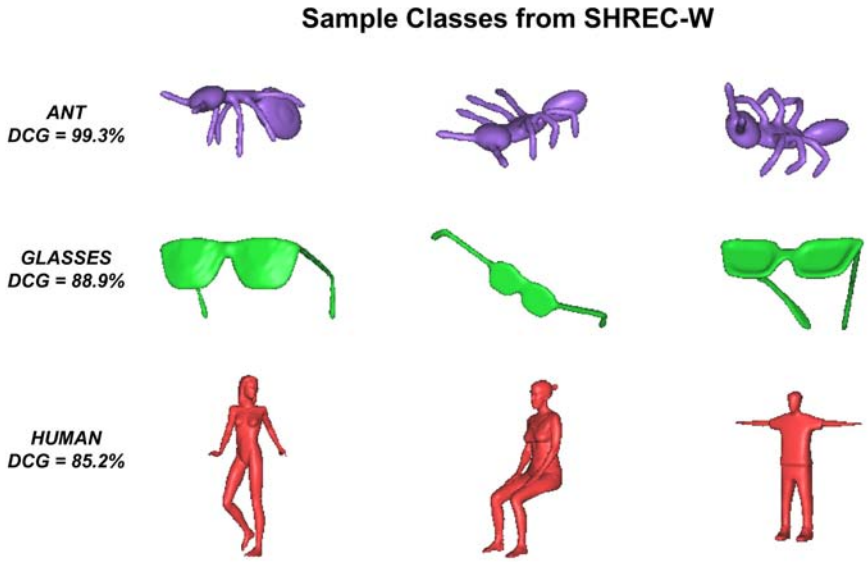

Figure 4.32: A few challenging SHREC-W classes where the density-based framework performs particularly well

to supervise the querying process and then to fuse the similarity scores resulting from different descriptors. Naturally, we employ instances of the density-based framework, namely *Radial* $(R, \hat{\mathbf{R}})$, *T-plane* $(D, \hat{\mathbf{N}})$ and *Sec-Order* (R, A, SI)-descriptors, but in a way different than the previous sections. Note first that all of these descriptors are radialized in the sense that, at a fixed distance from the origin r_k (or d_k for the *T-plane*), they capture the distribution of the remaining features at concentric shells. These remaining features are the radial direction $\hat{\mathbf{R}}$ for the *Radial*, the normal $\hat{\mathbf{N}}$ for the *T-plane* and the (A, SI)-pair for the *Sec-Order*. We refer to these distributions as cross-section descriptors. For instance, let us take the $N_R \times N_{\hat{\mathbf{R}}} = 8 \times 128 = 1024$-target pdf of the $(R, \hat{\mathbf{R}})$-feature. Recall that $N_R = 8$ is the number of points sampled within the R-domain and $N_{\hat{\mathbf{R}}} = 128$ is the number of points sampled on the unit-sphere, in order to form the target set for this descriptor. The 1024-point $(R, \hat{\mathbf{R}})$ descriptor is then considered as $N_R = 8$ chunks of $N_{\hat{\mathbf{R}}} = 128$-point cross section descriptors, each of which can be used to evaluate a similarity score $sim_k \triangleq s_k$ between two objects at a given concentric shell, say at a distance r_k from the origin. Of course, these individual scores do not capture the shape similarity to the full extent. However, this decoupling adds more degrees of freedom to the subsequent score fusion stage, where we learn a distinct weight w_k for each of the individual scores s_k by ranking risk minimization (cf. Section 3.2). Accordingly, for each of the *Radial*, *T-plane* and *Sec-Order* descriptors, we obtain $N_R = 8$ per-chunk similarity scores. Notice that under the L^1-metric or its invariant version, summing these scores is the same as taking the distance directly using the original full descriptors. The idea of fusing per-chunk scores subsumes a linear similarity model. More precisely, given a query q and a database object x, we seek an optimal similarity or scoring function of the form $\varphi(x, q) = \sum_k w_k s_k$ as in Section 3.3. In the following experiments, we use the *Invariant-L^1*-metric to quantify per-chunk similarities and compare the results against the basic *SUM* rule of the form $\varphi_0(x, q) = \sum_k s_k$.

The following performance results have been obtained on a modified version of PSB. Originally, PSB Training and Test Sets do not share the same shape classes. Accordingly, we have merged these two sets into a single one, consisting of 1814 models in 161 classes[2]. We have then split them into two subsets A and B of sizes 946 and 868, sharing the same 161 classes. We will clarify the use of these two sets in the context of the application chosen. This modification lets us treat the problem in an even more challenging complexity than in the original PSB Test Set, since the number of classes is now increased from 92 to 161.

4.10.1 Performance in the Bimodal Search

Recall that the bimodal search protocol assumes the existence of a training set categorized into a fixed set of concepts. Learning is done off-line. From the training set, we take one shape at a time as a query and the remaining ones as database shapes, then feed the pair-wise scores to the algorithm in Section 3.3 to learn a shape-specific SVM. In Section 3.4.1, we had to introduce two different heuristics to obtain a weight vector corresponding to each class concept, since minimizing directly a *per-concept* risk functional using all the available instances was computationally too complex. We remind that, in the *AVE-W* heuristic, the weight vector for a given concept class is computed as the average of the per query weight vectors corresponding to the training shapes within that class. In the second *PCMIN-W* heuristic, on the other hand, we minimize a *per-concept* risk by using only support vectors learned during the prior *per-query* learning.

In the bimodal experiments, we have taken the PSB Set A as the training set, which we have used to learn per-concept weight vectors. PSB Set B has been reserved for testing purposes. In Table 4.15, we provide the results of fusing 8 *Radial*-scores and 8 *T-plane* scores, making 16 scores in total. In Table 4.16, we report the performance after including 8 more scores corresponding to *Sec-Order* chunks in addition to *Radial* and *T-plane* scores. We also display the results of the basic *SUM* rule for reference.

Although, the learning-based score fusion does improve the average DCG performance significantly on the training set, it does not lead to a significant gain in the test set (only 1% using the *AVE-W* heuristic). That is, learning-based score fusion does not work well for certain concepts. This might be due to heuristics-based learning of per-concept weight vectors, but, we think that the following arguments better explain the situation.

- For some concepts, the linear similarity model might not be flexible enough to maintain good classification accuracy in the score difference domain. When instances from queries belonging to a certain concept are pooled together, the discrimination problem in the score difference domain might become more complex than what can be solved using a simple linear decision boundary.

- If the linear similarity model were totally unacceptable, we would not expect a good performance on the training set either, as in Tables 4.15 and 4.16. In fact, in only 4 out of 161 concepts in PSB Set A, the *AVE-W* fusion of the *Radial* and *T-plane* scores has worsened the performance by at most 2.3% DCG points with respect to the baseline *SUM* rule. In PSB Set B, on the other hand, 61 concepts (again out of 161) have suffered an average performance loss of 8.5% DCG points.

[2] The number of classes shared by training and test sets is 21, hence the merged PSB contains $90 + 92 - 21 = 161$ classes.

Table 4.15: DCG Performance of Score Fusion on Bimodal Protocol using 8 *Radial* and 8 *T-plane*-Scores

Rule	PSB Set A	PSB Set B
SUM	61.9±28.4	61.8±27.9
AVE-W	69.4±27.0	62.8±28.1
PCMIN-W	71.1±26.3	62.1±27.6

Table 4.16: DCG Performance of Score Fusion on Bimodal Protocol using 8 *Radial*, 8 *T-plane* and 8 *Sec-Order*-Scores

Rule	PSB Set A	PSB Set B
SUM	61.6±28.1	60.6±28.1
AVE-W	71.8±26.5	62.6±28.4
PCMIN-W	74.9±25.2	62.5±27.7

- In Tables 4.17 and 4.18, we provide the DCG scores when we use the basic *SUM* rule instead of learning-based score fusion (*AVE-W* or *PCMIN-W*) for negatively affected concepts (i.e., those concepts for which learning-based score fusion has worsened the DCG performance). The right most columns give the number of positively affected concepts, indicating in which proportion the learning-based score fusion generalizes satisfactorily. We deduce that the linear similarity model is adequate for the training set and generalizes well on the previously unseen instances of ∼100 concepts in the test set.

- A plausible reason for the lack of generalization for certain concepts is that the score differences between training shapes might not be representative enough for those between test shapes. In such cases, the parameters of the linear decision boundary might overfit the training instances.

4.10.2 Performance in the Two-round Search

In the two-round query formulation, the benefits of the proposed score fusion scheme become much more evident. In Section 3.4.2, we have introduced two variants of this protocol, with varying amount of user interaction needed. In the on-line variant, the system learns a *per-query* weight vector at query time after having been provided with M marked shapes (either as *relevant* or *not relevant*) of the first round. In the off-line variant, the user is asked to mark just one single shape among the M displayed ones after the first round. The second round proceeds with the weight vector associated with the marked shape, which has been learned off-line. To evaluate the performance in this search protocol, we have reserved the PSB Set A as the database shapes and PSB Set B as the query shapes. The first round results have been obtained by the basic *SUM* rule.

In Figures 4.33 and 4.34, we display the DCG performance of the *on-line* sub-protocol as a function of the number of *marked* items M from 4 to 32. Figure 4.33 illustrates the results for 16 *Radial* and *T-plane* scores, while in Figure 4.34, we have also included 8 *Sec-Order* scores. In these figures, the line at the bottom stands for the DCG of the first round (i.e., the performance of the *SUM* rule, DCG = ∼62% for all score combinations). The line

Table 4.17: DCG Performance of Score Fusion on Bimodal Protocol using 8 *Radial* and 8 *T-plane*-Scores when the basic *SUM* rule instead of learning-based score fusion has been used for negatively affected concepts

Rule	PSB Set B	# Positively Affected Concepts
SUM	61.8±27.9	-
AVE-W	64.6±24.2	100
PCMIN-W	64.8±23.8	88

Table 4.18: DCG Performance of Score Fusion on Bimodal Protocol using 8 *Radial*, 8 *T-plane* and 8 *Sec-Order*-Scores when the basic *SUM* rule instead of learning-based score fusion has been used for negatively affected concepts

Rule	PSB Set B	# Positively Affected Concepts
SUM	61.8±27.9	-
AVE-W	64.0±24.1	106
PCMIN-W	64.4±23.9	100

at the top stands for the DCG when all database models are marked as either relevant or non-relevant, serving as an empirical ideal, i.e., the maximum achievable DCG on this data set using the presented score fusion algorithm and the running set of description schemes (DCG = ∼75% for *Radial* and *T-plane* scores, DCG = ∼79% when *Sec-Order* scores are also included). Based on these results, we make the following comments:

- As the number of marked items M increases, we observe a steep increase in the DCG performance, compatible with theoretical fast rates of convergence proven in [11, 10].
- The DCG profile converges smoothly to the empirical ideal as the user marks more and more items in the first round.
- To give certain performance figures, for $M = 8$, DCG obtained after fusing *Radial* and *T-plane* scores becomes ∼67%, giving a 5% improvement compared to the baseline. When the *Sec-Order* scores are included, the DCG gets ∼68%.
- The 70% DCG barrier is reached after $M = 16$ marked items for *Radial* and *T-plane* scores. Adding *Sec-Order* scores reduces this number down to $M = 12$, showing in fact the *Sec-Order* scores can indeed be useful in contrast to the case of the basic *SUM* rule.

In Figures 4.35 and 4.36, we display the DCG performance of the *off-line* sub-protocol as a function of the number of *displayed* items M again 4 to 32. We emphasize that, in this mode, M refers to the number of *displayed* items and the user interaction needed is limited to mark just one shape, the first relevant one after the first round. Accordingly, here, M is not related to the convergence of the algorithm, but just an application-specific parameter determining how far the user is allowed to look for relevant items. Increasing M does not cost anything in terms of user interaction. After this clarification, we deduce the following:

- At $M = 1$, combining *Radial* and *T-plane* scores boosts the retrieval performance by 2.5%, adding the *Sec-Order* scores by 3.5%.

- The DCG profile keeps a slow but constant increase as the number of displayed items M in the first round is increased.
- In a typical retrieval scenario, displaying $M = 32$ items has no cost. These results tell us that we can obtain DCG improvements by $\sim$4% with respect to the baseline. Recalling from Section 4.8.2 that the performance of top 3D shape descriptors differ only by a couple of percentage points, this 4% gain can be considered as significant and comes virtually at no cost at the querying process. The only bottleneck is the off-line processing of the database shapes to learn the weight vectors, which may eventually be used in the second round.

When we contrast the on-line and off-line versions of the two-round search protocols, we arrive at a compromise. With on-line score fusion, we can obtain significant improvements as the user is asked to mark more and more items. In special applications where the user voluntarily marks the demanded number of items, the on-line scheme is preferable. The off-line scheme, on the other hand, comes at no cost at query time and still yields satisfactory improvements. Sample two-round searches using these two variants are shown in the Appendix E.

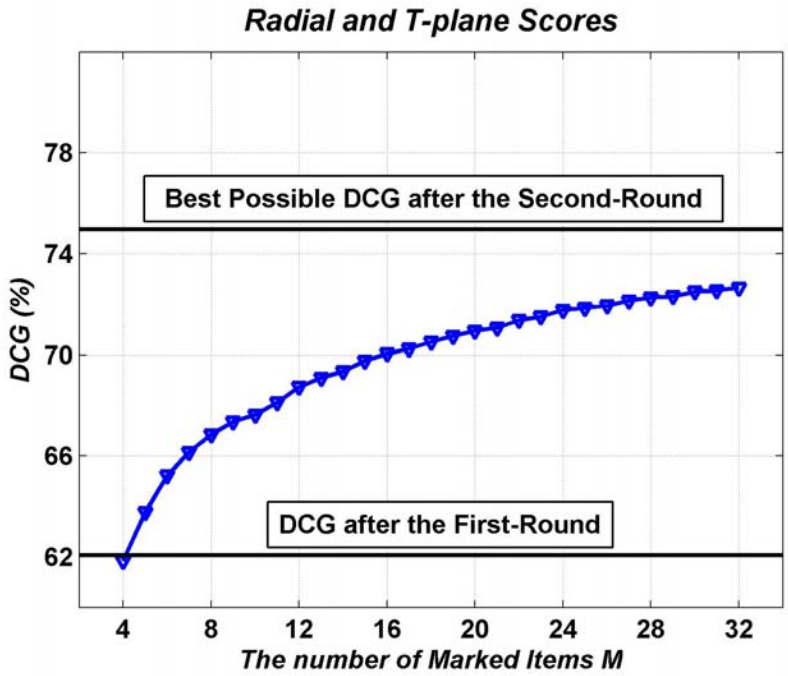

Figure 4.33: DCG performance of the two-round search with *on-line* learning as a function of the number of *marked* items M in the first round for *Radial* and *T-plane* scores

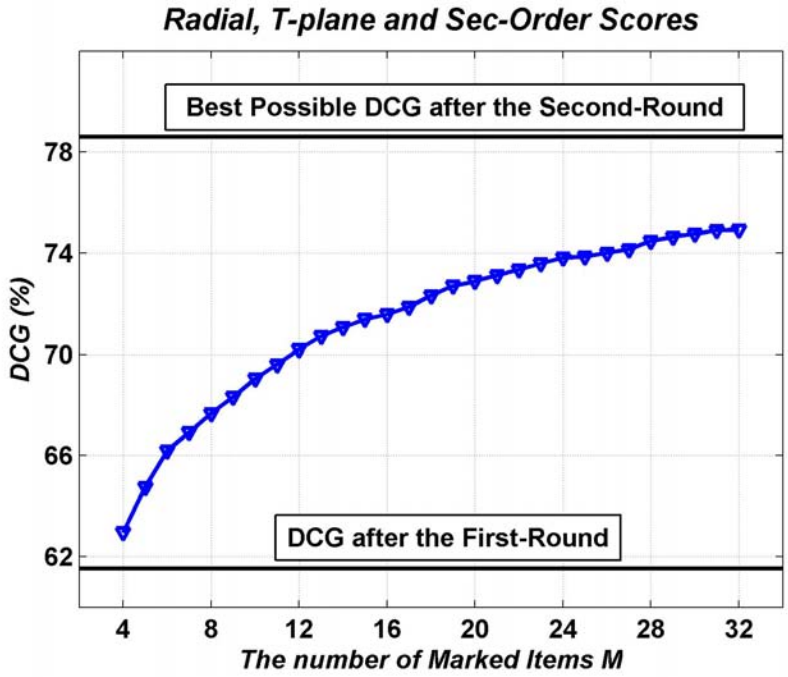

Figure 4.34: DCG performance of the two-round search with *on-line* learning as a function of the number of *marked* items M in the first round for *Radial*, *T-plane* and *Sec-Order* scores

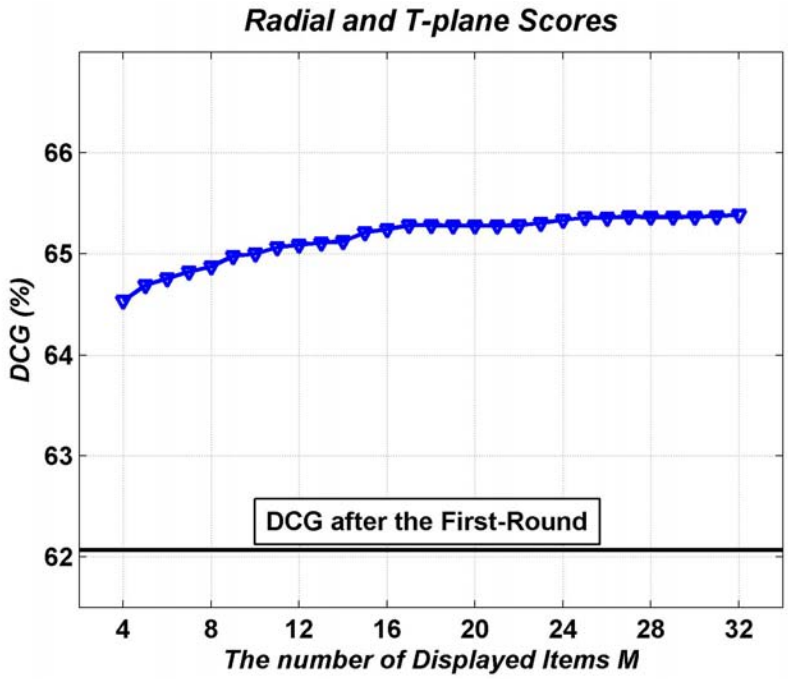

Figure 4.35: DCG performance of the two-round search with *off-line* learning as a function of the number of *displayed* items M in the first round for *Radial* and *T-plane* scores

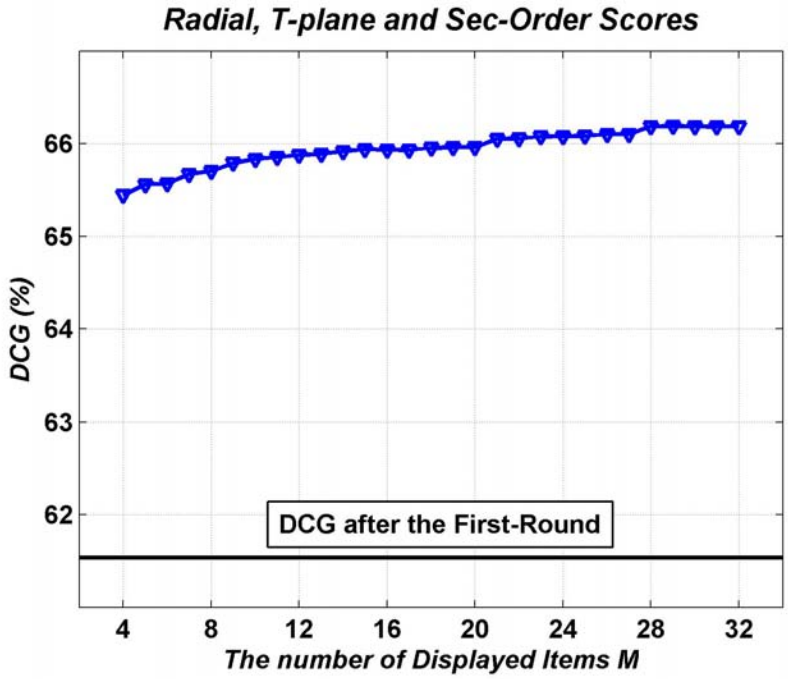

Figure 4.36: DCG performance of the two-round search with *off-line* learning as a function of the number of *displayed* items M in the first round for *Radial*, *T-plane* and *Sec-Order* scores

Chapter 5

Conclusion and Perspectives

5.1 Discussion and Conclusion

This thesis has dealt with two fundamentals problems in content-based 3D object retrieval:

(1) How to describe a 3D shape to obtain a reliable representative for the subsequent task of similarity search?

(2) How to supervise the search process to render it more effective and semantic?

Our contributions concerning *the first problem* are as follows:

- We have developed the density-based framework, which produces not only a single but a family of shape descriptors at various levels of discrimination.
- Our framework unifies histogram-based approaches and enables the passage from multivariate local surface characterization to global shape description with considerable ease. Performance-wise, density-based descriptors are significantly better than their histogram-based counterparts[1].
- We have shown that the density-based framework proves to be very effective in comparison to state-of-the-art descriptors. In fact, one of its instances performs better than all *purely* 3D methods and equally well as *DBI* [2], which is the best 2D method to date.
- We have tested our framework on four different 3D databases, which are selected from distinct domains, in order to show the potential of density-based descriptors in a wide range of applications targeted by the 3D object retrieval research. We have shown that their performance generalize well on these databases with varying mesh quality, semantic content and classification granularity.
- We have developed dedicated dimensionality reduction tools, marginalization and probability density pruning, which exploit the pdf structure. Furthermore, we have shown that marginalization can be employed for non-heuristic feature subspace exploration in terms of retrieval performance.
- We have demonstrated that the pdf structure can also be used to guarantee invariance to certain 3D transformations at matching stage.

[1] Meanwhile, we note that the *T-plane* descriptor is just equally well as *3DHT*.

- We have derived an upper bound on the mean integrated absolute error (*MIAE*) of the univariate KDE and illustrated the smoothing effect of the bandwidth parameter(s) in the context of shape discrimination.
- We have proposed a robust feature calculation scheme in order to obtain more reliable feature observations from low-resolution meshes.
- Finally, the present work constitutes a first application of FGT in the 3D domain. This fast approximation scheme turns the KDE method, which is computationally practical only in a few dimensions, into a computationally very efficient tool for 3D shape description.

We summarize and discuss the key aspects of the density-based framework below:

- **Local Characterization by Joint Features.** The features used in the density-based framework are the most basic and direct differential geometric quantities that can be associated with a 3D point lying on a surface. They are intuitive from a shape modeling perspective and computationally cheap to obtain. Furthermore, the fact that they have mathematically well-defined ranges helps us in the subsequent target selection.

  A joint consideration of these features characterize the surface point completely up to second-order. As we add more dimensions to a multivariate feature, we increase its specificity. When coupled with the KDE, this family of local features leads to a family of descriptors, each of which has a different level of discrimination.

  We have not particularly refrained ourselves from the use of non-invariant features, since invariance at feature-level comes usually at a certain loss of shape information, which cannot be restored back afterwards. We have employed PCA-based normalization to discover the major axes of the object to secure rotation invariance without loss of information, but at the cost of small errors for which we have relied on the robustness of the KDE. More destructive deficiencies such as mislabelings and wrong polarity assignments have been handled in an exact manner at the matching stage.

- **Robust Feature Calculation.** Although a triangular mesh leads to a plausible visualization of the 3D object at changing levels of resolution, it is somewhat artificial and approximates the continuous surface only in the limit, i.e., when the triangles are sufficiently small and regular. The Simpson averaging scheme provides a robust feature calculation option by smoothing the feature values over a mesh triangle so that mesh irregularities become a lesser problem.

- **Global Description via Pdfs.** If the pairwise correspondences between two sets of feature observations are known in advance, two shapes can be matched much more effectively than by using any known 3D shape descriptor. The superior performance of the *PDS* method [4] is a manifestation of this fact. This correspondence problem is the main bottleneck for global 3D shape descriptors, which all try to circumvent it in one way or another, by registering the shape information on some adequately chosen grid. The KDE-based framework too effects a kind of registration but, unlike other approaches, adopts a soft assignment strategy so that a feature observation does not only contribute to the closest grid point but also to the nearby ones as a function of its closeness. This strategy makes the resulting descriptors less sensitive to small shape variations and extrinsic effects.

Closely related to this robustness property is the smoothing behavior of the KDE. In fact, the idea of variability reduction by smoothing has been our main motive in using KDE. Adapting the example given by Edelman in [100], the method of comparing two feature sets must allow for some location uncertainty; otherwise, any small perturbation would render an object unrecognizable. Testing all possible perturbations within a small neighborhood around each feature observation raises combinatorial problems and hence is unfeasible. One simple way to achieve relative insensitivity to feature location uncertainty is to blur at least one of the feature sets before comparison, e.g., by convolving it with a bell-shaped kernel such as the Gaussian. The bandwidth parameter of the kernel accounts for this very uncertainty as discussed in Section 2.2.2. Furthermore, by judiciously adjusting the bandwidth parameter, the descriptors can be rendered smoother or more specific. Large bandwidths enhance shape commonalities, discard very fine maybe irrelevant details and favor the matching hypothesis by reducing descriptor variation, while small bandwidths produce more faithful descriptors and impose more stringent requirements for a positive match to occur.

Finally, along with the pragmatic benefits mentioned above, describing a 3D shape by the pdf of its local features is also philosophically appealing. The density-based framework is a generative method in the sense that it models the conditional density of a local feature $S \in \mathcal{R}_S$ given a certain shape X. Accordingly, let $\mathcal{S}$ be a set of feature observations regarding X. Given two shapes O_1 and O_2 modeled by the conditional pdfs $f_{S|O_1}$ and $f_{S|O_2}$ respectively, we consider the following question: *apart from X, which one of O_1 and O_2 could have also generated $\mathcal{S}$?* If the problem is treated as the one of binary classification, the optimal solution is given by the Bayesian decision rule of the form "assign $\mathcal{S}$ to O_1 if $f_S(\mathcal{S}|O_1) \geq f_S(\mathcal{S}|O_2)$, and to O_2 otherwise". The error of this rule can be expressed as [13]

$$Pr(error) = \frac{1}{2} \int_{\mathcal{R}_S} min\left(f_{S|O_1}, f_{S|O_2}\right),$$

which can be viewed as a measure of *confusion* between $f_{S|O_1}$ and $f_{S|O_2}$, as well as a measure of *similarity*. In the context of matching via pdfs, to determine which one of $f_{S|O_1}$ or $f_{S|O_2}$ better match $f_{S|X}$, we have to pick the model $O^*, i = 1, 2$, which maximizes the confusion with $f_{S|X}$, that is,

$$O^* = \underset{i=1,2}{argmax} \int_{\mathcal{R}_S} min\left(f_{S|X}, f_{S|O_i}\right).$$

As shown in Appendix B.1, the above maximization is equivalent to minimizing the L^1-distance between $f_{S|X}$ and $f_{S|O_i}, i = 1, 2$. Observe that this equivalence is distribution-free and independent of feature dimension. This result does not only explain the superior performance of the L^1-metric notably against other Minkowski metrics (cf. Section 4.4), but also provides a rigorous justification for the adequacy of our approach. We conclude that, from the perspective delineated above, the density-based framework is the right way to process local surface information for shape matching.

- **Marginalization.** This fairly general concept of probability theory has found an interesting application in our framework. Using marginalization, we can switch from

one descriptor to a less specific one, by discarding one or more feature components. Accordingly, we have avoided descriptor computation for every possible feature configuration and have been able to explore the whole feature space in terms of retrieval performance very efficiently and without resorting to any suboptimal heuristic-based feature selection method.

- **Invariance at Matching.** Another advantage that the pdf structure offers is to secure invariance against certain types of object transformations at the matching stage, without the need of recomputing the descriptor for every possible transformation. Starting from the change of variables formula, we have developed a minimization-based similarity measure, completely invariant to coordinate axis relabelings and mirror reflections, without sacrificing any intrinsic shape information. This measure has provided significant performance improvements in all databases.

Concerning *the second problem*, we have adapted the theoretically well-founded ranking risk minimization framework [10, 11] to the learning of inter-shape similarities for more effective and semantic retrieval as shown in Section 4.10 and illustrated in Appendix E. We have cast the problem into one of linearly combining similarity scores, which arise from different shape descriptors, and have shown that learning can be carried straightforwardly by the well-known SVM algorithm. This rigorous score fusion approach allows us to benefit from powerful aspects of individual descriptors by incorporating different levels of supervision into the search process. Our contributions on similarity learning are as follows:

- To the best of our knowledge, there is no prior work in the 3D domain using statistical learning techniques to combine multiple shape descriptors. The present work constitutes a first successful attempt to this challenging problem.

- The approach presented in this thesis is a direct application of a recently introduced rigorous statistical framework [10, 11], where consistency and fast rate of convergence of empirical ranking risk minimizers have been established. These theoretical results place our approach on well-founded grounds.

- Our algorithm operates on scores, and not on descriptors themselves, unlike other risk minimization-based approaches [88, 89]. In this sense, it is a general scheme, which does not depend on the type of description and can be used for arbitrary data modalities as long as one provides similarity scores between entities.

- We have illustrated the use of our scheme in two different search protocols. Experiments have shown that the two-round protocol benefits more from our score fusion framework than the bimodal protocol. The approach is able to boost the retrieval performance for as many as 100 shape concepts under bimodal query formulations. In the two-round search, on the other hand, we can obtain DCG improvements by as large as $\sim$10% even with a fairly modest amount of user interaction.

5.2 Perspectives

An immediate research direction to pursue is the extension of invariances, which can be achieved using the scheme presented in Section 2.4, to arbitrary 3D rotations. Such an extension would render the matching completely invariant to rotations as well, and would further improve the retrieval performance of the density-based descriptors. Apart from 3D

object retrieval, recovering the pose of a 3D object with respect to a given reference frame is still an open problem and constitutes the preliminary step in applications such as 3D model reconstruction from existing parts, texture and surface detail mapping, etc. [101]. In order to provide a solution to this problem within the density-based framework, we need to find unit-sphere partitionings that would remain closed under the action of an arbitrary 3D rotation as discussed in Section 2.4. Alternatively, we can relax the strict closedness requirement and rely on nearest-neighbor mapping to establish target correspondences after a certain 3D rotation.

Another possible direction of research is towards *parametric density estimation*. While the KDE approach coupled with FGT proved to be very useful; beyond four dimensions, descriptors get impractically large and the statistical accuracy is no longer guaranteed due to the curse of dimensionality. In our context, the virtues of parametric mixture models are as follows.

- Accuracy problems related to high dimensionality can be more effectively managed.
- They lead to more parsimonious descriptions. If a local feature pdf were available in terms of a parameterized analytical form, we would not need to sample it to obtain the descriptor vector but just store its parameters as the descriptor.

However, we should think of alternative matching schemes since the computational cost of parametric pdf estimation cannot be afforded at the matching stage. Given a set of i.i.d. feature observations $\mathcal{S} = \{s_k\}_{k=1}^{K}$ regarding the query and a set of database objects $\mathcal{O}$, each represented by its own parametric pdf, the matching problem would then be to select the pdf that best explains the set $\mathcal{S}$ by invoking a Bayesian classifier of the form "assign $\mathcal{S}$ to O^* such that $O^* = \underset{O \in \mathcal{O}}{argmax} \sum_k log f_S(s_k|O)$". This maximum likelihood procedure can be carried efficiently since the pdfs are analytically given. Furthermore, using a parametric pdf, we could more conveniently extend the type of invariances that can be secured at the matching stage. The research problems in this direction are:

- Finding a mixture model that would be flexible for a wide range of local features,
- Determining the number of parameters involved in the mixture,
- Efficient estimation of the mixture parameters.

Motivated by the fact that our framework is based on pdfs, a natural research direction would be an *information-theoretic analysis of the local surface features*. In the present work, we have quantified the descriptive power of the local features by their retrieval performance. Information theory provides us with universal measures for the same end and raises the following interesting issues:

- Let, for instance, Y be the discrete class label of a random 3D object O on which the feature S is evaluated. The mutual information value $I(S,Y)$ stands then for the amount of information that the feature S contains about the class label Y. It is the reduction in the uncertainty of the unknown class label due to the knowledge of the feature. Consequently, $I(S,Y)$ arises as a natural measure, which can be used to order different local features in terms of their descriptive power. It would also be interesting to compare such an ordering, once found, to the one given by an empirical performance evaluation.

- If we consider a 3D object as a communication channel, the entropy of the local feature can be viewed as the average number of bits transmitted to pass its true value to a recipient. This observation may have implications in 3D mesh compression.
- Along the same line with the above items but on an even more theoretical vein, it is tempting to investigate the relationship between local feature entropy and global shape complexity.

Finally, the score fusion scheme presented in this thesis leaves the door open for two important research issues:

- Scoring functions can be extended to *non-linear* forms to cope with situations where a linear decision boundary is not flexible enough to discriminate between score difference vectors. The linear scoring function $\varphi(\mathbf{w}) = \langle \mathbf{w}, \mathbf{s} \rangle$ induces a linear discrimination problem in the score difference domain where the optimization variable is again the weight vector $\mathbf{w}$. As a consequence, once the weight vector is learned in the score difference domain, it can directly be used in the score domain too. In principle, the discrimination problem can be non-linearized by the kernel trick, in which case the sought-after weight vector no longer lives in the score difference domain, but in a higher dimensional space: it is only implicitly given except for some special choices of the kernel such as the polynomial [94]. In a standard classification problem, this poses no problem as the decision function can be efficiently evaluated in terms of kernel values. In the retrieval problem as considered in this thesis on the other hand, decision values can only be used to find the relative rank of an item with respect to another. The main questions here are:
  - How can we resolve the final ordering from relative pairwise ranks in a way fast enough for a real-time application?
  - Does this resolution phase lead to a consistent final ordering?
- Although the pairwise ranking risk model has proved to be successful in the score fusion problem, it has two limitations [20]:
  - It is not the best metric used in practical retrieval systems.
  - Only the top few positions of the ranked list are of importance.

  The DCG criterion addresses these limitations naturally as first, by definition, it weights correct results at the top more than those at the bottom and second, it constitutes the most widely used measure to evaluate the performance of a retrieval algorithm. Accordingly, one can consider a formulation of the score fusion problem in terms of the DCG criterion.

Notations

Density-Based 3D Shape Description

O	3D object (surface)
S	Local surface feature
$\mathcal{R}_S$	Range space of the feature S
$\{s_k\}_{k=1}^K$	Set of observations (sources) for S
$\overline{\mathcal{R}}_S$	Discretized version of $\mathcal{R}_S$
$\{t_n\}_{n=1}^N$	Target set (note that $\overline{\mathcal{R}}_S = \{t_n\}_{n=1}^N$)
$f_{S\|O} \triangleq f_S(\cdot\|O)$	Probability density function of S given O
$f_S(t_n\|O)$	Value of the probability density function of S at t_n given O
$\mathbf{f}_{S\|O} \triangleq [f_S(t_n\|O)]_{n=1}^N$	Descriptor vector of the object O w.r.t. S

Statistical Similarity Learning

x, x'	Generic database objects
q	Query object
$sim(\cdot,\cdot)$	Generic similarity function
$s_k \triangleq sim_k(x,q)$	kth similarity value between x and q
$\mathbf{s} \triangleq [s_k]_{k=1}^K$	K-vector of similarities
w_k	Weight for the kth similarity value
$\mathbf{w} \triangleq [w_k]_{k=1}^K$	K-vector of weights
$\varphi(x,q) \triangleq \langle \mathbf{w}, \mathbf{s} \rangle$	Final similarity (or scoring) function

Appendix A

3D Object Databases

In this appendix, we list the main features of 3D mesh databases used for research purposes.

- **Princeton Shape Benchmark (PSB)**
  - 1814 models in 161 classes
  - Split into two sets: Training (907 models in 90 classes) and Test (907 models in 92 classes)
  - 3-level classification hierarchy also available
  - Example classes: animal, building, furniture, household object, human, musical instrument, plant, vehicle
  - Low resolution, non-regular meshes
  - Public
  - http://shape.cs.princeton.edu/benchmark/
- **Sculpteur Database (SCU)**
  - 858 archaeologically valuable object models: 513 of them categorized in 53 classes
  - 2-level classification hierarchy also available
  - Example classes: amphora, vase, pavement, statue, relievo, mould
  - High resolution, watertight meshes
  - Private
  - http://www.sculpteurweb.org/
    http://www.tsi.enst.fr/3dmodels/
- **SHREC Watertight Database**
  - 400 models in 20 classes
  - Example classes: ant, chair, glasses, hand, human, octopus, spring
  - Topology-driven equivalence classes
  - High resolution, watertight meshes
  - Open to SHREC'07 participants

- http://watertight.ge.imati.cnr.it/

- **Purdue Engineering Shape Benchmark (ESB)**
  - 865 CAD models in 45 classes
  - 2-level classification hierarchy also available
  - Example classes: bearing, contact switch, elbow, gear, handle, housing, thin plate, T-shaped part, U-shaped part
  - Middle resolution, watertight meshes
  - Open to SHREC'07 participants
  - https://engineering.purdue.edu/PRECISE/shrec/
    http://shapelab.ecn.purdue.edu/Benchmark.aspx

- **Konstanz Database**
  - 1839 models: 473 classified
  - Public
  - http://merkur01.inf.uni-konstanz.de/CCCC/

- **MPEG-7 Database**
  - 227 models: all classified
  - Private
  - http://merkur01.inf.uni-konstanz.de/CCCC/

- **National Taiwan University Database (NTU)**
  - 10911 models
  - Public
  - http://3d.csie.ntu.edu.tw/ dynamic/database/index.html

- **Informatics-Telematics Institute Database (ITI)**
  - 544 models in 13 classes
  - Example classes: animal, airplane, helicopter, car, couch, human
  - Public
  - http://3d-search.iti.gr/3DSearch/index.html
    http://shrec.iti.gr/download.html

- **McGill Database**
  - Articulated object models
  - Public
  - http://www.cim.mcgill.ca/ shape/benchMark/

- **Carnegie Mellon Database**
  - ∼2000 models

  - Private
  - http://amp.ece.cmu.edu/projects/3DModelRetrieval/

- **SHREC 3D Face Database**
  - 1000 3D face models
  - Open to SHREC'07 participants
  - http://give-lab.cs.uu.nl/SHREC/shrec2007/

Appendix B

Standard Dissimilarity Measures

In this appendix, we provide definitions for some of the standard measures to evaluate the dissimilarity between two histogram-based descriptors $\mathbf{f} = [f_i]_{i=1}^{N}$ and $\mathbf{f}' = \left[f'_j\right]_{j=1}^{N'}$. Except for the earth mover's distance, we assume that descriptors have the same dimension, i.e., $N = N'$, their entries f_i and f'_j are real numbers and they sum to one, i.e., $\sum_i f_i = \sum_j f'_j = 1$. In the following definitions, the index i runs from 1 to N and j from 1 to N'.

B.1 L^p-Distances

For $p \geq 1$, these measures are also known as *Minkowski* distances, L^1, L^2 and L^∞ being the most commonly used ones, especially because they are computationally very cheap. For $p < 1$, L^p-measures are called as *sublinear* norms. They emphasize small pointwise absolute differences $|f_i - f'_i|$ more than Minkowski norms. On a general basis, as p decreases, small differences, typically in the tails of distributions become more and more important. L^p-distances for vectors are given by

$$L^p(\mathbf{f}, \mathbf{f}') = \left(\sum_i \left| f_i - f'_i \right|^p \right)^{(1/p)}.$$

L^1-distance, in particular, has an interesting connection with binary classification [7]. Suppose that an observation s (regarding a random variable $S \in \mathcal{R}_S$) comes from one of the two equally likely classes $\mathcal{C}$ and $\mathcal{C}'$ represented by densities f and f' respectively. The misclassification probability $Pr(error)$ of a Bayesian decision rule of the form: assign s to $\mathcal{C}$ if $f(s) \geq f'(s)$, and to $\mathcal{C}'$ otherwise, is given by

$$Pr(error) = \frac{1}{2} \int_{\mathcal{R}_S} min(f, f').$$

Using the identity $min(a, b) = \frac{|a+b|}{2} - \frac{|a-b|}{2}$, it can be shown that

$$\begin{aligned} Pr(error) &= \frac{1}{2} - \frac{1}{4} \int_{\mathcal{R}_S} |f - f'|, \\ &= \frac{1}{2} - \frac{1}{4} L^1(f, f'). \end{aligned}$$

Thus, the maximum Bayesian probability of misclassification occurs at the minimum L^1-distance between class-conditional densities f and f'.

B.2 Symmetric Kullback-Leibler Distance

In probability theory, the Kullback-Leibler divergence KL_{div} is a non-symmetric measure of difference between two probability distributions. A symmetric version of this measure can be obtained as follows

$$\begin{aligned} KL(\mathbf{f}, \mathbf{f}') &= KL_{div}(\mathbf{f}||\mathbf{f}') + KL_{div}(\mathbf{f}'||\mathbf{f}), \\ &= \sum_i f_i log\frac{f_i}{f_i'} + \sum_i f_i' log\frac{f_i'}{f_i}, \\ &= \sum_i (f_i - f_i') log\frac{f_i}{f_i'}. \end{aligned}$$

Note that KL_{div} is a special case of the more general α-divergence given by (in the continuous case)

$$\alpha_{div}(f||f') = \frac{1}{\alpha - 1} log \int f' \left(\frac{f}{f'}\right)^{\alpha},$$

where $\alpha \in [0, 1]$. KL_{div} can be obtained in the limit as $\alpha \to 1$.

B.3 χ^2-Divergence

In statistics, χ^2-test is a generic method to decide whether two data sets are drawn from the same distribution or not. The acceptance hypothesis is retained based on the significance of the χ^2-divergence value given below

$$\chi^2(\mathbf{f}, \mathbf{f}') = \sum_i \frac{(f_i - f_i')^2}{f_i + f_i'}.$$

In matching two descriptors, the above value is taken directly as a measure of dissimilarity without performing any significance test.

B.4 Bhattacharyya Distance

Bhattacharyya Distance is an alternative measure that can be used to decide whether two statistical distributions differ or not:.

$$B(\mathbf{f}, \mathbf{f}') = 1 - \sum_i \sqrt{f_i f_i'}.$$

In view of the discussion for the L^1-distance, Bhattacharyya distance can be related to the Bayesian probability of misclassification. Consider the binary classification setup as in the above. Accordingly, from (B.1) and the fact that $min(a, b) \leq \sqrt{ab}$, we have

$$Pr(error) \leq \int_{\mathcal{R}_S} \sqrt{f f'}.$$

Thus, an *upper bound* on the Bayesian probability of misclassification is maximized at the minimum Bhattacharyya distance between class-conditional densities f and f'.

B.5 Histogram Intersection

Histogram intersection is originally proposed for matching color histograms [67]. The associated distance function is expressed as

$$HI(\mathbf{f}, \mathbf{f}') = 1 - \sum_i min(f_i, f'_i).$$

By invoking $min(a, b) = \frac{|a+b|}{2} - \frac{|a-b|}{2}$, we can see that HI is equivalent to L^1-distance.

B.6 Earth Mover's Distance

A descriptor entry f_i can be a two-tuple (m_i, w_i), where $m_i \in \mathbb{R}^m$ is interpreted as a bin center in a multidimensional histogram and $w_i \in \mathbb{R}$ as the amount of mass concentrated around m_i. In such a case, descriptors $\mathbf{f} = [(m_i, w_i)]_{i=1}^{N}$ and $\mathbf{f}' = \left[(m'_j, w'_j)\right]_{j=1}^{N'}$ are called as signatures. Let W be the total amount of mass contained in the descriptor $\mathbf{f}$, that is, $W = \sum_i w_i$ (likewise for W'), and let η be a ground distance between bin centers such as L^2. The earth mover's distance is then defined as the minimum value of an optimization problem [102]:

$$EMD(\mathbf{f}, \mathbf{f}') = \min_{c_{ij}} \frac{\sum_{i,j} \eta(m_i, m'_j) c_{ij}}{min\,(W, W')},$$

$$\text{such that}$$

$$c_{ij} \geq 0, \forall i, j,$$

$$\sum_j c_{ij} \leq w_i, \forall i,$$

$$\sum_i c_{ij} \leq w'_j, \forall j,$$

$$\sum_{i,j} c_{ij} = min\left(W, W'\right),$$

where the optimization variables c_{ij} stand for the amount of mass to be carried from the bin m_i to the bin m'_j. The use of EMD is motivated by the fact that the bin centers in two different descriptors might not be aligned. The computational complexity of the above program is above $O(N^3)$ and limits its practical usability for large signatures. A recently introduced algorithm reduces this complexity down to an empirically shown $O(N^2)$ when the ground distance η is chosen as L^1 [103].

Appendix C

KDE and Related Issues

In this appendix, we provide some derivations related to kernel density estimation (KDE). First, in C.1, we derive the univariate KDE using a fixed scalar bandwidth as in [13]. In C.2, we derive the upper bound on the mean integrated absolute error (*MIAE*) in the univariate shape description context, which were used to illustrate the smoothing effect of the KDE in Section 2.2.1. This derivation is our original contribution. In C.3, following [7], we recapitulate the asymptotic mean integrated squared error (*AMISE*) and extend the expression for the univariate Scott bandwidth to the case of arbitrarily weighted observations.

C.1 Derivation

Suppose we want to estimate the pdf f_S of some random variate $S \in \mathbb{R}^m$ at some target $t \in \mathbb{R}^m$. Let $\mathcal{N}(t)$ be some neighborhood around the target t, with volume $Vol[\mathcal{N}_(t)]$. The probability that S falls inside $\mathcal{N}(t)$ is

$$Pr\{S \in \mathcal{N}_t\} = \int_{\mathcal{N}(t)} f_S(u)du.$$

For sufficiently small $\mathcal{N}(t)$ and continuous f_S, a reasonable approximation to $f_S(t)$ would then be

$$f_S(t) \simeq \frac{Pr\{S \in \mathcal{N}(t)\}}{Vol[\mathcal{N}(t)]}. \tag{C.1}$$

Given a set of independent observations $\{s_k\}_{k=1}^K$, a natural estimate for $Pr\{S \in \mathcal{N}_t\}$ can be obtained by counting the number of s_k's falling inside $\mathcal{N}(t)$ in which case we can write

$$Pr\{S \in \mathcal{N}_t\} \simeq \frac{\#\{s_k\text{'s inside } \mathcal{N}(t)\}}{K} = \frac{1}{K}\sum_{k=1}^{K} \mathbb{I}\{s_k \in \mathcal{N}(t)\}, \tag{C.2}$$

where $\mathbb{I}\{\cdot\}$ is one when the predicate inside the curly brackets is true, and zero otherwise. Now suppose momentarily that the neighborhood $\mathcal{N}(t)$ is an m-dimensional hypercube with edge length h so that $Vol[\mathcal{N}_(t)] = h^m$. Accordingly, combining Eqs. C.1 and C.2, we can write

$$f_S(t) \simeq \frac{1}{Kh^m}\sum_{k=1}^{K} \mathbb{I}\{s_k \in \mathcal{N}(t)\}. \tag{C.3}$$

The kernel family of density estimators arises in a quest to generalize Eq. C.3. Observe that, when the neighborhood is chosen to be a hypercube, we can analytically express the 0-1 counting function $\mathbb{I}\{\cdot\}$ as

$$\mathbb{I}\{s_k \in \mathcal{N}(t)\} = \mathcal{K}\Big(\frac{I(t-s_k)}{h}\Big),$$

where $\mathcal{K}$ is interpreted as a kernel function and I is the $m \times m$ identity matrix. Thus, the 0-1 counting function occurs as just a special case of a generic kernel $\mathcal{K}$, corresponding to a hypercube neighborhood of fixed volume. One can also identify $h^{-1}I$ as the bandwidth parameter matrix H, which remains the same for all the observations, i.e., $H_k = H, \forall k$.

Allowing kernels that are smoother than the 0-1 counting function lies at the heart of KDE methodology. The specific form of the kernel paramaterized by the bandwidth H determines implicitly the shape and the size of the neighborhood chosen. In fact, one does not need to define a crisp neighborhood such as a hypercube. When a smooth function of the observation-target distance is chosen as a kernel, all the observations contribute to the density estimate. In a nutshell, closer observations contribute more, far-away ones less. As a result, KDE can also be viewed as a data interpolation method. Thus, the KDE equation for arbitrary kernels reads as

$$f_S(t) = \frac{1}{K\,|H|}\sum_{k=1}^{K}\mathcal{K}\left(H_k^{-1}(t-s_k)\right). \tag{C.4}$$

For the Gaussian kernel given by

$$\mathcal{K}_{Gaussian}(u) = (2\pi)^{-m/2}exp\Big(-\frac{1}{2}\left\|u\right\|^2\Big),$$

we have

$$f_S(t) = \frac{1}{(2\pi)^{-m/2}K\,|H|}\sum_{k=1}^{K}exp\Big(-\frac{1}{2}(t-s_k)^T H^{-2}(t-s_k)\Big). \tag{C.5}$$

Eqs. 2.4 and 2.5 in Section 2.2 are more general forms of Eqs. C.4 and C.5, where each observation induces a differently sized of neighborhood around the target t through the variable bandwidth parameter H_k. Another distinction is the scalar importance weight w_k associated with each observation, which is set to $1/K$ above, in the abscense of any prior information regarding the sampling density.

C.2 Derivation of the Upper Bound on MIAE

Let f and f' be two real-valued functions with scalar domains, given by

$$\begin{aligned} f(t) &= \frac{1}{\sqrt{2\pi}hK}\sum_{k=1}^{K}exp\left\{-\frac{1}{2}\left(\frac{t-s_k}{h}\right)^2\right\}, \\ f'(t) &= \frac{1}{\sqrt{2\pi}hK}\sum_{k=1}^{K}exp\left\{-\frac{1}{2}\left(\frac{t-s'_k}{h}\right)^2\right\}, \end{aligned}$$

where s_k's are known deterministic scalars, s'_k's are their randomly perturbed versions so that $s'_k = s_k + \epsilon_k$. The additive perturbations ϵ_k's are zero-mean i.i.d. random variables.

Both f and f' are positive and integrate to one. In the following, we derive an upper bound on the expected variation between f and f' in the integrated absolute sense. We call this measure of variation as mean integrated absolute error ($MIAE$). Concretely,

$$\begin{aligned} MIAE(f,f') &\triangleq E\left\{\int_{\mathbb{R}}|f-f'|\right\}, \\ &= \int_{\mathbb{R}} E|f-f'|, \end{aligned}$$

where $E\{\cdot\}$ is the expectation operator. We can write the point-wise difference $f(t)-f'(t)$ as

$$f(t)-f'(t)=\frac{1}{\sqrt{2\pi}hK}\sum_{k=1}^{K} exp\left\{-\frac{1}{2}\left(\frac{t-s_k}{h}\right)^2\right\}\left[1-exp\left\{\frac{(t-s_k)\epsilon_k}{h^2}-\frac{1}{2}\left(\frac{\epsilon_k}{h}\right)^2\right\}\right].$$

Let us now use the Hermite identity $exp(2ab-b^2)=\sum_{n=0}^{\infty}\frac{1}{n!}b^nH_n(a)$, where H_n are the Hermite polynomials. Identifying that $a=(t-s_k)/\sqrt{2}h$ and $b=\epsilon_k/\sqrt{2}h$, we get

$$f(t)-f'(t)=\frac{1}{\sqrt{2\pi}hK}\sum_{k=1}^{K}\sum_{n=1}^{\infty} exp\left\{-\frac{1}{2}\left(\frac{t-s_k}{h}\right)^2\right\}\frac{1}{n!}\left(\frac{\epsilon_k}{\sqrt{2}h}\right)^n H_n\left(\frac{t-s_k}{\sqrt{2}h}\right).$$

Taking the expected absolute value of the above expression and using the triangle inequality, we obtain

$$E\left|f(t)-f'(t)\right|\leq\frac{1}{\sqrt{2\pi}hK}\sum_{n=1}^{\infty}\frac{1}{2^{n/2}n!}\frac{E|\epsilon|^n}{h^n}\sum_{k=1}^{K} exp\left\{-\frac{1}{2}\left(\frac{t-s_k}{h}\right)^2\right\}\left|H_n\left(\frac{t-s_k}{\sqrt{2}h}\right)\right|,$$

where we have used the fact that $E|\epsilon_k|=E|\epsilon|,\forall k$ by the i.i.d. assumption. Now, integrating both sides of the above expression with respect to t, we obtain an upper bound on $MIAE$,

$$MIAE(f,f')\leq\sum_{n=1}^{\infty}\frac{C_n(h)}{2^{n/2}n!}\frac{E|\epsilon|^n}{h^n}, \tag{C.6}$$

where $C_n(h)\triangleq\frac{1}{\sqrt{2\pi}h}\int e^{-u^2/2h^2}|H_n(\frac{u}{\sqrt{2}h})|du$. Furthermore, Cramer's inequality for Hermite polynomials tells that $|H_n(u)|\leq C2^{n/2}\sqrt{n!}\ e^{u^2/2}$, where C is a numerical constant less than 1.09 in value [8]. Thus,

$$\begin{aligned} C_n(h) &\leq \left(C2^{n/2}\sqrt{n!}\right)\frac{1}{\sqrt{2\pi}h}\int exp\left(-\frac{u^2}{2h^2}\right)exp\left(\frac{u^2}{4h^2}\right)du, \\ &= \left(C2^{n/2}\sqrt{n!}\right)\frac{\sqrt{2}}{\sqrt{2\pi}\sqrt{2}h}\int exp\left(-\frac{1}{2}\frac{u^2}{2h^2}\right)du, \\ &= C2^{(n+1)/2}\sqrt{n!}\ . \end{aligned}$$

Finally, combining the above result with the bound in (C.6), we get

$$MIAE(f,f')\leq C\sum_{n=1}^{\infty}\frac{1}{\sqrt{n!}}\frac{E|\epsilon|^n}{h^n}, \tag{C.7}$$

where we have lumped the $\sqrt{2}$ factor into the constant C.

C.3 AMISE and the Scott Bandwidth

The asymptotic mean integrated squared error ($AMISE$) of a univariate kernel estimator is given by the sum of the asymptotic integrated squared bias ($AISB$) and the asymptotic integrated variance (AIV):

$$AMISE(h) = AISB(h) + AIV(h).$$

When the observations are equally weighted by $1/K$, we have [7]:

$$\begin{aligned} AISB(h) &= \frac{1}{4} \left\| \mu_2(\mathcal{K}) \right\|^2 R(f'') h^4, \\ AIV(h) &= \frac{R(\mathcal{K})}{Kh}, \end{aligned}$$

where $R(\phi) \triangleq \int \phi^2$ and $\left\| \mu_2(\mathcal{K}) \right\| \triangleq \int u^2 \mathcal{K}(u) du$. Observe that $AISB$ does not depend on the number of observations K neither on how they are weighted. Thus, to derive an expression for the $AMISE$-optimal bandwidth for arbitrarily weighted observations by w_k, where $\sum_k w_k = 1$, we need to work with the variance term as follows. The univariate kernel estimator $\hat{f}$ of a density f is given by

$$\hat{f}(s) = \frac{1}{h} \sum_{k=1}^{K} w_k \mathcal{K}_h \left(s, s_k \right),$$

where $\mathcal{K}_h(s, s_k) \triangleq \frac{1}{h} \mathcal{K}\left(\frac{s - s_k}{h} \right)$ and $\{s_k\}_{k=1}^K$ are i.i.d. realizations of a random variable S. The variance of the estimate $\hat{f}(s)$ is given by

$$\begin{aligned} Var\left\{ \hat{f}(s) \right\} &= W \; Var\left\{ \mathcal{K}_h(s, S) \right\}, \\ &= W \; E\left\{ \mathcal{K}_h(s, S) \right\}^2 - W \; \left[E\left\{ \mathcal{K}_h(s, S) \right\} \right]^2, \end{aligned}$$

where $W \triangleq \sum_{k=1}^K w_k^2$. Expanding the first and the second terms above in Taylor series and then integrating the result, the integrated variance IV can be shown to be [7]:

$$IV(h) = W \; \frac{R(\mathcal{K})}{h} - W \; R(f) + \ldots.$$

Finally, based on Theorem 6.1, page 131 in [7], we get

$$\begin{aligned} AMISE_w(h) &= W \; \frac{R(\mathcal{K})}{h} + \frac{1}{4} \left\| \mu_2(\mathcal{K}) \right\|^2 R(f'') h^4, \\ h^* &= \left[W \; \frac{R(\mathcal{K})}{\left\| \mu_2(\mathcal{K}) \right\|^2 R(f'')} \right]^{1/5}, \end{aligned}$$

for the univariate kernel estimator using arbitrarily weighted observations.

To find the Normal reference rule or the Scott bandwidth, we proceed by assuming a Gaussian reference density f with variance σ^2 and a Gaussian kernel $\mathcal{K}$. Plugging in the corresponding expressions for $R(f'')$, $R(\mathcal{K})$, and $\left\| \mu_2(\mathcal{K}) \right\|^2$, we obtain the Scott bandwidth:

$$h_{Scott} = \left(\frac{4}{3} W \right)^{1/5} \sigma.$$

The m-dimensional multivariate generalization of this result for a diagonal bandwidth matrix $H = diag(h_1, \ldots, h_m)$ can be obtained by working with multidimensional Taylor expansions [7]. Here, we simply state the result:

$$H_{Scott} = \left(\frac{4}{m+2}W\right)^{1/(m+4)} diag(\sigma_1, \ldots, \sigma_m).$$

Appendix D

Marginalization Results

Starting from the pdf of an m-dimensional feature $S = (S_1, \ldots, S_m)$, the marginalization analysis is carried in $m-1$ stages. At a certain stage $t = 1, \ldots, m-1$, we integrate t-tuple of components out from the pdf of the original m-dimensional feature. At a given stage t, there are $\binom{m}{t}$ such operations, yielding $2^m - 2$ possible marginalizations in total.

The following tables present the DCG results of the exhaustive marginalization analysis applied on the pdf-based descriptor of the 6-dimensional feature $(R, \hat{R}_x, \hat{R}_y, \hat{N}_x, \hat{N}_y, A)$, as presented in Section 4.6.2. The analysis involves 5 stages, thus the total number of marginalizations is $2^m - 2 = 62$. In these tables, a given row corresponds to a specific marginalization where the retained components are indicated by gray-shaded cells and the removed one(s) by white cell(s). The black cell at the right-most column stands for the feature configuration, which yields the maximum DCG of the corresponding stage.

Table D.1: DCG (%) Results of One-Component Marginalization (the white cell in a given row indicates the component removed)

R	$\hat{R}_x$	$\hat{R}_y$	$\hat{N}_x$	$\hat{N}_y$	A	**DCG**	**Max**
						59.9	
						62.6	
						62.0	
						62.2	
						61.6	
						60.6	

Table D.2: DCG (%) Results of Two-Component Marginalization (the white cells in a given row indicate the two components removed)

R	$\hat{R}_x$	$\hat{R}_y$	$\hat{N}_x$	$\hat{N}_y$	A	**DCG**	**Max**
						60.0	
						59.3	
						58.2	
						59.8	
						63.4	
						62.1	
						59.7	
						62.0	
						62.0	
						59.8	
						58.2	
						60.8	
						60.1	
						60.6	
						59.8	

Table D.3: DCG (%) Results of Three-Component Marginalization (the white cells in a given row indicate the three components removed)

R	$\hat{R}_x$	$\hat{R}_y$	$\hat{N}_x$	$\hat{N}_y$	A	**DCG**	**Max**
■	■	■				49.7	
■	■		■			60.8	
■	■			■		59.2	
■	■				■	58.1	
■		■	■			59.5	
■		■		■		59.8	
■		■			■	57.2	
■			■	■		54.9	
■			■		■	60.5	
■				■	■	59.7	
	■	■				57.7	
	■	■		■		57.1	
	■	■			■	53.0	
	■		■	■		57.7	
	■		■		■	61.5	■
	■			■	■	60.0	
		■	■	■		57.1	
		■	■		■	59.9	
		■		■	■	59.5	
			■	■	■	55.1	

Table D.4: DCG (%) Results of Four-Component Marginalization (the white cells in a given row indicate the four components removed)

R	$\hat{R}_x$	$\hat{R}_y$	$\hat{N}_x$	$\hat{N}_y$	A	**DCG**	**Max**
						49.1	
						49.1	
						54.9	
						53.9	
						50.1	
						42.1	
						58.1	
						56.2	
						52.5	
						56.2	
						56.2	
						50.9	
						46.1	
						55.0	
						53.8	

Table D.5: DCG (%) Results of Five-Component Marginalization (the white cells in a given row indicate the five components removed)

R	$\hat{R}_x$	$\hat{R}_y$	$\hat{N}_x$	$\hat{N}_y$	A	**DCG**	**Max**
						37.7	
						39.9	
						40.5	
						44.8	
						42.8	
						38.5	

Appendix E

Sample Two-round Searches

In this appendix, we illustrate the two-round application of learning-based score fusion. We remind the *on-line* and *off-line* versions of this protocol below:

- **On-line.**
  - The user enters a query shape.
  - In the first round, the system returns M ranked databases shapes using the basic *SUM* rule and asks the user to label *all of them* as either *relevant* or *non-relevant*.
  - The system learns a weight vector using the score difference vectors corresponding to these marked shapes *on-line*.
  - In the second round, the system uses the learned weight vector to evaluate a weighted similarity score and returns a new set of database shapes.
- **Off-line.**
  - The user enters a query shape.
  - In the first round, the system returns M ranked databases shapes using the basic *SUM* rule and asks the user to label *just one shape*, the very first one that he/she finds *relevant*.
  - In the second round, the system uses the weight vector corresponding to the marked shape, which has already been learned *off-line*, to evaluate a weighted similarity score and returns a new set of database shapes.

Figures E.1 and E.2 illustrate two applications of the on-line mode using a “couch" and a “human" 3D model as queries respectively. Figures E.3 and E.4 illustrate two applications of the off-line mode using a “plant" and a “bench" 3D model as queries respectively.

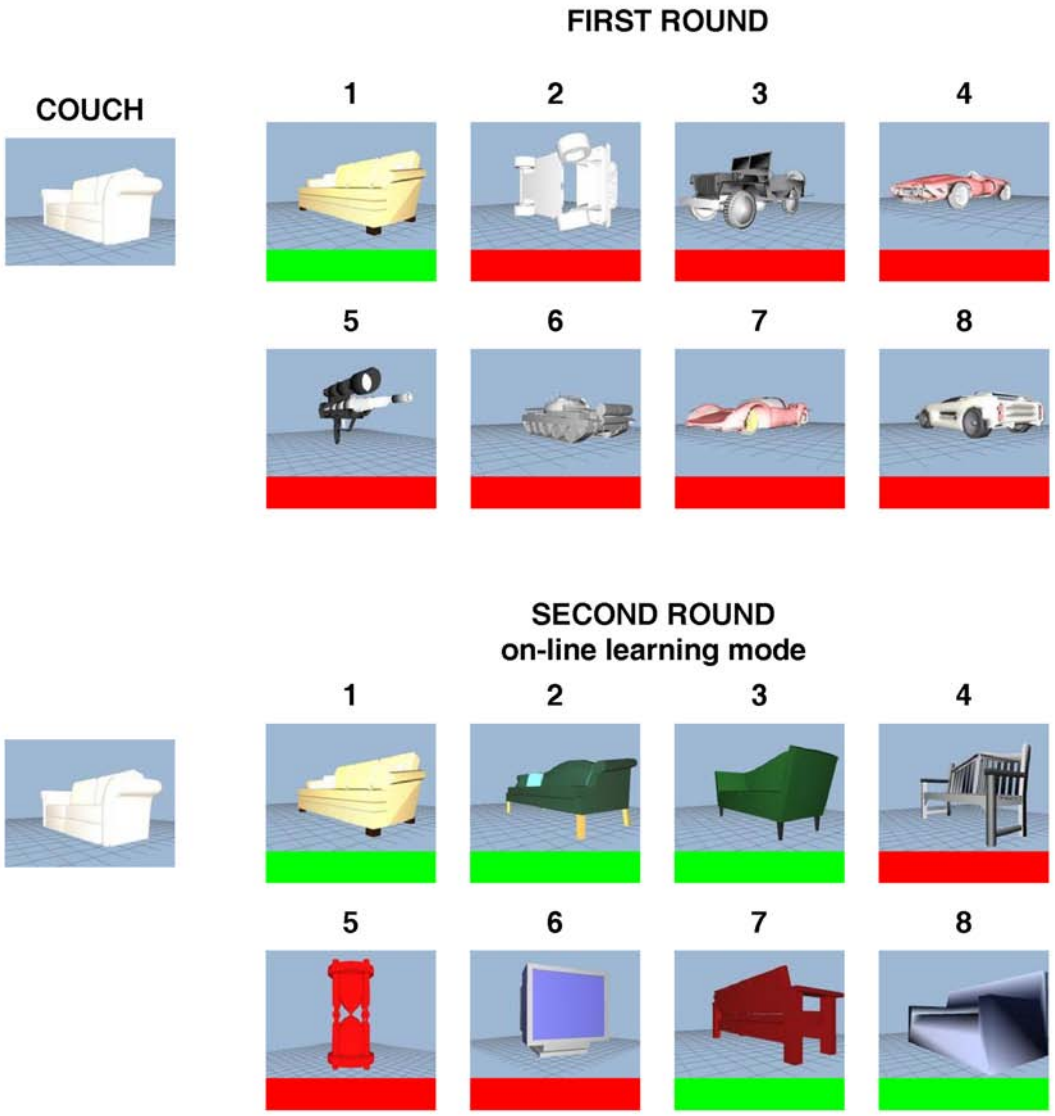

Figure E.1: Two-round search with *on-line* learning on a “couch" query. In the first round, only the first-ranked item is relevant (marked as green, the remainder as red); the second round retrieves four more “couch" models. Notice also that the fourth-ranked “bench" model is plausible to human intuition but not used as a positive result in performance evaluation.

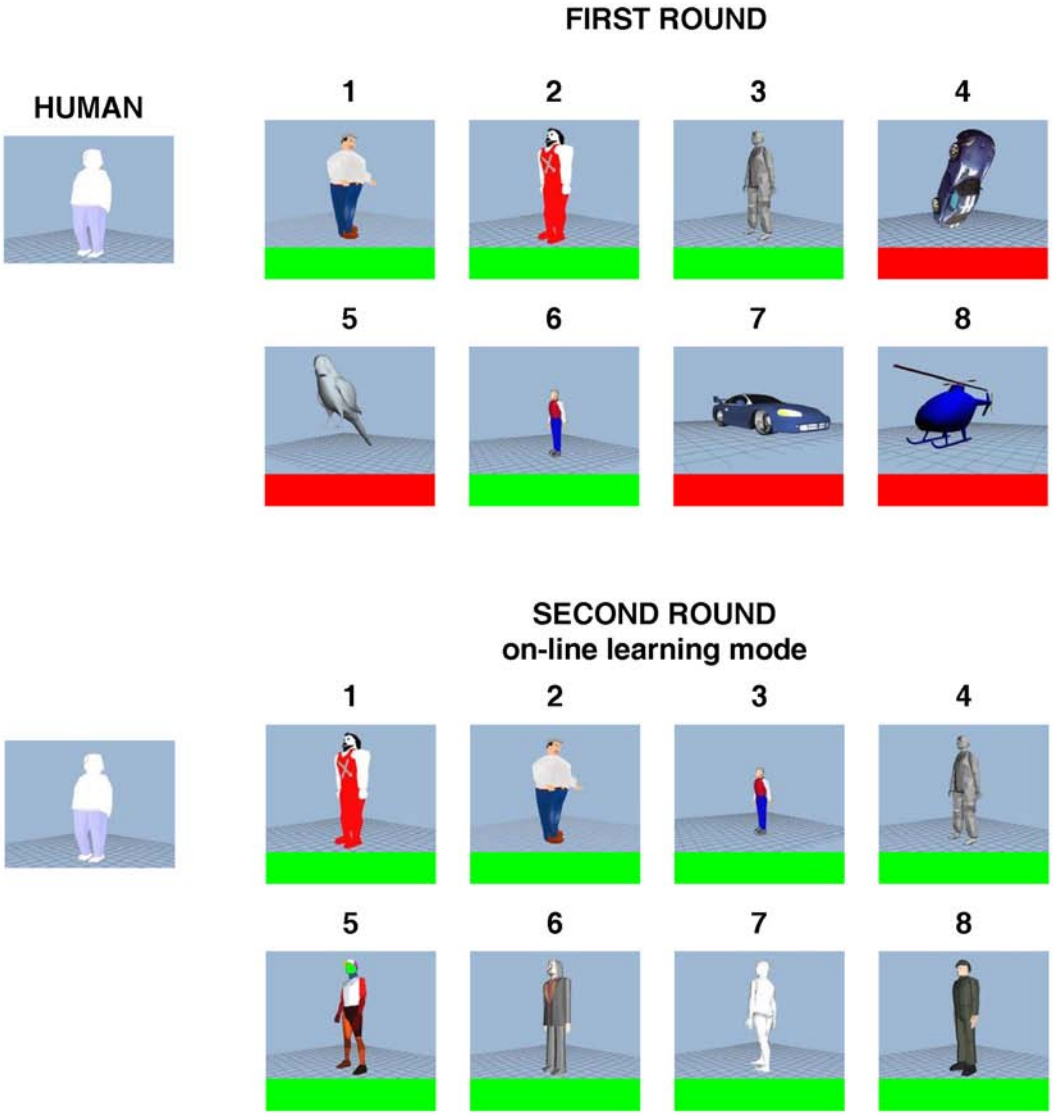

Figure E.2: Two-round search with *on-line* learning on a “human" query. In the first round, the system returns non-relevant “car", “bird" and “helicopter" models (marked as red) in addition to other “human" models (marked as green); the secound round eliminates the non-relevant ones and returns only “human" models.

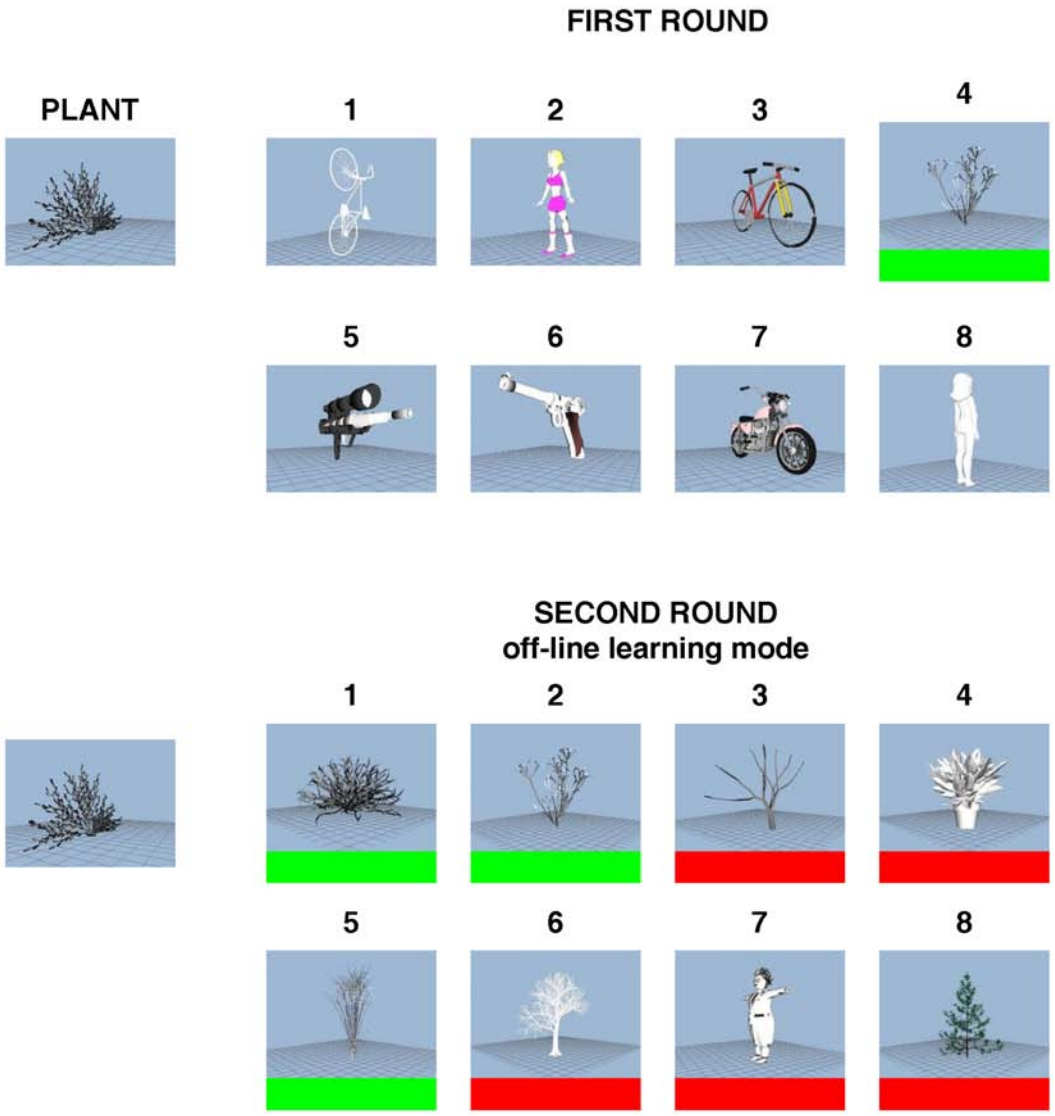

Figure E.3: Two-round search with *off-line* learning on a “plant" query. In the first round, the very-first relevant item occurs only at the fourth position (marked as green); the second round retrieves two more crisply correct matches along with three “tree" models (positions: 3, 6 and 8) and one “potted-plant", which are also plausible (but not used as positive results for performance evaluation). The only totally non-relevant item of the second round is the seventh-ranked “human" model.

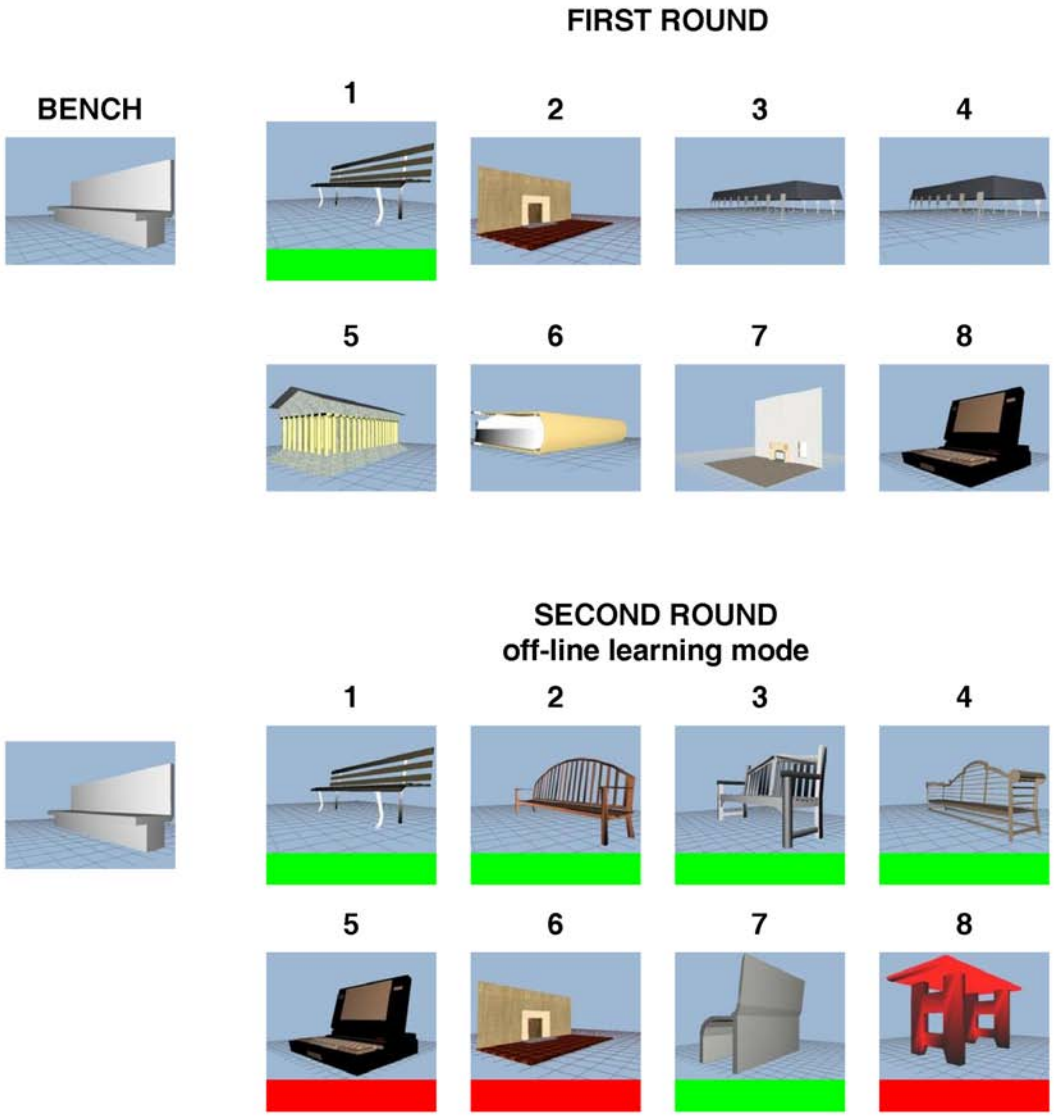

Figure E.4: Two-round search with *off-line* learning on a “bench" query. In the first round, the first item is marked as relevant (in green); the second round returns four more “bench" models at positions 2, 3 and 7 respectively. The eight-ranked “desk" is semantically similar to the query as well.

Appendix F

Publications

F.1 Publications Related to the Thesis

1. **C. B. Akgül**, B. Sankur, Y. Yemez, F. Schmitt. Density-Based 3D Shape Descriptors. *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 32503, 2007.

2. **C. B. Akgül**, B. Sankur, F. Schmitt, Y. Yemez. Multivariate Density-Based 3D Shape Descriptors. *Proceedings of the Shape Modeling International 2007 (SMI'07)*, Lyon, France, June 13-15, 2007.

3. **C. B. Akgül**, F. Schmitt, B. Sankur, Y. Yemez. Multivariate Density-Based 3D Shape Descriptors. *Proceedings of the Shape Retrieval Context 2007 (SHREC'07)*, Lyon, France, June 15, 2007.

4. **C. B. Akgül**, B. Sankur, Y. Yemez, F. Schmitt. Improving Efficiency of Density-Based Shape Descriptors for 3D Object Retrieval. *Computer Vision / Computer Graphics Collaboration Techniques (MIRAGEŠ07)*, Springer LNCS Series, vol. 4418, pp. 330-340, INRIA Rocquencourt, France, March 28-30, 2007.

5. **C. B. Akgül**, B. Sankur, F. Schmitt, Y. Yemez. Density-based Shape Descriptors for 3D Object Retrieval. *International Workshop on Multimedia Content Representation, Classification and Security (MRCS'06)*, Springer LNCS Series, vol. 4105, pp. 322-329, Istanbul, Turkey, September 11-13, 2006.

6. **C. B. Akgül**, B. Sankur, Y. Yemez, F. Schmitt. A Framework for Histogram-Induced 3D Descriptors. *Proceedings of the 14th European Signal Processing Conference (EUSIPCO'06)*, Florence, Italy, September 4-8, 2006.

7. **C. B. Akgül**, B. Sankur, F. Schmitt, Y. Yemez. Feature-Level and Descriptor-Level Information Fusion for Density-Based 3D Shape Descriptors. *IEEE 15th Signal Processing and Communications Applications (SIU)*, Eskisehir, Turkey, June 2007.

8. **C. B. Akgül**, B. Sankur, F. Schmitt, Y. Yemez. A New Framework for 3D Shape Descriptors. *IEEE 14th Signal Processing and Communications Applications (SIU)*, Antalya, Turkey, April 2006.

9. **C. B. Akgül**, B. Sankur, F. Schmitt, Y. Yemez. 3D Object Matching via Multivariate Shape Distributions. *IEEE 13th Signal Processing and Communications Applications (SIU)*, Kayseri, Turkey, May 2005.

In Preparation

1. Score Fusion by Ranking Risk Minimization for 3D Object Retrieval, 2007.
2. 3D Pose Normalization using Density-Based Shape Descriptors, 2007.

F.2 Other Publications in 2004-2007

1. **C. B. Akgül**, B. Sankur, A. Akin. Extraction of cognitive activity related waveforms from functional near infrared spectroscopy signals. *Medical and Biological Engineering and Computing*, vol. 44, pp. 945-958, November 2006.
2. **C. B. Akgül**, B. Sankur, A. Akin. Spectral analysis of event-related hemodynamic responses in functional near infrared spectroscopy. *Journal of Computational Neuroscience*, vol. 18, pp. 67-83, January 2005.
3. **C. B. Akgül**, B. Sankur, A. Akin. Extraction of cognitive activity related waveforms from functional near infrared signals. *Proceedings of the 14th European Signal Processing Conference (EUSIPCO'06)*, Florence, Italy, September 4-8, 2006.
4. **C. B. Akgül**, B. Sankur, A. Akin. Evidence of cognitive activity in functional near infrared spectroscopy signal. *OSA Optical Techniques in Neuroscience Topical Meeting*, Miami Beach, Florida, USA, April 2004.
5. D. Lesage, J. Darbon, **C. B. Akgül**. An Efficient Algorithm for Connected Attribute Thinnings and Thickenings. *International Symposium on Visual Computing (ISVC'06)*, Springer LNCS Series, vol. 4292, pp. 393-404, Lake Tahoe, Nevada, USA, November 6-8, 2006.
6. J. Darbon, **C. B. Akgül**. An Efficient Algorithm for Attribute Openings and Closings. *Proceedings of the 13th European Signal Processing Conference (EUSIPCO'05)*, Antalya, Turkey, September 2005.
7. **C. B. Akgül**, B. Sankur, A. Akin. Evidence of cognitive activity in functional optical signals. *IEEE 12th Signal Processing and Communications Applications (SIU)*, Cesme, Turkey, April 2004.
8. **C. B. Akgül**, B. Sankur, A. Akin. Extraction of cognitive activity related waveforms from functional optical signals. *IEEE 12th Signal Processing and Communications Applications (SIU)*, Cesme, Turkey, April 2004.
9. E. Yörük, **C. B. Akgül**. Color image segmentation using fast watersheds and PDE-based regularization. *IEEE 12th Signal Processing and Communications Applications (SIU)*, Cesme, Turkey, April 2004.

Bibliography

[1] S. Ortiz. 3D searching starts to take shape. *Computer*, 37(8):24–26, 2004.

[2] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranic. Feature-based similarity search in 3D object databases. *ACM Comput. Surv.*, 37(4):345–387, 2005.

[3] J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. In *Proc. of the Shape Modeling International 2004 (SMI'04)*, pages 145–156, Genoa, Italy, 2004.

[4] T. Funkhouser and P. Shilane. Partial matching of 3D shapes with priority-driven search. In *Symposium on Geometry Processing*, June 2006.

[5] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37(5):509–530, April 2005.

[6] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer Series in Statistics. Springer, 2004.

[7] D. W. Scott. *Multivariate Density Estimation. Theory, Practice and Visualization.* Wiley, New York, 1992.

[8] L. Greengard and J. Strain. The fast Gauss transform. *SIAM J. Sci. Statist. Comput.*, 12:79–94, 1991.

[9] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. *ICCV*, 1:464, 2003.

[10] S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and scoring using empirical risk minimization. In *COLT*, pages 1–15, 2005.

[11] S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and empirical risk minimization of U-statistics. *The Annals of Statistics*, to appear, 2007.

[12] R. J. Campbell and P. J. Flynn. A survey of free-form object representation and recognition techniques. *Comput. Vis. Image Underst.*, 81(2):166–210, 2001.

[13] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-Interscience, 2000.

[14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, New York, NY, 2001.

[15] D. G. Kendall. The diffusion of shape. *Adv. Appl. Probab.*, 9:428–430, 1977.

[16] M. Kazhdan, T. Funkhouser, and S. 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, Aire-la-Ville, Switzerland, 2003. Eurographics Association.

[17] D. V. Vranić. *3D Model Retrieval.* PhD thesis, University of Leipzig, 2004.

[18] E. Paquet, A. Murching, T. Naveen, A. Tabatabai, and M. Rioux. Description of shape information for 2D and 3D objects. *Signal Processing: Image Communication*, 16:103–122, 2000.

[19] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser. A planar-reflective symmetry transform for 3D shapes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3), July 2006.

[20] D. Cossock and T. Zhang. Statistical analysis of Bayes optimal subset ranking. Technical report, Yahoo! Research, 2006.

[21] N. Vayatis. Is there life beyond the classification problem? In *Mathematical Foundations of Learning Theory - II*, Paris, France, June 2006.

[22] T. Zhang. Theory and algorithms for large scaling ranking problems. In *Mathematical Foundations of Learning Theory - II*, Paris, France, June 2006.

[23] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Proc. of the Shape Modeling International 2004 (SMI'04)*, pages 167–178, Genoa, Italy, 2004.

[24] S. Goodall, P. H. Lewis, K. Martinez, P. A. S. Sinclair, F. Giorgini, M. Addis, M. J. Boniface, C. Lahanier, and J. Stevenson. SCULPTEUR: Multimedia retrieval for museums. In *Proc. of the Image and Video Retrieval: Third International Conference (CIVR'04)*, pages 638–646, 2004.

[25] T. Tung. *Indexation 3D de bases de données d'objets par graphes de Reeb améliorés.* PhD thesis, Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France, 2005.

[26] D. Giorgi, S. Biasotti, and L. Paraboschi. Shape retrieval contest 2007: Watertight models track. In R. C. Veltkamp and F. B. ter Haar, editors, *SHREC2007: 3D Shape Retrieval Contest, Techical Report UU-CS-2007-015*, pages 5–10. June 2007.

[27] S. Jayanti, K. Kalyanaraman, N. Iyer, and K. Ramani. Developing an engineering shape benchmark for CAD models. *Computer-Aided Design*, 38(9):939–953, September 2006.

[28] R. C. Veltkamp and F. B. ter Haar. SHREC2007, 3D shape retrieval contest. Technical Report UU-CS-2007-015, Department of Information and Computing Sciences, Utrecht University, 2007.

[29] Y. Kalyanaraman and K. Ramani. Shape retrieval contest 2007: CAD models track. In R. C. Veltkamp and F. B. ter Haar, editors, *SHREC2007: 3D Shape Retrieval Contest, Techical Report UU-CS-2007-015*, pages 11–12. June 2007.

[30] E. Paquet and M. Rioux. Nefertiti: a query by content software for three-dimensional models databases management. In *Proc. of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling (NRC'97)*, page 345, Washington, DC, USA, 1997. IEEE Computer Society.

[31] E. Paquet and M. Rioux. Nefertiti: A tool for 3D shape databases management. *SAE Transactions: Journal of Aerospace*, 108:387–393, 2000.

[32] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, 2002.

[33] Y. Liu, H. Zha, and H. Qin. The generalized shape distributions for shape matching and analysis. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications (SMI'06)*, Matsushima, Japan, June 2006.

[34] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *Proc. of the 6th International Symposium on Advances in Spatial Databases (SSD'99)*, pages 207–226. Springer-Verlag, 1999.

[35] B. K. P. Horn. Extended Gaussian images. *Proc. of the IEEE*, 72:1671–1686, 1984.

[36] S. B. Kang and K. Ikeuchi. The complex EGI: A new representation for 3D pose determination. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 15(7):707–721, 1993.

[37] T. Zaharia and F. Prêteux. Three-dimensional shape-based retrieval within the MPEG-7 framework. In *Proceedings of the SPIE Conference 4304 on Nonlinear Image Processing and Pattern Analysis XII, San Jose, CA*, pages 133–145, January 2001.

[38] T. Zaharia and F. Prêteux. Shape-based retrieval of 3D mesh models. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME 2002)*, Lausanne, Switzerland, August 2002.

[39] D. V. Vranić and D. Saupe. 3D shape descriptor based on 3D Fourier transform. In *Proc. of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS'01)*, Budapest, Hungary, September 2001.

[40] H. Dutağacı, B. Sankur, and Y. Yemez. Transform-based methods for indexing and retrieval of 3D objects. *3DIM*, 00:188–195, 2005.

[41] J. Ricard, D. Coeurjolly, and A. Baskurt. Generalizations of angular radial transform for 2d and 3d shape retrieval. *Pattern Recognition Letters*, 26(14):2174–2186, 2005.

[42] D. Saupe and D. V. Vranić. 3D model retrieval with spherical harmonics and moments. In *Proceedings of the DAGM 2001*, Munich, Germany, September 2001.

[43] D. V. Vranić. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2003)*, pages 757–760, Barcelona, Spain, September 2003.

[44] D. V. Vranić and D. Saupe. Description of 3D-shape using a complex function on the sphere. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME 2002)*, pages 177–180, Lausanne, Switzerland, August 2002.

[45] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3D models. *ACM Trans. Graph.*, 22(1):83–105, 2003.

[46] H. Laga, H. Takahashi, and M. Nakajima. Spherical wavelet descriptors for content-based 3d model retrieval. In *SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 15–25, Washington, DC, USA, 2006. IEEE Computer Society.

[47] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *ACM SIGGRAPH*, pages 203–212, Los Angeles, CA, USA, August 2001.

[48] T. Tung and F. Schmitt. The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes. *International Journal of Shape Modeling (IJSM)*, 11(1), June 2005.

[49] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Proc. of the Shape Modeling International 2003 (SMI'03)*, page 130. IEEE Computer Society, 2003.

[50] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3D model retrieval. In *Computer Graphics Forum*, volume 22, pages 223–232, September 2003.

[51] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 21(5):433–449, 1999.

[52] M. Novotni and R. Klein. 3D zernike descriptors for content based shape retrieval. In *Proc. of the 8th ACM symposium on Solid modeling and applications (SMA'03)*, pages 216–225, New York, NY, USA, 2003. ACM Press.

[53] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz. A reflective symmetry descriptor for 3D models. *Algorithmica*, 38(1):201–225, 2003.

[54] A. Başkurt, M. Daoudi, T. F. Ansary, S. Mahmoudi, F. Prêteux, J. Ricard, J-P. Vandeborre, and T. Zaharia. Etat de l'art sur l'indexation 3D. Technical Report http://liris.cnrs.fr/semantic-3d/, 2003.

[55] M. Hiroshi and I. Akira. 3D object recognition using MEGI model from range data. In *Proceedings of IEEE International Conference on Pattern Recognition*, pages 843–846, Jerusalem, Israel, 1994.

[56] J. Z. Xu, M. Suk, and S. Ranka. Hierarchical EGI: a new method for object representation. In *3rd International Conference on Signal Processing*, pages 926–929, October 1996.

[57] J. J. Koenderink and A. J. van Doorn. Surface shape and curvature scales. *Image Vision Comput.*, 10(8):557–565, 1992.

[58] D. M. Healy, D. N. Rockmore, P. J. Kostelec, and S. S. B. Moore. FFTs for the 2-sphere - improvements and variations. *J. Fourier Analy. Appl.*, 9(4):341–385, 2003.

[59] M. M. Kazhdan. *Shape Representations and Algorithms for 3D Model Retrieval.* PhD thesis, Princeton University, 2004.

[60] D. Bespalov, A. Shokoufandeh, W. C. Regli, and W. Sun. Scale-space representation of 3D models and topological matching. In *Proc. of the eighth ACM symposium on Solid modeling and applications (SMA'03)*, pages 208–215, New York, NY, USA, 2003. ACM Press.

[61] S. Biasotti, S. Marini, M. Mortara, G. Patanè, M. Spagnuolo, , and B. Falcidieno. 3D shape matching through topological structures. In *Proc. of the 11th International Conference on Discrete Geometry for Computer Imagery*, volume 2886 of *Lecture Notes in Computer Science*, pages 194–203. Springer, 2003.

[62] D. McWherter, M. Peabody, A. C. Shokoufandeh, and W. Regli. Database techniques for archival of solid models. In *Proc. of the sixth ACM symposium on Solid modeling and applications (SMA'01)*, pages 78–87, New York, NY, USA, 2001. ACM Press.

[63] J. Milnor. *Morse Theory, vol. 51 of Annals of Mathematics Studies.* Princeton University Press, Princeton, NJ, 1963.

[64] J. E. Barros, J. C. French, W. N. Martin, P. M. Kelly, and T. M. Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 392–403, 1996.

[65] G. Hetzel, B. Leibe, P. Levi, and B. Schiele. 3d object recognition from range images using local feature histograms. In *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume 2, pages 394–399, 2001.

[66] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to imge databases. In *Proc. of the IEEE International Conference on Computer Vision (ICCV'98)*, Bombay, India, 1998.

[67] M. J. Swain and D. H. Ballard. Indexing via color histograms. In *Proc. of the IEEE International Conference on Computer Vision (ICCV'90)*, pages 390–393, 1990.

[68] M. Meyer, M. Desbrun, P. Schröder, and A. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In H. C. Hege and K. Polthier, editors, *Visualization and Mathematics III*, pages 35–57. Springer-Verlag, Heidelberg, 2003.

[69] M. P. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Upper Saddle River, New Jersey, 1976.

[70] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 902, Washington, DC, USA, 1995. IEEE Computer Society.

[71] C. Dorai and A. K. Jain. COSMOS - A representation scheme for 3D free-form objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1115–1130, 1997.

[72] W. H. Press, B. P. Flannery, and S. A. Teukolsky. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[73] N. S. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice Hall Professional Technical Reference, 1990.

[74] M. Rosenblatt. Remarks on some non-parametric estimates of a density function. *Ann. Math. Statist.*, 27:832–837, 1956.

[75] E. Parzen. On estimation of probability density function and mode. *Ann. Math. Statist.*, 33:66–72, 1962.

[76] T. Cacoullos. Estimation of a multivariate density. *Ann. Inst. Statist. Math.*, 18:178–189, 1966.

[77] D. W. Scott, R. A. Tapia, and J. R. Thompson. Kernel density estimation revisited. *J. Non-linear Analysis Theory Meth. Applic.*, 1:339–372, 1977.

[78] D. W. Scott and G. R. Terrell. Biased and unbiased cross-validation in density estimation. *J. Amer. Statist. Assoc.*, 82:1131–1146, 1987.

[79] B. W. Silverman. Choosing the window width when estimating a density. *Biometrika*, 65:1–11, 1978.

[80] B. W. Silverman. Spline smoothing: The equivalent variable kernel method. *Ann. Statist.*, 12:898–916, 1984.

[81] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

[82] D. Comaniciu. *Non-parametric Robust Methods for Copmputer Vision*. PhD thesis, Rutgers University, 2000.

[83] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 24(5):603–619, May 2002.

[84] T. Duong. *Bandwidth selectors for multivariate kernel density estimation*. PhD thesis, University of Western Australia, 2004.

[85] D. Comaniciu, V. Ramesh, and P. Meer. The variable bandwidth mean shift and data-driven scale selection. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV'01), San Jose, CA*, pages 438–445, Vancouver, Canada, July 2001.

[86] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

[87] T. Zaharia and F. Prêteux. Indexation de maillages 3D par descripteurs de forme. In *Actes 13ème Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA 2002), Angers, France*, pages 48–57, January 2002.

[88] R. Herbrich, T. Graepel, and K. Obermayer. *Large margin rank boundaries for ordinal regression*. MIT Press, Cambridge, MA, 2000.

[89] A. Shashua and A. Levin. Ranking with large margin principle: Two approaches. In *In Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2003.

[90] K. Crammer and Y. Singer. Pranking with ranking. In *In Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2001.

[91] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[92] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, 2003.

[93] G. Fung, R. Rosales, and B. Krishnapuram. Learning rankings via convex hull separation. In *In Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2006.

[94] B. Schölkopf and A. J. Smola. *Learning with kernels.* The MIT press, Cambridge, MA, 2002.

[95] C-C. Chang and C-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[96] A. Ihler. *Kernel Density Estimation Toolbox for MATLAB (R13).* Copyright 2003 Alexander Ihler, 2003.

[97] P. Cignoni. *MeshLab v1.0.0*, 2007. http://meshlab.sourceforge.net/.

[98] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recogn. Lett.*, 15(11):1119–1125, 1994.

[99] P. Schröder and W. Sweldens. Spherical wavelets: efficiently representing functions on the sphere. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 161–172, New York, NY, USA, 1995. ACM Press.

[100] S. Edelman. On learning to recognize 3-d objects from examples. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(8):833–837, 1993.

[101] M. Kazhdan. An approximate and efficient method for optimal rotation alignment of 3d models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(7):1221–1229, 2007.

[102] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, 2000.

[103] H. Ling and K. Okada. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(5):840–853, 2007.