Roberto Célio Lau Lam

Correspondence of Three-dimensional Objects

Ph.D. Thesis in Computer Science (Informatics)

Developed under supervision of:

Prof. Johannes Martinus Hubertina du Buf

University of the Algarve

Faculty of Sciences and Technology

2016

Roberto Célio Lau Lam

Correspondence of Three-dimensional Objects

Ph.D. Thesis in Computer Science (Informatics)

Developed under supervision of:

Prof. Johannes Martinus Hubertina du Buf

UAlg FCT
UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS E TECNOLOGIA

University of the Algarve

Faculty of Sciences and Technology

2016

DECLARATION

I hereby declare to be the author of this work, which is original with some parts already published. Authors and works consulted are properly cited in the text and appear in the included reference list.

Declaro ser o autor deste trabalho, que é original e contém partes já publicadas. Os autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

COPYRIGHT

Faro, 2016

Roberto Lam

Acknowledgments

First many thanks go to Prof. Hans du Buf, for his supervision based on his experience, for providing a stimulating and cheerful research environment in his laboratory, for letting me participate in the projects that produced results for papers, thus made me more aware of the state of the art in Computer Vision, especially in the area of 3D recognition. Also for his encouraging support and his way to always find time for discussions, and last but not the least for the cooking recipes...
Many thanks go also to my laboratory fellows, to João Rodrigues, who invited me to participate in FCT and QREN projects, Jaime Carvalho Martins and Miguel Farrajota, for discussing scientific and technical problems, but also almost all problems in the world.
To all persons, that worked in, or visited the Vision Laboratory, especially those with whom I have worked with, almost on a daily basis.
A special thanks to the Instituto Superior de Engenharia at UAlg and my colleagues at the Department of Electrical Engineering, for allowing me to suspend lectures in order to be present at conferences.
To my family, my wife and my kids.

Some of the research published in this thesis was supported by the Portuguese Foundation for Science and Technology (FCT) through ISR/LARSyS pluri-annual funding (UID/EEA/50009/2013), FCT project SmartVision: active vision for the blind (PTDC/EIA/73633/2006) and FCT project Blavigator (RIPD/ADA/109690/2009).

NOME: Roberto Célio Lau Lam
FACULDADE: Faculdade de Ciências e Tecnologia
ORIENTADOR: Johannes Martinus Hubertina du Buf
DATA: Setembro de 2016
TITULO DA TESE: Correspondência de Objetos Tridimensionais

Resumo

O desenvolvimento tecnológico em sensores para aquisição de modelos tridimensionais, tem promovido a investigação relativa ao reconhecimento de objetos tridimensionais (3D). Tal como o reconhecimento de objetos em visão computacional, o reconhecimento de objetos 3D pode ser feito com base na forma dos contornos dos objetos. O grau de dificuldade da tarefa é maior devido, em primeiro lugar, à passagem do plano 2D para o espaço 3D e em segundo lugar pela forma de representação dos objetos. O uso de malhas poligonais não regulares, para representar objectos 3D, tem custos computacionais elevados devido à sua complexidade.

Esta tese aborda o reconhecimento de objetos tridimensionais. Sendo o processo de reconhecimento/classificação de objetos feito tendo como base as características especificas da forma/textura do objecto, iniciamos o estudo por uma revisão sumária da representação de objectos tridimensionais e os descritores globais e locais. A utilização de descritores globais permitem a captura da estrutura global do objeto, das relações geométricas entre os limites locais do objecto e permite assim reduzir o intervalo de procura na base de objectos.

Apresentamos um novo descritor global que é constituído por dois componentes: (a) um conjunto de valores colhidos de um processo de alisamento, repetido, da superfície do objeto e (b) um conjunto de valores obtidos pela aplicação formal da morfologia matemática. A aplicação sucessiva da filtragem Gaussiana sobre a superfície conduz ao alisamento da mesma e consequente redução da sua informação estrutural. No primeiro conjunto usamos essa informação e a taxa de contração da área de superfície. O segundo conjunto (b) de parâmetros é obtido através da evolução do volume diferença, entre o volume dilatado e o volume erodido, tendo como elemento estruturante uma esfera. Este descritor foi aplicado num conjunto de 42 objetos e obteve resultados similares aos métodos submetidos a SHREC 2010.

Tendo em conta que a representação mais vulgar dos objetos 3D é em malhas poligonais, estendemos o conceito piramidal utilizado em processamento de imagens multi-escala, às malhas poligonais tridimensionais. Pela utilização do filtro Gaussiano, com tamanho (raio) em intervalos constantes é possível no espaço multi-escala Gaussiano, na oitava seguinte, diminuir o numero de vértices sem perda de informação relevante. Este processo de redução de vértices do objecto permite diminuir o custo computacional associado aos vértices. Além disso, utilizando os valores *Shape Index* (SI) dos vértices, construímos mapas de saliência que evidenciam áreas côncavas e convexas. Os mapas de saliência permitem a extração de pontos-chave e para esse objectivo, foram usadas as primeiras 4 escalas (primeira oitava). Os pontos-chaves extraídos são centrais nas áreas classificadas como convexas e côncavas. O método de extração pontos-chave foi testado num conjunto de cinco objectos com a metodologia de validação humana proposta por Dutagaci et al. [2012]. No teste o nosso método obteve a 3ª classificação, em três objectos, na comparação com outros seis métodos.

PALAVRAS-CHAVE: Objetos tridimensionais, malhas poligonais, reconhecimento, pontos-chave, curvaturas, geometria diferencial, multi-escala.

Correspondence of Three-dimensional Objects

Abstract

This thesis focuses on the categorization and recognition of three-dimensional (3D) objects, using shape characteristics. We first present a brief review on the representation of 3D objects, using local and global descriptors (feature vectors). Using a global descriptor allows the representation of information regarding the complete structure of the object, including the relationships between local boundaries and salient features. A global descriptor can also reduce the representation space, resulting in a narrow interval for object database queries.

We propose a new global descriptor composed by two sets of features: the first set (a) is obtained by a repeated smoothing process of an object's surface, and the second set (b) is achieved using volumetric morphology. When Gaussian filtering is repeatedly applied over a surface it becomes smoother and eliminates some of the high-frequency details. We use this information plus the contraction ratio of the surface as the first component (a) of our descriptor. The second set (b) is collected from the evolution of the volume difference between the dilated and the eroded object, by using a sphere as structuring element. The complete descriptor was tested on a set of 42 objects and achieved similar results as the methods presented in SHREC 2010.

As 3D objects are commonly represented by polygonal meshes, we extend the pyramidal concept used in 2D image processing to 3D polygonal meshes. By using a Gaussian filter, in which the size is increased by a constant step, it is possible to reduce the number of vertices on a multi-scale Gaussian space (in the next octave), without any relevant loss of information. This diminishes the computational cost associated with processing vertices. Besides that, we can use the vertices' *Shape Index* (SI) values to build saliency maps showing convex and concave areas. These maps are useful for keypoint extraction, which are usually located at the centre of these areas. The obtained keypoints were tested against five other models, proposed by Dutagaci et al. [2012], where we ranked third, by comparison with six other methods.

KEYWORDS: Three-dimensional objects, 3D meshes, recognition, keypoints, curvatures, differential geometry, multi-scale, octave.

Contents

Chapter 1

Introduction

Abstract: This chapter introduces the scope of this thesis as well the problems faced in classification and recognition of three-dimensional objects. Representations of volumetric objects and their characterization will be described for the recognition process.

1.1 Scope of the thesis

Recent advances in hardware, in general or even in graphics processors, has driven software development in a way never seen before. The last decade was very productive in terms of animation films. They can be pure animated films or mixtures of regular films with avatar characters (CGI). Simulation, or augmented reality, has been used for a better understanding of natural and man-made phenomena. Three-dimensional (3D) objects are everywhere and their digital versions can be incorporated in computer software. It is now possible to embed a statue, placed in an Egyptian museum, in software applications for remote rendering. To increase the accuracy of human-machine interaction in software programs used to control manufacturing systems, complex synoptic elements can be replaced by 3D elements. The growing use of digital representations of three-dimensional objects promoted the creation of interest groups in 3D object representations and algorithms related to them; for example for partial or full object recognition.

The AIM@SHAPE project (2004 to 2007) had the following mission:

> advance research in the direction of semantic-based shape representations and semantic-oriented tools to acquire, build, transmit, and process shapes with their

> associated knowledge. We foresee a generation of shapes in which knowledge is explicitly represented and, therefore, can be retrieved, processed, shared, and exploited to construct new knowledge

(AIM@SHAPE [2004]; Visionair [2011]). The canonical representation of partial 3D objects will enable the recognition of object parts, and by their combination the recognition of entire objects. The SHape REtrieval Contest (SHREC), within the AIM@SHAPE project, started in 2006 to evaluate the effectiveness of 3D shape retrieval algorithms.

Meanwhile, the study of macromolecules (proteins) has also experienced an important contribution of hardware/software developments. The three-dimensional models obtained by X-ray protein analysis are stored in a large database, the Protein Data Bank (PDB); see Berman et al. [2013] for further studies. The PDB file format describes the three-dimensional structure of a protein, and it provides a description and annotation, including atomic coordinates and connectivity. The visualization/rendering of these macromolecules started in 1968 with the Brookhaven Raster Display; at present time there exists a large variety of viewers for PDB data. Recently, due to the fact that proteins are vital parts of living organisms, large-scale studies, particularly of their structures and functions, has lead to the proteomics research field. In almost all cases, different conformations[1] of proteins lead to changes of their biological function. Also, the 3D configuration of a protein aids in identifying its interactions and function. In order to try to predict the functions of a protein, bioinformatics researchers have been working on extracting and recognizing the auxiliary structures (secondary, tertiary and quaternary) and how these are connected [Axenopoulos et al., 2013].

Classical object recognition was based on image processing. In complex scenes, where an object is among other objects, to find the correct keypoints and to apply the proper segmentation algorithm are the principal steps in detecting and recognizing objects. First attempts in the recognition of three-dimensional objects were applied to range images [Besl and Jain, 1985] and to stereoscopic images [Devernay and Faugeras, 1994; Wildes, 1991]. One of the main problems related to 3D reconstruction of objects acquired with these technologies is the self-occlusion of parts of the object, and occlusions in case of complex scenes. With advanced sensor technology, both active and passive with fast and accurate range scanners,

[1]Conformation is a change in the shape of a protein, sometimes they may suffer reversible structural changes to perform its biological function.

it is now possible to obtain multiple range images, to extract their polygonal meshes and to fuse them in a common coordinate system. Curless [1999] predicted the use of 3D fax, i.e, the actual 3D printers.

3D polygonal meshes produced by range scanners are now being used in very distinct research and application fields. As stated before, there are computer vision researchers working on the recognition process of three-dimensional objects. Multidisciplinary groups, with researchers from computational geometry, statistics and optimization theory, are trying to define knowledge on three-dimensional shape. Humans can easily recognize 3D objects or parts. We can recognize part of a chair, associate its main function and, if necessary, use a different chair or another object to sit on. There are many practical applications that can benefit from this knowledge: 3D object and surface recognition, storing, indexing and retrieval of 3D objects in digital libraries. If the quality of knowledge produced is better it can be incorporated in a new research field of object definition and its semantic recognition, as well as in algorithms and methods used for image processing and machine vision.

Since the beginning of research in computer/machine vision, a huge collection of publications has been produced. However, there also are excellent textbooks that cover most subjects: *Multiple View Geometry in Computer Vision* by Hartley and Zisserman [2003], *An Invitation to 3-D Vision: From Images to Geometric Models* by Ma et al. [2003], *Computer Vision: Algorithms and Applications* by Szeliski [2010], *Introductory Techniques for 3-D Computer Vision* by Trucco and Verri [1998], *Computer Vision: A Modern Approach* by Forsyth and Ponce [2012], *Three-dimensional Computer Vision: A Geometric Viewpoint* by Faugeras [1993] and *3D Imaging, Analysis and Applications* by Pears et al. [2012].

The main focus of this thesis is on the recognition of three-dimensional objects. The main topics are:

- To study local and global characteristics of three-dimensional objects (surfaces) and the influence of resolution on the quality of these characteristics. To extract the most faithful and reliable information according to coarse/fine resolution.

- To develop a scheme for the extraction of points, or surface parts, with most important characteristics.

- To develop methods for the recognition and categorization of objects.

In the next sections, a brief introduction to registration, representation and recognition of 3D objects is presented. At the end, the structure of this thesis will be explained.

1.1.1 Object representation

Shape analysis is an old problem of computer vision and image processing. In general the shape of an object is a binary representation. The shape can be seen as the contour (surface) which divides the volume into two sets: inside and outside of the object. The representation of shape in classical image processing has been extensively researched, with well-known solutions. The three-dimensional case is work in progress. This is due to (1) recent advances in 3D scanning, and (2) representing data in three-dimensions is more complex than in 2D (the plane), because of all the problems related to discontinuities. In some applications, where the objects can be easily segmented, the methods from image analysis/processing can be used for shape analysis.

In the 3D case we move from 2D image analysis to 3D scene analysis (known in computer graphics as *view frustrum*). Devices used to produce range or stereoscopic images can sense only the surfaces, and passive or active techniques are used to obtain 3D data. In passive mode there is a set of methods called *Shape from X*, where X means the visual cue used for shape extraction: focus, shading, texture or stereo disparity [Horn, 1970; Bulthoff, 1990; Curless, 1999; Devernay and Faugeras, 1994]. From 2D images the surfaces of volumetric objects can also be obtained by the extraction of isosurfaces of three-dimensional scalar fields. A polygonal mesh representing the isosurface can be obtained by the marching cubes algorithm [Lorensen and Cline, 1987; Newman and Yi, 2006] or by constructing the correspondences of planar contours (planar cross sections) [Bajaj et al., 1996; Braude et al., 2007]. The representation of a 3D shape is related to the used method and to the applications that will use it. Computer-aided design, visualization and graphics are fundamental in a variety of end applications, from manufacturing (CNC machinery) and health care (computerized axial tomography) to entertainment (gaming). The output of sensor systems can vary from 3D points to range (depth) maps, from polygons to scalar fields. The accuracy of systems may produce a truthful representation of the scene or a noisy surface (in some cases *non-manifold*; see Section 2.2.1). In the latter case, a lack of precision can hamper the next stages of the analysis process.

Raw data collected from a sensor system can be a set of points with 3D coordinates.

They are designated *point cloud*, P, where $P = \{v_1, v_2, ..., v_n\}$ and $v_i \in R^3$. These points, or vertices, can be gathered in a structured manner, e.g. sampled on a regular grid, and they can be enhanced by information such as texture and the surface normal at each vertex v_i. In the case of a structured point cloud, there is common information on adjacent vertices and an implicit mesh connectivity exists among neighboring points. After being constructed, three-dimensional surfaces can be explicitly represented by polygonal meshes [Hoppe et al., 1993; Hoppe, 1996; Botsch et al., 2006, 2010], by splines [Farin, 1996; Eck and Hoppe, 1996; Zorin and Schröder, 2000] or implicitly by surfaces represented by a scalar field [Wu and Kobbelt, 2003; Velho et al., 2002].

Polygonal meshes are very common in representing 3D objects or surfaces. They were introduced in the early days of computer graphics. The data structures that support them are easily implemented, and then used in rendering algorithms for visible-surface determination or illumination/shading models [Foley et al., 1990; Watt, 1993]. Of all polygonal meshes, triangular ones are more suitable for representing the *two-manifold* nature of physical surfaces and, if compared to quadrilaterals or n-gons, they can better describe surfaces with possible ambiguities caused by non-planar, concave or self-intersecting polygons (polygons with more than 3 sides).

1.1.2 Object recognition

Object recognition is one of the most common tasks in humans life. We can recognize objects in real scenes as well as in still images. Humans can recognize objects when seeing them from different view points and distances (sizes), and when they are rotated or translated. An object can be partially occluded, because the actual view point can also be recognized [Bulthoff et al., 1995], depending on the degree of *previous knowledge* of that object. Nobody can recognize something without having previous knowledge of it. As Tulving and Thomson [1973] stated in their seminal paper: "*Retrieval operations complete the act of remembering that begins with encoding of information about an event into the memory store.*"

In computer vision, recognition of three-dimensional objects is not an easy task. Regardless the complexity of building a volumetric object by using image processing techniques, with all the problems and ambiguities that can be due to pose, illumination or occlusion, recognizing 3D objects is more complex than planar ones (2D projections). In typical image analysis, the understanding of contents and context comes from recognizing the objects. The

shape of an object can be defined by its contour, or silhouette. Therefore, in some image processing applications, object analysis can be equivalent to shape analysis when object segmentation can be easily obtained.

The analysis of three-dimensional objects, for the purpose of recognition or classification, is more complex but very similar to the process as used in 2D image processing: the 3D object representation results in a non-numeric *form* of the original object. All important object characteristics must be preserved for later use. Based on the object representation, a description must be produced: a set of numeric values which quantify the characteristics considered important. A descriptor (vector) is generated for each object, and the purpose of the description vector (also denominated feature vector) is to reduce, or narrow, the comparison (search) space of object characteristics [Loncaric, 1998; Veltkamp, 2001].

A general-purpose recognition algorithm must handle three problems: (1) normalization. 3D objects have their own coordinate systems and units, and the recognition algorithm must normalize the objects before processing them; otherwise, the subsequent steps will be hampered by inconsistencies, outlier values, and so on. (2) feature extraction. The most important characteristics (features) of the object must be chosen. They should have enough discriminative power to distinguish one object from all others. (3) similarity search. For any given object, the algorithm should efficiently find the nearest neighbors in feature space. The selection of 3D object descriptors (feature vector) must obey some properties: invariance to geometric transformations (translation, rotation and scaling or reflection), robustness to level-of-detail (different sampling rate, mesh tessellation), feature representation in multi-resolution, and robustness to noise and outliers. Lastly, the object descriptors can be acquired in the spatial or in the frequency domain. In the spatial domain, the object representation is in the ordinary Euclidean space. In the frequency (spectral) case, the object representation is build by using a proper mathematical transform operation [Vranic and Saupe, 2001; Kazhdan et al., 2003; Novotni and Klein, 2003; Papadakis et al., 2007].

1.2 Overview of the thesis

Methods for 3D object recognition can be classified into two groups, model-based and view-based. Despite the difficulties and complexity of 3D model reconstruction, model-based methods are subject to intense research because of the reasons explained in Section 1.1. The work presented in this thesis addresses the model-based (also referred to as geometry-based)

paradigm in the spatial domain.

Chapter 2 presents a more detailed overview of the registration, integration and representation of three-dimensional objects. It explains generically most known and the most significant features for characterizing 3D objects. It finalizes by presenting some preliminary conclusions that will lead us towards developing an invariant object recognition and categorization algorithm.

Chapter 3 introduces local and global descriptors for compact representations of 3D objects. It shows that these provide very important information for similarity search in categorization and recognition. It also introduces the concept of multi-scale space.

Chapter 4 discloses two features sets that, if integrated, allow invariant recognition of 3D objects. These features are area- and volume-based. They are obtained by applying a smoothing filter and by applying mathematical morphology.

Chapter 5 introduces multi-scale theory by using Gaussian filtering, Laplacian and Difference-of-Gaussian operators. It extends the sub-sampling or decimation mechanism to 3D objects. It re-defines saliency maps by using Shape Index, and illustrates the extraction of keypoints in a multi-scale scheme.

Final remarks and ideas for future research are presented in Chapter 6.

Parts of this thesis and related work have already been published and also presented at conferences. Chapters 3 and 4 were published in 2009, 2011 and 2012 in KDIR, IbPRIA, ISCV, ICIAR and IGI-Global, respectively. Chapter 5 is being prepared for submission to a journals like Int. J. of Computer Vision, Computers & Graphics or Graphical Models. Appendix B lists all publications.

Chapter 2

Overview of 3D object representation and recognition

Abstract: This chapter presents a more elaborate overview of the registration and representation of three-dimensional objects. It explains generically most known and most significant features for characterizing 3D objects. It finalizes by presenting some early conclusions that will lead us towards developing an invariant object recognition and categorization algorithm.

2.1 Introduction

The interest in three-dimensional objects, in digital form, is motivated by many practical rendering applications: animation and visual simulation (for gaming or simulation in augmented reality), anatomy (for planning surgical procedures, remote analysis and consult) and preserving archaeological objects (for remote rendering, similarity studies). Besides these few examples, and because of the proliferation of 3D objects, there is an interest for producing a consistent knowledge about three-dimensional objects [AIM@SHAPE, 2004; Visionair, 2011]. The possibility of representing real 3D objects, in digital format, has begun with the research work on sensors. There is a vast variety of sensors available on the market, the Kinect from Microsoft and the Structure Sensor from Occipital are good examples [Microsoft, 2009; Structure Sensor, 2014]. 3D imaging systems can be classified as passive or active ones. A passive system acquires scene information without using its own source of illumination or electromagnetic radiation. Active systems employ light projection, usually in infra-red or with visible wavelengths.

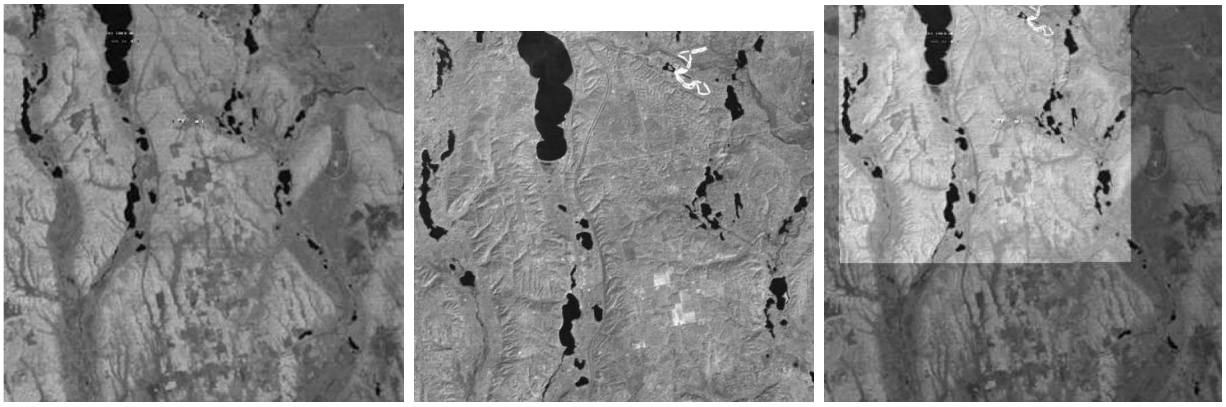


Figure 2.1: From left to right: multispectral scanner image, thematic mapper image and overlay after resampling of the second image. Adapted from Fig. 1.1 in Goshtasby [2005].

In general, the majority of range imaging devices do not capture a 3D object in one shot; it is necessary to capture its shape from all sides. Scanning sensors must scan *all* surfaces of an object, and then the object surface can be described by a finite number of range images, where each image has its own coordinate and reference system. In order to get a unique and entire object from the image set it is necessary to merge all images into one common coordinate/reference system. This is usually called the registration process [Chen and Medioni, 1992; Turk and Levoy, 1994; Hung et al., 1999]. The merging of two or more overlapping images depends on many factors. The images could have been taken from the same scene at different times, from different viewpoints and/or by different sensors. Multi-view registration, where images of a same scene are captured from different viewpoints, is used to get a bigger or better view of the scene. In 2D, thus concerning image processing, registration is restricted to the plane, and correspondences between images are established using 2D geometric transformations. Figure 2.1 shows the re-sampling and merging of two Landsat images. The registration process can be quite complex due to the diversity of the images: they can be acquired from different applications, by different types of sensors, and they can have a different quality. It is almost impossible to conceive an algorithm that can be applied to all registration tasks. Registration must deal with geometric deformations as well as noise corruption and resolution. Goshtasby [2005] and Zitová and Flusser [2003] proposed that all registration methods must include 4 steps:

- *Preprocessing.* Due to the variety of images to be registered and because of some types of degradation it is necessary to perform a preprocessing step. Some scale adjustment, noise removal or segmentation is applied in order to prepare the images for the feature

selection step. For example, if two images to be registered have a different size or resolution, one of them must be resampled to the scale of the other image. This adjustment will make feature correspondence easier.

- *Feature detection/selection.* Objects are distinguishable because of their different characteristics at edges, contours, corners, closed-boundary regions, etc. The type of features to select in an image (object) depends on the type of image under registration. Images from man-made scenes have different features than those from natural scenes. The features used must reflect this.

- *Feature correspondence.* This step involves determining the correspondence between features selected in one image and those in a reference image. Another possibility is to select features in both images independently and then finding the correspondences between them. The former technique is appropriate when features contain synthesis information such as regions or templates. Feature descriptors and similarity measures, as well spatial relationships of the features are used in the matching process.

- *Transform/Mapping functions.* The alignment of sensed images depends on the type and parameters of the transforms. If the coordinates of a set of corresponding points in the images are known, the transform can consist of re-sampling the sensed image to the geometry of the reference image. Otherwise, a more complex transformation must be used in order to minimize the geometric difference.

Registration of 3D objects from 2D views (projections) can be done if it is possible to obtain the 3D shape from the images. The extraction of 3D shape from 2D images is feasible if the sensed object/scene fulfils certain conditions. There is group of methods, designated *Shape from X*, that extract the shape by means of specific features: shading, texture and focus. Images from smoothly shaded objects can provide their shape. For humans it is possible to infer the shape of an object by interpreting the shading variation. Likewise, in computer vision the shading variation is interpreted as changes of the surface normal over the object, as the brightness changes depend on the angle between local surface orientation and the incident illumination [Zhang et al., 1999][1]. Results can be improved by using controlled sources of light, such as structured light or photometric stereo [Geng, 2011; Woodham, 1989].

[1]Simplification of Lambertian surface concept

Texture is another feature that can reveal an object's shape, as the shape can be determined from the variation of the surface structure. The major problem of using texture in extracting shape is the type of texture. Textures in images are commonly referred to as *texels*, and they can be deterministic or stochastic. Deterministic textures consist of repeated fixed geometric shapes, such as circles, squares/rectangles or other patterns. Stochastic textures are patterns defined by their statistic properties. In case of deterministic textures, shape information can be found from the relation between normal vectors (3D object surface) and perspective distortion (image processing) by means of known distorted shape parameters. Spatial frequency properties (e.g. power spectrum, granularity or orientation distribution) are main characteristics of stochastic texels. Knowledge of theses properties and the projection of the shape onto the image plane can be used to construct the 3D shape [Clerc and Mallat, 2002; Malik and Rosenholtz, 1997].

The last way of estimating a shape is by using the focus. It is known, mainly by photographers, that the image will be blurred at a distance of the focal plane, i.e., the object surface that is not in the focal plane will be blurred. Moving the object surface in discrete steps relative to the focal plane allows to establish the surface's shape [Nayar and Nakagawa, 1994; Favaro et al., 2003; Marturi et al., 2013]. Despite that a partial 3D shape can be obtained from a single view, with "Shape from X" methods, the above methods are not practical in terms of robustness and speed.

In order to get the entire shape of an object/scene, it is necessary to have well defined geometric relations between the images. Most common approaches for capturing 3D shape use multiple views because of the complexity of solving the correspondence problem without previous calibration of the sensor system. The multiple-view case is less complex than the single-view one, since all views are captured under conditions that allow for an accurate calibration process. Registration is done in 3D space and correspondences of surfaces are computed using 3D geometric transformations. Passive multi-view imaging systems use cues related to human vision. In stereo vision, depth is perceived by merging and processing disparity between left and right images, possibly also using knowledge of photogrammetry. With the advances in the field of computer graphics, the techniques and geometry involved become more mature and accurate. In addition to classical stereo images, the binocular view, 3D shape can be captured from more than two views, using multiple and simultaneous sensing devices. Matching more images usually leads to better results. Figure 2.2

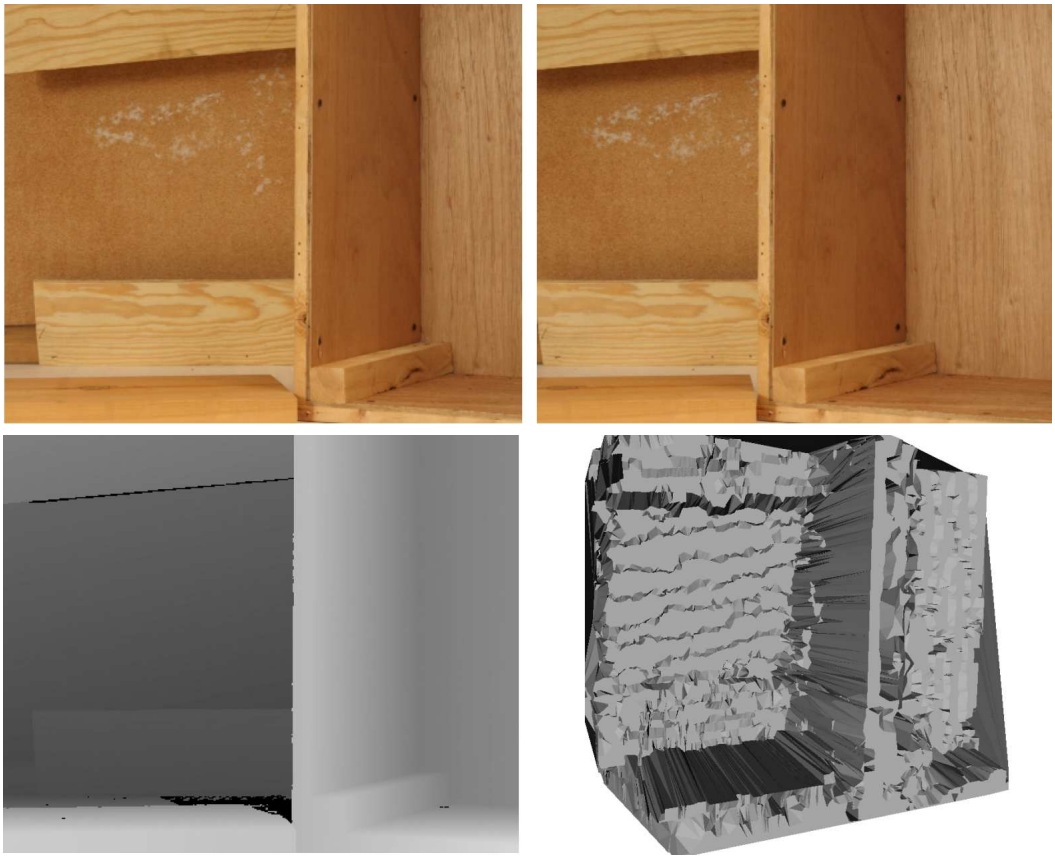


Figure 2.2: Top: left and right image of Wood1 from Middlebury Stereo Vision - Datasets. Bottom: ground truth of the left image and triangulated 3D surface disparity map.

shows the left and right images, the ground truth of the left image (Wood1, Hiebert-Treuer et al. [2006]) and a 3D triangulated disparity map [Rodrigues et al., 2012b]. The stereo correspondence in passive systems can be in sparse form, by the matching of profile curves (object boundaries or occluding contours). The location of profile curves depends on camera viewpoint; therefore, matching curves of two images can lead to wrong shape measures. If there are more than three images with little variation it is possible to fit a local circular arc over epipolar plane, and then match the corresponding points on contour edges. A surface mesh can be obtained directly from the matched points [Boyer and Berger, 1997; Liu et al., 2008]. Today, most stereo matching methods are based on dense correspondences, and most are applied to image-based modeling or rendering. The characterization and taxonomy as proposed by Scharstein and Szeliski [2002] has four steps for dense stereo matching algorithms: 1) matching cost computation (to estimate pixel-matching similarities), 2) costs to support aggregation, 3) disparity computation and 4) disparity refinement. The first and second steps, see Scharstein and Szeliski [1998], can be applied locally (defined by window

limits) or globally, by skipping the aggregation step and doing the disparity computations in an energy-minimization framework.

Active systems use their own source of illumination with single- or multi-view approaches. They employ one or more sources of controlled light so they can extract 3D structures even from smooth and texture-less areas, which is hard for passive systems. The correspondence task can be eliminated or becomes more simple. In general passive systems are very demanding computationally, and the 3D information is acquired from processing the stereo images for extracting feature correspondences. This can be difficult because of noise caused by scene textures and illumination. Active systems, with IR light or dynamic laser projection and fast cameras can grasp images in milliseconds. They, like the passive ones, are based on geometry (triangulation) and optical physics: the reflectance of light on surfaces. Spot and stripe scanners and systems which use patterns of structured light (area scanners) are the three groups in which most common systems can be classified. Spot scanners use a collimated laser beam for the distance. This approach allows to control the spatial sampling of the scene. Some spot scanners, with a complex opto-mechanical mechanism, can allow translations/rotations of the sensors or laser beam. Stripe scanners use a cylindrical lens to produce a plane of light which is projected onto the scene. The stripe of light is read by a linear array of sensors. The profile image is computed with the help of epipolar geometry. Figure 2.3 shows the basic mechanisms of spot and stripe scanners [Pears et al., 2012].

Reconstruction of 3D surfaces has been the main goal from the beginning of using 3D depth scanners in computer vision. Since then faster and more robust registration algorithms have been proposed. The Iterative Closest Point (ICP) algorithm, proposed by Besl and McKay [1992], provides a solution of the alignment problem of two 3D point sets (A and B). ICP has become a standard due to the large number of algorithms which are based on it. ICP is a three-step algorithm: (a) pair the closest 2 points, one from A and the other from B, (b) find the translation that minimizes the mean square error (MSE) of the distance between linked points and (c) apply the translation to A and update the MSE. These three steps are iterated until the MSE has converged to a minimum. At the same time, Chen and Medioni [1992] proposed an algorithm with a similar iterative scheme: they used the distance between the points on one surface (P) to the other surface (Q), being the distance measured over the normal vector of p_i on P and the intersected point q_j on Q, figure XX shows the 2D version of the method. The transformation should minimizes the

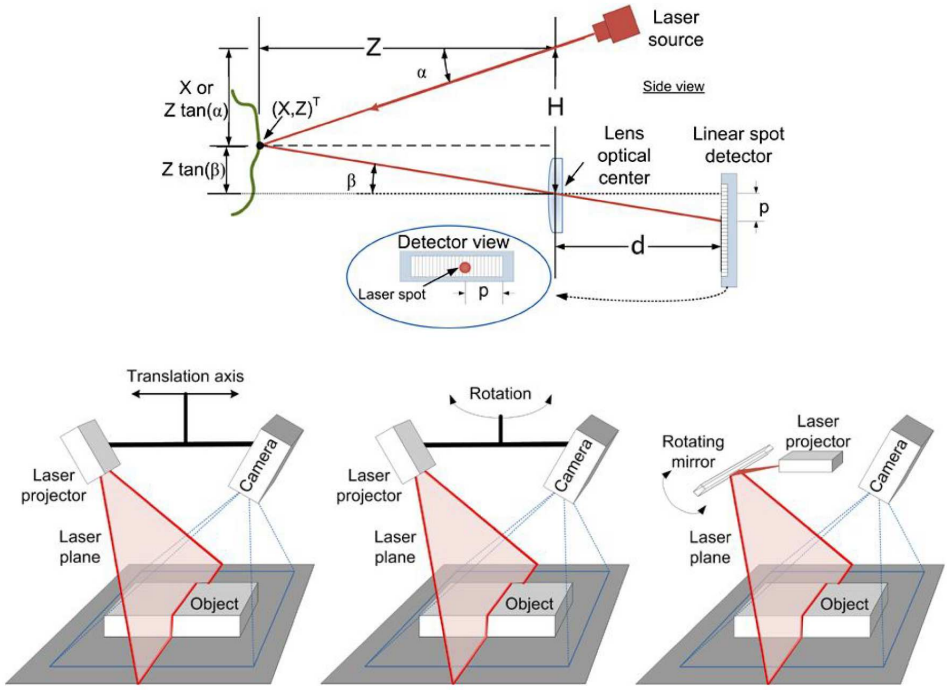


Figure 2.3: Top: Spot scanner, diagram of a single point optical triangulation sensor based on a laser beam and a linear spot detector. Bottom: Stripe scanner, from left to right: translation of the laser and sensor, rotation, and only the laser beam is deflected by a rotating mirror. Figures retrieved from Pears et al. [2012, Chap. 3].

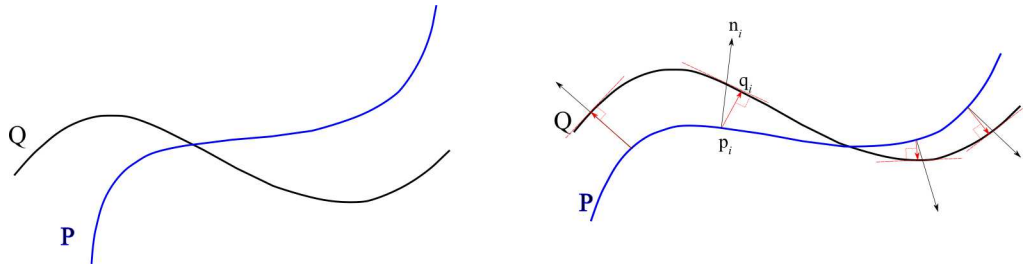


Figure 2.4: Alignment of two 2D curves. Left: Curves before the Chen [1992] method. Right: One iteration of the method.

distances and the differences between corresponding tangent planes. This method has two main restrictions: some previous registration knowledge is required, and the geometry of the data acquisition setup must be known or a high-level matching process will be needed. Due to the local nature of the ICP alignment algorithm, variations with more complex mathematical formulations were proposed. One of them solves the correspondences between the two point sets by computing the motion (rotation and translation) with Singular Value Decomposition (SVD)[Umeyama, 1991]. Correspondences can be based on local descriptors, or invariances

to 3D rigid transformations, e.g. *spin images* that are local projections of the 3D surface features in 2D images [Johnson and Hebert, 1999; Assfalg et al., 2004]. In some cases, the initial, coarse alignment can be estimated by using Principal Component Analysis (PCA). However this approach has some drawbacks, namely some data sets lead to mis-selection of the components or their orientations. More information on 3D registration is provided by Tam et al. [2013] and Díez et al. [2015].

Once the partial surfaces, or their point clouds, are aligned, they must be interconnected to form a single surface, or a single point cloud. The obtained surface must be represented in a form which is suitable for a final purpose. Some 3D registration approaches use the extra information given by the scanners, e.g. active stripe scanners, to extract the partial 3D surfaces and to align them into a coherent object surface. The *volumetric range image processing* algorithm uses a weighted distance function to represent each depth image, and their merging is done by a weighted averaging procedure [Curless, 1999].

With the development of new scanners in medical imaging, namely Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), began a research field to extract and represent the 3D surfaces of organs and bones. The *marching cubes* algorithm extracts iso-surfaces in the scanned CT and MRI volumes. It takes advantage of data coherency in the image stack, by extracting the surface based on 256 possible voxel configurations [Lorensen and Cline, 1987; Newman and Yi, 2006]. Surface reconstruction has been an active research topic in the past two decades, focusing on priors and assumptions made in order to correct or avoid data imperfections [Berger et al., 2014; Loke, 2006].

The actual representation of a 3D surface depends on the final purpose. Applications like Computer-Aided Design (CAD) for architecture and structural engineering, or Computer-Aided Manufacturing (CAM), or a 3D video game, are quite different. In case of CAD surfaces, it may be necessary, due to aerodynamic or aesthetic restrictions, to ensure a certain smoothness. As in mathematical analysis (smooth of a function), surfaces belongs to class C^0 if they are continuous, class C^1 possess all the surfaces that are differentiable and its derivatives are continuous (normal surface changes in smoothly way), C^2 contains the surfaces that have continuous second derivatives (surface curvatures variation are smooth). This definition can be extended, as the one in mathematics, to C^∞.

In general, a 3D object representation can be seen as an abstraction of a solid object. It can be represented by its surface (object boundaries) and the actual implementation will

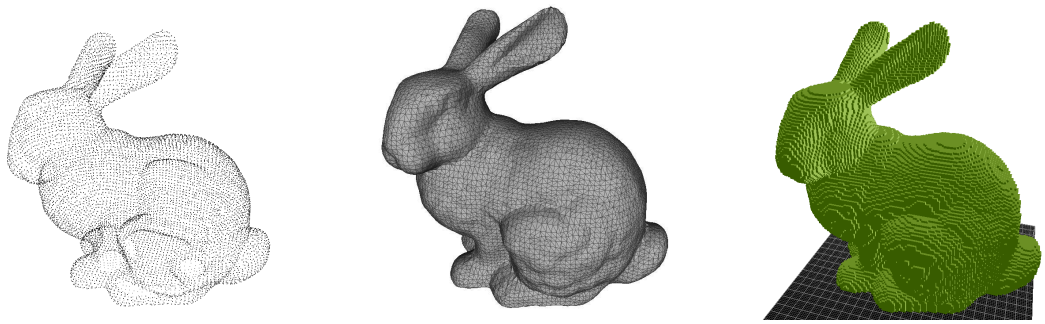


Figure 2.5: From left to right: point cloud, triangle mesh and voxel representation of the Bunny data (Stanford 3D Scanning Repository).

depend on some factors like computer memory, data structure, an approximation defined by a finite number of parameters and the efficiency of typical operations. Furthermore, the rendering of complex surfaces should be fast and accurate, it may be necessary to compute intersections between surfaces, to edit surfaces with different levels of resolution, and to compute model differences and errors [Hubeli and Gross, 2000]. The taxonomy of 3D representations covers three groups. At a low level, we have raw data, in the form of a point cloud, a range image or a polygon soup. At the middle level, we have solids represented by voxels, a Binary Space Partition (BSP tree) or a Constructive Solid Geometry (CSG). Surfaces are cast into polygonal meshes. To coarse meshes some subdivision scheme can be applied to define smooth surfaces, or they can be approximated by parametric and implicit methods. Fig. 2.5 shows the Bunny data in different representations: point cloud, triangular mesh and voxel. Finally, the third group contains the high-level representations, like scheme graph, skeleton, or they can be application specific [Botsch et al., 2010]. From the acquisition to the production of 3D models (their representations) we have a new field of computer science, *digital geometry processing*. It integrates mathematical models, data structures and algorithms for the analysis and manipulation of geometric data. The profusion of data sources, the need of processing operations and the rendering or manufacture technologies have lead to a mature and wealth of 3D representations of geometric data.

2.2 Polygonal meshes

The versatility of polygonal meshes made them a popular representation in many different areas in computer graphics and geometric processing, or in computational geometry. The use of polygon or triangle meshes as an alternative to spline surfaces is based on their sim-

plicity; they are suitable for flexible and efficient processing [Kobbelt, 2003]. The conceptual representation of polygonal meshes allows for the use of efficient data structures, and to implement fast algorithms to query, add, remove or transverse mesh elements.

With polygonal meshes it is possible to construct an adequate surface by applying spline surfaces (also known as NURBS) or surface subdivision to smooth localized areas. Both methods use the vertices and their connectivity to approximate the original, piecewise linear surface. The splines method uses *control points* to locally control the surface. By using either piecewise polynomial or B-spline functions, each control point influences the local structure. Topologically complex surfaces must be decomposed into elementary surface patches, and they must be smoothly connected. Processing costs are mainly due to geometric linking. Surface subdivision schemes use the original polygonal mesh as initial surface and then apply a repeated subdivision procedure in which new vertices are introduced in the center of a polygon, or at its edges. Their positions are adjusted in consonance to a set of local averaging rules. The subdivision methods are confined to produce meshes with subdivision connectivity (or semi-regular connectivity). To overcome the restriction caused by semi-regular connectivity, most algorithms, or applications for geometric processing, use arbitrary triangle meshes with irregular connectivity. They are very useful because they can represent smooth surfaces with a better precision [Besl, 1995; Sun et al., 2002; Morvan and Thibert, 2002b].

2.2.1 Triangular meshes

Triangle meshes are a particular case of polygon meshes. Triangles can represent a surface in a better way compared to other types of polygons. Apart from this intuitive notion, Morvan and Thibert [2002a] developed explicit approximations of the areas and normals of smooth surfaces, and the closest triangular mesh, closely inscribed in the smooth surface, even if all vertices of a triangle do not reside in the surface. Another advantage of using triangular meshes is that they allow to determine intrinsic features of the mesh/surface. From here on we will use the term mesh for surface and will only address triangular meshes.

A mesh is an explicit representation of a geometric and topological structure, formally represented by a set of vertices $V = \{v_1, ..., v_V\}$ with $v_i \in R^3$, and a set of triangular faces $F = \{f_1, ..., f_F\}$, with $f_i \in V \times V \times V$. Each triangular face has its vertices in V. Each vertex v_i from the mesh is associated to its corresponding surface point, there is a *one-to-one*

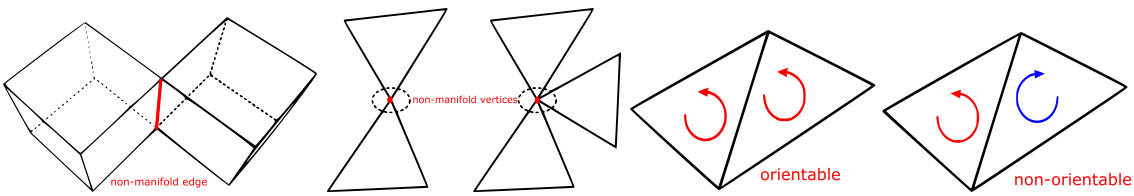


Figure 2.6: Non-manifold geometric representations. From left to right: edge, vertex and non-orientable facets.

correspondence between each vertex $v_i \in V$ to a p_i. If P is the set of points in the surface, then the correspondence is defined as:

$$P = \{p_1, ..., p_V\}, \quad p_i := p(v_i) = \begin{pmatrix} x(v_i) \\ y(v_i) \\ z(v_i) \end{pmatrix} \in R^3.$$

A continuous, piecewise linear mesh has an approximation error with the original smooth surface. This error decreases as the number of triangles increases and their size gets smaller. Hence, vertex density can be adapted to the local surface curvature. Planar zones can have a sparse number of vertices while curved areas must be described with a more dense vertex population.

One important mesh property is the *manifoldness*. Manifolds are topological spaces with extra structures or constraints. From physicist viewpoint a n-manifold is an object with *n degrees of freedom*. Hence, 1-manifold are lines and curves, and 2-manifold are surfaces. All real objects have this property. Local manifold existence, also known by geodesic neighborhood, to a surface point p can be defined as: every point q within a small sphere with radius δ, centered on point p, has an image contained in a circle with radius $\varepsilon = f(\delta)$, around the image of p. A more intuitive definition of the manifold of p is to consider the surface around it: every point inside a geodesic neighborhood is homeomorphic to a disk. Triangular meshes generated from discrete points can lead to degenerate situations with non-manifold edges, non-manifold vertices or self-intersections, see Fig. 2.6. Most algorithms do not deal with non-manifold meshes because of the missing geodesic neighborhood. The existence of non-manifold edges and vertices hampers the computation of intrinsic surface features, such as curvature, area, volume and so on. Another global characteristic of a mesh is that it can be (or not) *watertight*: the mesh is closed or open. If it is necessary to compute the volume of an object, its mesh must be watertight.

Triangle meshes are very useful for applying geometric operations. They are compact and can be rendered fast. Most graphics engines support lists of triangles with efficient queries:

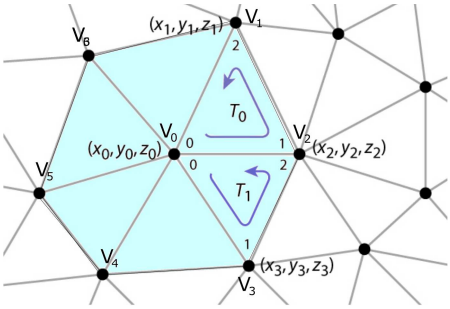


Figure 2.7: Triangle mesh structure with 2 triangles exposing the counterclockwise indexing of their vertices (x_i, y_i, z_i) and the 1-vicinity around V_0 in light blue.

all vertices of a triangle, all triangles around a vertex, finding triangle strips, implement subdivision schemes and mesh editing. Data structures of triangle meshes can be a simple set of separated triangles, an indexed triangle set (with shared vertices), triangle strips/fans (used by compression schemes for transmission hardware), or triangle-neighbors (to support adjacency queries). For very large and complex polygon meshes, some work has been done to create more efficient data structures [Serna et al., 2011; Luffel et al., 2014].

2.2.2 Feature characterization

In the previous subsection (2.2.1), the correspondence, with some restrictions, between an object and its triangle mesh was discussed. Once a triangle mesh is created, it is possible to compute mesh features, from very simple ones like the area to more complex ones like surface curvature. Features can be local or global. Local features are subject to neighborhood context: the radius of vicinity can depend on the local mesh resolution. It is possible to overcome this limitation by applying bicubic interpolation over the local patch.

In triangle meshes the neighborhood of a vertex is usually defined in terms of edges around it: *1-vicinity* comprises all triangle facets within a displacement at most *1* edge from a given vertex: for *2-vicinity* the triangles must have their vertices at most *2* edges away. Figure 2.7 illustrates the *1-vicinity* (or 1-ring) of V_0 in light blue.

Basic geometric features are area centroid and normal (of one or more triangles). These features can be used as weight factors in subdivision schemes, or in decimation for reducing the number triangles in a mesh. More complex features can be based on differential geometry. It is possible to compute Gaussian curvature based on the intrinsic geometry of the surface. The *Egregium theorema* established that Gaussian curvature is an intrinsic invariant; it can

be determined from the measurements of lengths on the surface itself [Gray et al., 2006]. A regular surface can be considered a metric space, with its own intrinsic metric if it is possible to define a parametrization $\phi = \phi(u, v)$ and surface element as

$$I = ds^2 = E(u, v)du^2 + 2F(u, v) + G(u, v)dv^2, \tag{2.1}$$

being $E = \phi_u \cdot \phi_u$, $F = \phi_u \cdot \phi_v$, $G = \phi_v \cdot \phi_v$ (ϕ_v, ϕ_v are partial derivatives) and I known as first fundamental form. The second fundamental form is defined as

$$II = Ldu^2 + 2Mdudv + Ndv^2 \tag{2.2}$$

with $L = \phi_{uu} \cdot n$, $M = \phi_{uv} \cdot n$, $N = \phi_{vv} \cdot n$ and n the normal vector. Local surface curvature can be expressed by the first and second fundamental forms and their partial derivatives (first and second order). From a practical point of view it is possible to express local curvature on the basis of two principal curvatures K_1 and K_2 (the maximum and minimum values of all curvatures at the reference point) associated with their respective orthogonal vectors; see Fig. 3.4. From the principal curvatures we can compute the *mean curvature* $K_M = (K_1 + K_2)/2$ and the *Gaussian curvature* $K_G = K_1.K_2$. These two curvatures (zero) reflect important local properties: the Gaussian curvature defines a developable surface (one of the principal curvatures is zero) while the mean curvature indicates the possibility of a flat surface [Marsh, 2005].

Extensive work has been done in order to estimate geometric features such as normal vectors, distances and curvatures on discrete surfaces. Hamann [1993] used quadratic polynomials for local mesh approximation, the paraboloid fitting of adjacent vertices being done by a least-squares scheme. An almost complete derivation of surface properties by a discrete approximation of curvature tensors for polyhedral surfaces was proposed by Taubin [1995] and Alboul and Echeverria [2005]. Based on differential geometry, discrete operators for triangulated 2-manifolds were proposed by Meyer et al. [2003]; see Fig. 3.6. They derived the first and second order differential attributes (normal vector, mean curvature K_M, Gaussian curvature K_G, principal curvatures, and principal directions) for piecewise linear surfaces such as arbitrary triangle meshes. The definition of the curvatures K_M and K_G will be elaborated more and used in Chapter 3.

2.2.3 Feature invariance

Recognition of objects is one of the most important aspects of human and machine vision. An important aspect is to deal with object constancy: the capacity to recognize an object in all different viewing conditions. It must be possible to extract common features with different viewpoints or even different illuminations of the object. To be able to determine generic characteristics, from one or a class of objects, the visual system faces problems related to size, orientation or both (like deformation). There are, based on neuropsychological evidences, a few theories about constancy. Concerning viewpoint, there are three principal theories: (a) Viewpoint invariant, suggesting that object recognition is based on the structural information of the object. Recognition is independent from the location of viewpoint. (b) Viewpoint dependent, which states that recognition depends on the position of the viewpoint. An object seen from a new, never experienced, viewpoint is more difficult to recognize and recognition is slower. (c) Mixture of previous theories: viewpoint invariant in case of specific objects and viewpoint dependent in case of recognizing object classes - a classification [Tarr and Bülthoff, 1995; Biederman, 1987; Liter and Bülthoff, 1998]. As the human visual system relies on object constancy in a geometric perspective, the features extracted of a polygon mesh must also stay invariant in distinct conditions. The way how features are extracted and their type will determine the robustness of the invariance. If features only concern intrinsic information of the polygon mesh, i.e., they are object-centric, illumination does not affect invariance, and neither does pose (size, location or orientation) because a pre-processing stage of normalization will guarantee invariance. On the other side, if features cover extrinsic information, i.e. view-centric, there will arise problems which are not easily solved. Some approaches combine information from object and view-centric methods, since local and global information can be used to overcome the limitations of each individual method [Al-Osaimi et al., 2008]. The use of features extracted in 2.5D (combination of 2D and 3D space) is justified by psychological evidences which show that the human visual cortex relies on local and global information [Lee et al., 1998; Vogel et al., 2007].

2.3 Recognition process

We have presented a brief overview of involved issues, i.e., sensing, registration and the extraction of geometric as well intrinsic features of 3D objects. However, our goal is to

develop a method for recognizing objects or surfaces. This also involves the identification of similar, but different, objects as members of the same class. Recognition of objects is a process in which the visual system assigns semantic attributes, knowledge about how to use them, and how they associate to others. For the latter functions, the visual system must have some previous knowledge of the object. Recognition is part of a complex system which leads to a higher level of awareness. Perception systems must have a form of memory retrieval; they must store information or events. Accessing memory has two aspects: recognition and recall. Recognition is the relation of an event or physical object with a previous experience or encounter. It involves a comparison process of information in memory. Recall is related to remembering a fact, event or object that is not present (in the sense of retrieving a representation, a mental image or concept), and requires a memory search [Tulving and Thomson, 1973].

Recently large repositories of 3D objects have been created and they are still growing due to the interests of the graphics and vision communities. Regarding retrieval of 3D objects, there has been a lot of research in computer vision devoted to the recognition of 3D shape. It started with 2D images and with image processing, and evolved to 3D through a vast set of techniques conceived for capturing and representing 3D objects [Besl and Jain, 1985; Osada et al., 2002; Bimbo and Pala, 2006; Barra and Biasotti, 2013; Johnson, 1997; Kazhdan, 2004; Fehr, 2009]. The literature provides many theories and methods for recognition and recall of 3D objects, but here we will only describe the work that most influenced our own work.

The spin-images method was conceived for surface matching [Johnson, 1997]. It uses a shape descriptor, constructed on top of a surface representation. A local basis plane is calculated at an oriented 3D point (defined with normal vector) and all the points on the surface are expressed relative to that basis. The mapping (projection) from 3D to 2D begins with the tangent plane to each mesh vertex. Oriented points (p-vertex, P-tangent plane and n-surface normal at vertex) allow to define two coordinates: α, the radial distance to the surface normal, and β, the signed axial distance from the tangent plane to an other point of the surface; see Fig. 2.8.

For each oriented point of the surface it is possible to generate a spin image in 2D space, because of the α and β mapping parameters. The contribution of all vertices, within the support of the spin image, are fed into a 2D binning accumulator, indexed by (α, β), which will show dark zones at concentrations of projected points; see Fig. 2.9.

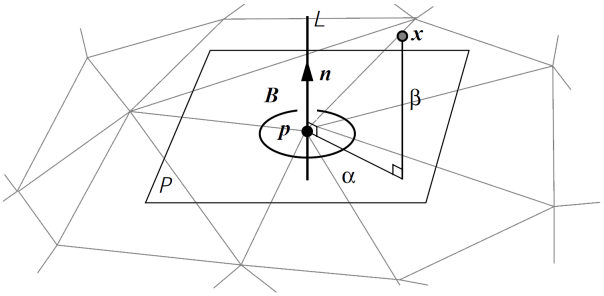


Figure 2.8: From an oriented point (vertex p, tangent plane P and normal n) on the surface, any point x can be mapped on $[\alpha, \beta]$ space. Figure retrieved from Johnson [1997]

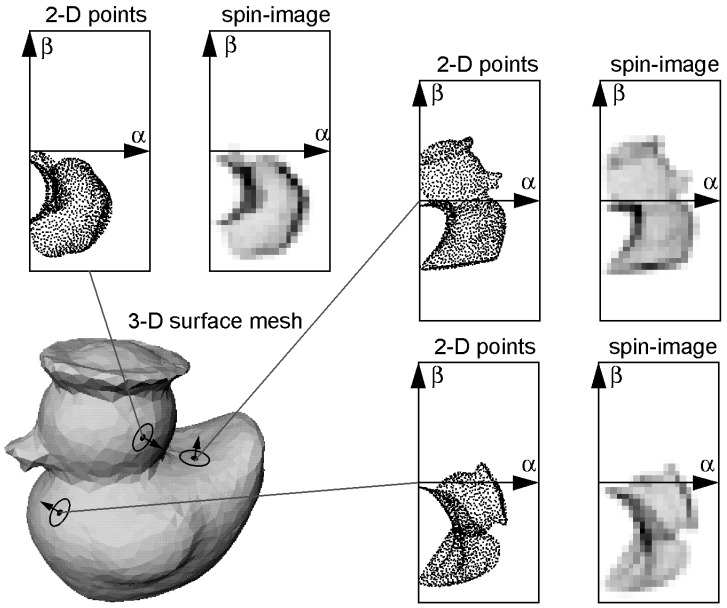


Figure 2.9: Duck model with three oriented points and corresponding spin- maps and images. Figure retrieved from Johnson and Hebert [1999]

Surface correspondence is computed by comparing spin images from all points on both surfaces/objects involved in the matching. When two spin images are similar, a correspondence between the two points is established. After computing the correspondences, they are clustered with geometric constraints and a transform is performed in order to align the two surfaces. To refine the alignment, a variation of ICP (Iterative Closest Point) [Besl and McKay, 1992] is used.

Later, an enhanced method was proposed for object recognition in cluttered 3D scenes [Johnson and Hebert, 1999]. All spin images of each model (they used 20 models and approximately 100 scenes) were computed and, because of the higher correlations of spin images from a same surface, compressed by principal component analysis (PCA). The generated

eigenvectors, designated *eigen-spin images* by the authors, and their eigenvalues lead to a lower dimensional subspace. For the recognition process, scene spin images are compared to compressed model spin images. To this purpose, scene spin images must be projected onto the new object subspace [Johnson and Hebert, 1999].

Based on grid techniques from image processing for 2D recognition, Assfalg et al. [2007] proposed a method to extract an 18-feature vector (signature) from the spin images. They count the number of vertices (surface points) that are in each of three independent zones of the spin-images, crowns ($\beta > 0$ and $\beta < 0$) and circular sectors. To reduce the large number of signatures generated by each model, a model generates only one signature for each vertex of the mesh. The signatures are subjected to fuzzy clustering and the number of clusters, because of the fuzzy nature, is computed for each model.

More recently some work has been done using deep learning, with neural network methods for the recognition and classification of 3D objects [Hinton and Salakhutdinov, 2006; Bengio, 2009; Schmidhuber, 2015]. This new approach was motivated by the growing processing power of multi-core processors as well as the emerging GPUs. The training of neural networks is a very demanding task, it implies finding the optimal weight matrix in a recurrent process. The good results obtained in pattern recognition, in vision, sound (audio, speech) as well as in natural language processing [Hinton et al., 2006; Hamel and Eck, 2010; Wang and Wang, 2014; Dahl et al., 2012], increased interests to extend its use to 3D object classification and recognition. Some research work has been done within the view-centric approach, with and without partial intrinsic information of 3D objects (2.5D). In general, the most significant features of images are extracted and fed into a deep neural network. Significant features are selected according to the type of object. For CAD objects the features are Zernike moments, extracted from 2D images under restricted illumination [Qin et al., 2014]. Socher et al. [2012] used richer 2D images, high-quality RGB and depth images (RGB-D), with a combination of convolutional (CNN) and recursive neural networks (RNN) for feature learning and classifying the RGB-D images. The CNN layer learns low-level, translation-invariant features, which are then fed into the RNN to extract and compose higher-order features. Wu et al. [2015] used 2.5D depth maps, after converting them to volumetric representation (in voxels), to recognize object category or envision the shape, in case of incomplete depth maps. As far as we know, only since last year some work on Deep Learning was applied to fully 3D or volumetric data [Bu et al., 2015; Wu et al., 2015; Xie et al., 2015;

Fang et al., 2015].

2.4 Initial conclusions

From a human perspective, retrieving visual memories is a two-step process: recognition and recall. In computer vision, recognition is usually done by combinatoric matching of relevant features. Some recent work has addressed the neuropsychological model in which local and global features are combined to produce the best results [Fehr, 2009; Mousa et al., 2007; Bustos et al., 2012]. We will develop our approach under this assumption and, in order to develop an invariant object categorization/recognition model, apply the following considerations: (1) Extracted features play an equally important role in biological vision as in computer vision, both for characterizing the most significant aspects that are present and for abstraction of the scene. (2) Object retrieval is a multi-level task with categorization at an early stage [Curran et al., 2002], and the *Gist* concept of a scene can be extended to very complex 3D objects. (3) The existing cortical top-down (and feedback) mechanisms which activate and control visual attention for object recognition [Oliva and Torralba, 2006] can be simulated by a deep learning mechanism. The basic feedforward processing of information for *immediate* recognition tasks, the plasticity and learning can be mimicked by some of the methods from machine learning [Assfalg et al., 2007; Bu et al., 2015; Qin et al., 2014].

More details and references concerning recognition models will be given in the introduction of Chapter 3, notably the representation of 3D objects and by local and global features (descriptors) and the concept of multi-scale. These concepts will be deepened and applied in the subsequent chapters.

Chapter 3

3D object characterization

Abstract: Shape has been studied for a very long time. While philosophers primarily focus on mechanisms of shape perception by humans, engineers are more concerned with shape-related attributes. Features, also referred to as descriptor vectors, are essential in computer vision. They encode distinct properties, and with enough discriminative power they allow to classify or recognize objects. Global features describe entire objects, so they are often used in object classification. On the other hand, local features reflect properties of limited neighborhoods, and they are employed in the recognition of a specific object within a class. The design of shape descriptors must be suited to the task of 3D model retrieval. They must enable an efficient comparison of 3D shapes according to existing metrics. In this chapter we will describe some descriptors used for polygon meshes. The discussion section will promote the Shape Index (SI) as a local descriptor for use in our proposal of a recognition scheme.

3.1 Introduction

The recent advances in software and hardware has led to the use of 3D objects in a variety of fields like gaming, engineering design, television programs and films. A more efficient transmission over Internet and the increasing use of cloud resources require better representations. For some applications it is necessary to search for three dimensional objects that satisfy certain similarity criteria. Nowadays, this searching is done in an imprecise and obsolete way, by the help of input text or by scrolling a hierarchical catalogue. The first method depends on the experience of the user, and the second one is slow and not appropriate for automatic processing. In order to overcome these problems, researchers have investigated the retrieval of 3D shapes, entire objects or parts of them, from a given set of shapes. Despite the non-

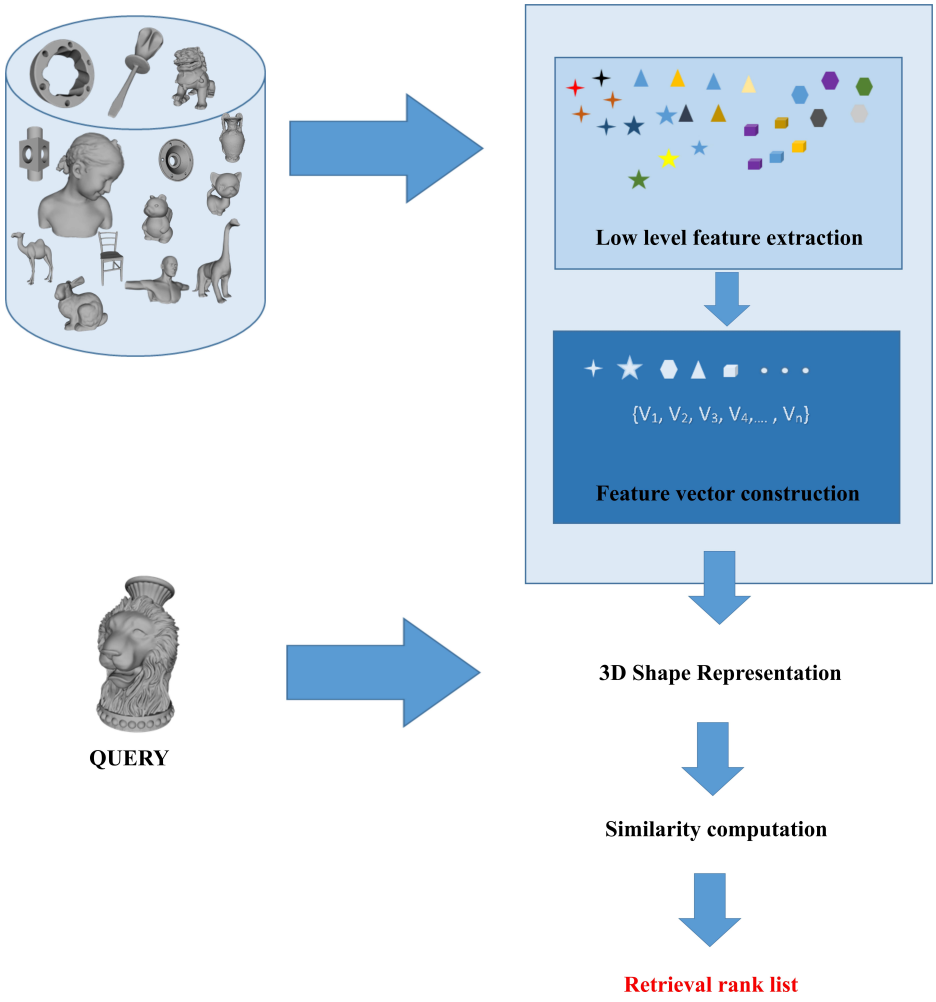


Figure 3.1: Retrieval of 3D objects. Figure adapted from Li et al. [2008].

existence of a universal definition of shape we will use the following one: "*the remaining features after applying scaling, translation and rotational transforms on it*" [Kendall, 1977]. The problems are to restrict the transformations in order to keep the fundamental features, and to define a suitable metric to measure similarity, or dissimilarity, among a set of objects.

Content-based information retrieval (CBR) is an old research field in pattern recognition [Foote, 1999; Qiao et al., 2013; Schroth et al., 2011]. CBR is supported by a metric space and a correlation metric. In computer vision the retrieval of an object from some type of storage is based on the extraction of important features from the borders (surface), and/or interior (volume) of the stored objects; see Fig. 3.1.

A good selection of the most important shape features can reduce the domain where 3D object shape is defined. The mapping selection of the important features must fulfil at least the following properties: (a) size - the feature space should be much smaller then the shape's

space, (b) continuity - small changes in input pattern should produce small effects in the feature space, and (c) cluster consistency - within a local vicinity, the correspondence between input and feature space should be preserved. Since the 1990's, with a growing interest in 3D search engines, 3D shape descriptors have been under investigation. A comprehensive list of surveys on this subject has been produced. The dual approach (view vs. object centric) lead to a distinction of 2D/3D descriptors, local/global context and more specific to the methods used [Weiss, 1993; Tang and Godil, 2012; Tangelder and Veltkamp, 2008; Kazmi et al., 2013; Li and Iyengar, 2014; Heider et al., 2011]. Moreover, shape analysis methods can be divided into two groups, the ones that use information related to the object's surface or boundary, or to its interior or volume [Loncaric, 1998].

Different metrics have been used to measure the similarity of two objects, for example the distance between two feature vectors. Let S be a set of objects, therefore, a metric on S is defined by a function $d : S \times S \to \mathbb{R}$. For all $a, b, c \in S$ the following properties must be satisfied.

$$\text{Positivity} : \; d(a,b) \geq 0 \tag{3.1}$$

$$\text{Identity} : \; d(a,b) = 0 \Leftrightarrow a = b \tag{3.2}$$

$$\text{Symmetry} : \; d(a,b) = d(b,a) \tag{3.3}$$

$$\text{Triangle inequality} : \; d(a,b) + d(b,c) \geq d(a,c) \tag{3.4}$$

The most known distance metric is the Euclidean distance,

$$L_2(a,b) = \left[\sum_{i=0}^{N} (a_i - b_i)^2 \right]^{1/2}, \tag{3.5}$$

which is a special case of the Minkowski distance

$$L_p(a,b) = \left[\sum_{i=0}^{N} |a_i - b_i|^p \right]^{1/p} \tag{3.6}$$

($p = 1$ is also designated Manhattan or city-block distance). In cases where the two sets of features have different dimensions, the Hausdorff metric can be used [Huttenlocher et al.,

1993]. The Hausdorff distance has direction property (asymmetric feature) and is defined as the maximum distance of all points from set A to their closest point in set B. If $A = a_1, a_2, \ldots, a_n$ and $B = b_1, b_2, \ldots, b_m$, the Hausdorff distance is defined as

$$H(A, B) = \max(\overrightarrow{h}(A, B), \overrightarrow{h}(B, A)), \tag{3.7}$$

with $\overrightarrow{h}(A, B) = \max_{a \in A} \min_{b \in B} \| a - b \|$. Distance $\| \cdot \|$ can be any distance measure, e.g. the Euclidean distance. The function $\overrightarrow{h}(A, B)$ measures the distance from A to B (directed Hausdorff distance). If $\overrightarrow{h}(A, B) = d$, then each point of A must be within distance d of any point of B. Since $H(A, B)$ is the maximum of $\overrightarrow{h}(A, B)$ and $\overrightarrow{h}(B, A)$, it measures the degree of mismatch between two sets. Another metric is based on the statistical dependence between the two sets of features. Distance correlation, also referred to as Cosine distance, measures the similarity of two feature vectors by

$$C(A, B) = \frac{\sum_{i=1}^{N} (a_i - \overline{a})(b_i - \overline{b})}{\sqrt{\sum_{i=1}^{N} (a_i - \overline{a})^2 \sum_{i=1}^{N} (b_i - \overline{b})^2}}, \tag{3.8}$$

with N the number of features and $\overline{a}$, $\overline{b}$ the means of both feature vectors. The value ranges from -1 (opposite) to 1 (exactly the same). The value 0 indicates that there is no correlation (decorrelation, the two sets are orthogonal).

Using Kendall's [1977] shape definition, for a retrieval/recognition system this means that selected shape features must be invariant under some affine transformation. They must be unaffected by rotation, translation and scale transforms. Acquiring features which are invariant in view-centric mode is a complex task because of object pose, perspective projection and the intrinsic parameters of the camera. The use of concepts from invariance theory can lead to a more systematic approach [Forsyth et al., 1991]. Formal invariance theory, which is an abstract algebra on actions of groups, deals with the subject of defining an explicit description of polynomial functions that are invariant to transformations from linear groups. Useful advances for 3D shape classification/recognition were introduced by geometric invariance theory, especially the *moduli spaces* concept [Mumford et al., 1994]. Basic concepts of invariance theory for vision are detailed by Forsyth et al. [1991] and by Weinshall [1993].

Since invariant features are defined in the context of a particular transformation, translation, rotation and isotropic scaling are classified by Weinshall [1993] as rigid invariants to the action of a similarity transform group. Apart from being invariant, the descriptors should

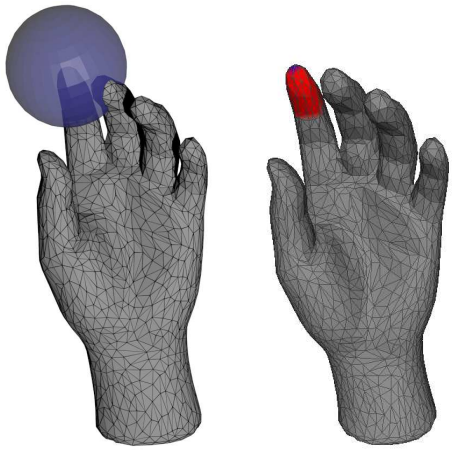


Figure 3.2: Euclidean (left) and Geodesic (right) distances to define a local neighborhood at the top of the index finger.

be robust to noise and sampling errors. We stress the importance of differential invariance, i.e., invariant functions of the position and derivatives at nearby surface points (curvature and torsion are differential invariants under Euclidean transformations). See the 4th and 5th examples in Forsyth et al. [1991].

The rest of this chapter is structured as follows: Sections 3.2 and 3.3 introduce concepts that underlie the choice of features that best represent shape. Section 3.4 presents the concept of multi-scale mesh representation. We conclude with a discussion in Section 3.5.

3.2 Local features

3D objects often have some interesting localized characteristics. To have discriminative power, local descriptors should reflect the surface shape on a limited domain. They can be restricted to a vicinity in two ways: by a Euclidean distance r (surface features are computed within a sphere of radius r) or by a Geodesic distance r (measured *over* the surface); see Fig. 3.2.

To compute local descriptors inside a small vicinity, both distances can be used because they yield similar results [Heider et al., 2011]. Local descriptors are more sensitive to noise than global ones because they are designed to capture local detail. *Outliers* within the vicinity will distort the associated descriptor. The introduced bias will be reduced as the vicinity becomes larger. This is why global descriptors are more robust to noise than local ones. However, some local descriptors can also be used, in an integral form, as global descriptor

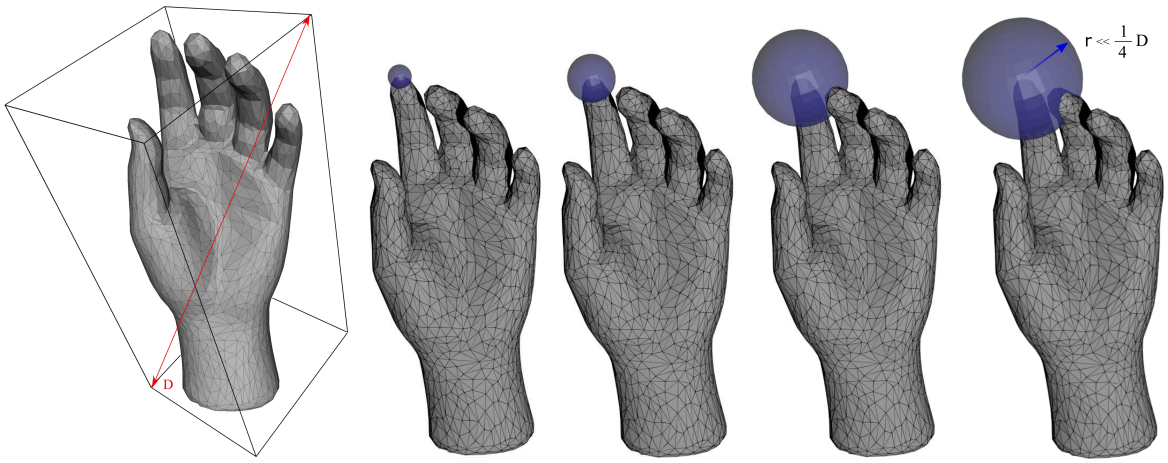


Figure 3.3: Hand model (left) and increasing sphere with radius of 0.05, 0.10, 0.20 and 0.25 times the diagonal of the Hand's bounding box. The intersection curves are a function of the radius.

[Pottmann et al., 2009]. A local descriptor must be invariant to geometric changes, it must be *quite* invariant to translations and rotations, but perhaps less to scaling and occasionally to bending. In addition to geometric invariance, a descriptor must resist to geometric noise (vertex displacements), mesh noise (changes in vertex connectivity) and global topological noise (handles and tunnels). Hence, it must *reliably describe* the shape of the local surface around a point. Usually, local descriptors should hold an expressive comparison function, one that reflects the noticed shape changes. Heider et al. [2011] presented a survey of local shape descriptors and their evaluation.

Local descriptors, in object-centric mode represented by a mesh, are generally classified into one of these groups: ring-based descriptors or expanding descriptors. For ring-based descriptors, the values are locally sampled with a neighborhood restriction (1-ring, 2-ring, ...). Concentric spheres or geodesic rings are used to define a local domain and the sampling is done over a metric that has a meaningful representation of the local shape. The *Blowing bubbles* method [Mortara et al., 2003], which uses concentric spheres with their center at a vertex of the mesh, provides a multi-scale platform. Local features, surface curvature and interior information of the shape are characterized by the intersection of the sphere surface (with increasing radius; each radius is a scale) and the object surface. The intersection curves, at specific radii/scales, can be classified according to the number of connected components; see Fig. 3.3. Curvature maps, proposed by Gatzke et al. [2005], use the surface curvatures to establish a distinct signature. To gather the information around a vertex they adopt a

Geodesic fan neighborhood: geodesic distances are defined over the mesh. Geodesic fans represent a structure around the vertex, with equally spaced spokes. Point samples are also equally spaced along the spokes and each set of points which is equidistant from the central vertex is considered as a ring. Each sampled point can be referenced by a local geodesic polar map. Local features are defined in the form of *Curvature maps*. They are build in one and in two dimensions. 1D curvature maps are obtained by plotting the distance, of an n-ring, versus the accumulated curvature values on that ring. 2D maps take into account the orientation of the spokes by accumulating values sampled on the spoke.

Traditionally, curvature has been used as shape descriptor. Studying the geometry of surfaces is based on differential properties of the surface, which allow to intrinsically discriminate infinitesimal neighborhoods. Formally, regular surfaces [1] allow to compute some geometric values, such as lengths of curves on the surface, delimited areas or angles between curves by the First and Second fundamental forms, see Equations 2.1 and 2.2. At a point of a regular surface, it is possible to compute its principal curvatures. They are the eigen values of the shape operator (II *form*) and they show how much the surface bends at that point. More information and formal definitions of surface differential geometry can be found in do Carmo [1976] and Spivak [1999].

Classification based on local properties of a surface can be supported by the principal curvatures, the maximum and minimum curvatures [Lipschutz, 1969]. To extend the previous definitions we will introduce useful concepts to describe these curvatures in a practical manner. Consider a surface S, a 2-manifold, and a point P on it. The tangent plane at P is defined by the normal vector N. The intersections of orthogonal planes (through N), with S, define a set of planar curves. The curvature of each of them is defined as the inverse of the curvature's radius, see Fig. 3.4. Consider K_1 as the maximum curvature of all planar curves (containing P and N) and K_2 as the minimum. Gaussian curvature can be expressed as $K_G = K_1.K_2$ and Mean curvature as $K_M = \frac{1}{2}(K_1 + K_2)$. The Gaussian curvature allows to infer, in a rough way, the relation between surface and the tangent plane. For parabolic as well as cylindrical surfaces K_G will be zero because one of the principal curvatures will be zero. Elliptic surfaces have the curvature's center on the same side of the surface and all K values are either positive or negative. In case of more complex surfaces, i.e. hyperbolic, the centers of K_1 and K_2 will be on opposite sides (the saddle case). Based on the principal cur-

[1] $S \in R^3$ is regular surface if for each $p \in S$ exists a neighborhood V in R^3 and a map M: $U \leftarrow V \cap S$ ($U \in R^2$). M is differentiable, a homeomorphism, and it's differential is one-to-one.

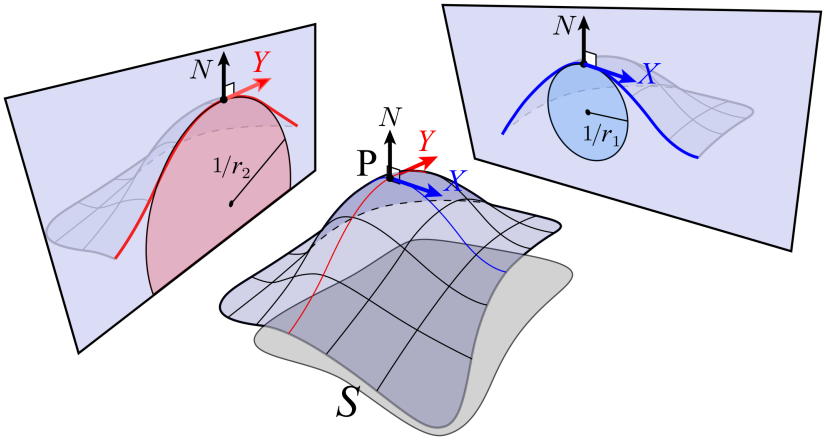


Figure 3.4: Top: principal curvatures, minimum (left) and maximum (right). Bottom: tangent defined by normal vector N on point P. Image adapted from brickisland.net [2015].

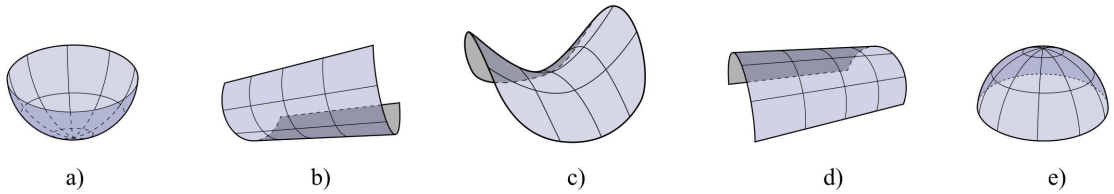


Figure 3.5: Shape Index: a) concave ellipsoid with SI $\in [-1, -5/8)$, b) concave cylinder with SI $\in [-5/8, -3/8)$, c) hyperboloid with SI $\in [-3/8, 3/8)$, d) convex cylinder with SI $\in [3/8, 5/8)$ and e) convex ellipsoid with SI $\in [5/8, 1]$. Designation as used by Cantzler and Fisher [2001].

vatures, and in pursuing a more descriptive indicator, the *Shape Index* (SI) and *Curvedness* (C) were proposed. SI is defined as

$$\mathrm{SI} = \frac{2}{\pi} \cdot \arctan\left(\frac{K_1 + K_2}{K_1 - K_2}\right) \quad K_1 > K_2, \tag{3.9}$$

and

$$\mathrm{C} = \sqrt{\frac{K_1^2 + K_2^2}{2}}. \tag{3.10}$$

These two parameters, in combo, allow to classify local patches. SI, varying in the range $[-1, +1]$, reflects local shape; see Fig. 3.5, and C is a positive value that describes the magnitude of curvature at the surface point [Koenderink and van Doorn, 1992; Phillips, 1996]. The drawback of SI is that it is undefined in planar areas, but there C is well defined and equals zero. A study comparing the performance of Mean/Gaussian curvatures *vs.* Shape Index/Curvedness suggested that SI/C has some advantages at lower values, in quasi-planar zones [Cantzler and Fisher, 2001].

All previous theory based on differential geometry is defined on continuous surfaces. The move from the continuous to the discrete domain brings a few extra problems which are related to the sampling process. The quality of representation depends on voxel size, or in case of polyhedral surfaces (meshes) on mesh quality, i.e., how smooth or noisy the mesh is. Apart from numerical errors, computational approximations are acceptable if the discrete data are regularly distributed, as in range images. For triangular meshes, the surface representation demands more complex computations to estimate the curvature at a vertex without a smooth approximation. Estimation of principal curvatures of meshes has been an important research topic [Chen and Schmitt, 1992; Taubin, 1995]. A simple way to obtain the Gaussian curvature of a vertex v, in a polyhedral surface, is by evaluating the vertex angle excess $K_G(v) = 2\pi - \sum_{i=1}^{m} \theta_i(v)$ with θ_i the inner angles of the m triangles around vertex v [Polthier and Schmies, 1998]. However, these authors did not take into consideration the local tessellation of the mesh, nor the area or type of triangle. Meyer et al. [2003] used the weighted area of surrounding facets in the 1-ring to compute the Mean and Gaussian curvatures. The area A_j from each triangle around the vertex is computed, taking into account whether they are obtuse or not, and the area is delimited by the barycenter, or circumcenter of each triangle. Their formulas are

$$K_G(V_i) = \frac{2\pi - \sum_{j=1}^{m} \theta_j}{\sum_{j=1}^{m} A_j} \tag{3.11}$$

and

$$K_M(V_i) = \frac{1}{2A} \sum_{j \in N_1(V_i)} (\cot\ \alpha_{ij} + \cot\ \beta_{ij})\,(V_i - V_j)\,, \tag{3.12}$$

In a perfectly planar region, curvature $K_G = 0$ since the sum of the angles θ_j will be 2π. See Fig. 3.6, and for a more detailed explanation we refer to Meyer et al. [2003].

3.3 Global features

Global features must be able describe an entire object, i.e., all the shape information. The problem is to create a unique feature vector with relatively low computational complexity and which is compact. In general they can represent complex object information but they are weak in discriminating details. Methods for extracting global features are under research since the beginning of the 1990's. Some of them are 3D versions of methods as used in image processing, like the area, volume, moments and Fourier coefficients [Zhang and Chen,

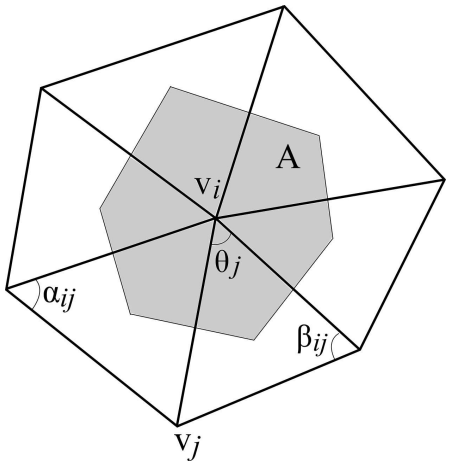


Figure 3.6: 1-Ring neighborhood of vertex v_i. All triangles around it are non-obtuse and vertices v_j have one edge to v_i. The grey area (A) is Voronoi region used as weight in K_M and K_G discrete curvature operators.

2001; Restrepo et al., 2012]. A global feature can be extracted by iteratively smoothing of the polygonal mesh. Applying a smoothing filter forces vertices to change their position. Vertices in high-curvature regions tend to have a bigger displacement than those in low-curvature regions. This approach was proposed for the alignment of point-based surfaces, by smoothing the *surfel* cloud using MSL, Moving Least Squares [Li and Guskov, 2005]. Other coarse features can be based on the bounding box, the fractional occupancy (ratio of object area/volume to bounding box area/volume), cords and statistical moments [Paquet et al., 2000]. The notion of distributions, from statistics, provides an alternative solution. Instead of computing the similarity of two objects by a direct comparison of global features, the object distributions can compared [Osada et al., 2002; Ohbuchi et al., 2002; Mahmoudi and Sapiro, 2009]. Spatial mapping can provide additional information. In this approach, the captured features carry information on the physical locations of an object. Usually, concentric shells and sectors around the object's centroid are used. Then, similarity is computed using histograms gathered at uniformly sampled concentric shells and sectors [Ankerst et al., 1999].

An extension of concentric shells was proposed by using a ray-based descriptor vector. This method uses the rays from the centroid to the last surface intersection. The measured distances serve to compute spherical harmonics, similar to the Fourier transform of a 1D or 2D signal but on sphere [Vranic and Saupe, 2001]. This method needs pose normalization because it is not rotation invariant. To overcome this limitation, the spherical function is

described in terms of its energy at different frequencies. Since these values do not change when the function is rotated, the resulting descriptor is therefore rotation invariant [Kazhdan et al., 2003].

3.4 Multi-scale space

Instead of describing an object at a fixed (like the real) scale some features have a multi-scale nature. For example, when we approach a tree from afar, more details and a more precise view will emerge. In image processing, the reverse process is applied by successively smoothing an original image. The use of a controlled smoothing filter allows to remove noise or other high-frequency structures, *controlled* meaning a filter with parameters which influence its behavior.

The concept of multi-scale representation is commonly used in computer vision for the design of methods for automatic analysis and extraction of data from real-world data. Extracting information from image data depends on the type of underlying structures in the data and the operator structure and resolution (size).

In image processing, basic problems are which operator to use, where and when to apply it, and its size (scale). If these questions are not properly addressed, it is very difficult to make sense of the operator response [Lowe, 1999; Burger and Burge, 2013]. It was necessary to develop a systematic theory for describing image structures at different scales. This has resulted in a mature theoretical scale-space framework. The structures at coarse scales should be simplifications of the corresponding structures at finer scales; see Fig. 3.7 [Koenderink, 1984; Lindeberg, 1996]. In case of polygonal meshes, the multi-scale concept can be applied in different forms, for example by using concentric spheres, shells or geodesics to define scales [Mortara et al., 2003; Sun et al., 2009], or by applying some smoothing transformation over the mesh [Lam and du Buf, 2009, 2011a,b]. For 3D object recognition, the Multi Scale Shape Index (MSSI) was proposed by Bonde et al. [2013]. It is extracted from a point cloud without any structured mesh, and the features are obtained by combining scale space filtering and Shape Index. After calculating a characteristic scale at each 3D point, then the values are used to estimate principal curvatures. In fact, the MSSI is a triplet of shape index, curvedness and the characteristic scale [Koenderink and van Doorn, 1992]. We will explore the concept of 3D multi-scale polygonal meshes in Chapter 5.

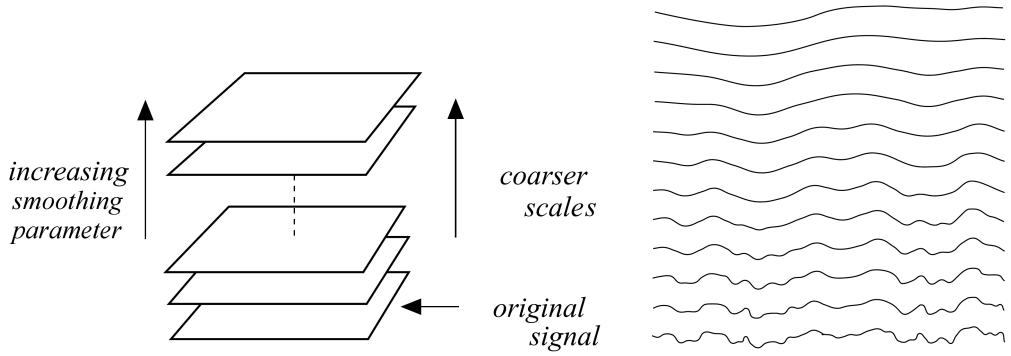


Figure 3.7: Left: 2D multi-scale scheme. Right: a 1D signal (bottom) will show less details at coarser scales (top).

3.5 Discussion and outlook

In this chapter we revised important issues related to three-dimensional object representation. In addition to the formal definition of *shape* [Kendall, 1977], we presented the concepts of local and global descriptors. The First and Second Fundamental Forms allow to compute areas, main curvatures as well the lengths of curves on surfaces. The discrete operators, based on differential geometry, permit to extract surface features. It has been shown that global features based on differences of volumes, or areas, under a smoothing transformation, have enough descriptive power to classify 3D objects [Lam and du Buf, 2009, 2011b,a]. This will be detailed in Chapter 4.

Local features have the advantage of being restricted to a neighborhood, while global ones describe entire objects with less emphasis on local structures. Exploring a 3D multi-scale space scheme may allow us to obtain knowledge regarding the consistency of local information within the multi-scale space. Chapter5 will go deeper into the multi-scale concept.

Chapter 4

Categorization and recognition of 3D objects

Abstract: In this Chapter we present a method for retrieving 3D polygonal objects by using two sets of multiresolution signatures. Both sets are based on the progressive elimination of object details by iterative processing of the 3D meshes. The first set, with five parameters, is based on mesh smoothing. This mainly affects an object's surface. The second set, with three parameters, is based on difference volumes after successive mesh erosions and dilations. Characteristic feature vectors are constructed by combining the features at three mesh resolutions of each object. In addition to being invariant to mesh resolution, the feature vectors are invariant to translation, rotation and size of the objects. The method was tested on a set of 40 complex objects with mesh resolutions different from those used in constructing the feature vectors. By using all eight features, the average ranking rate obtained was 1.075: 37 objects were ranked first and only 3 objects were ranked second. Additional tests were carried out to determine the significance of individual features and all combinations. The same ranking rate of 1.075 can be obtained by using some combinations of only three features.

4.1 Introduction

The processes of classification and recognition of 3D objects has become a big challenge in computer vision. It is driven by the enormous number of 3D objects available in databases. CAD applications compose complex models with corners, edges and joints from simpler ones with smooth surfaces. In medicine 3D complex protein structures play an important role in pharmacology and related areas. Orthopaedics began to use 3D models to plan final prostheses. The World Wide Web, as for the game industry, started to embody 3D

models. The computer vision challenge is to develop reliable methods for classification and recognition of 3D objects or shapes. Usually, due to the existing work already done in image processing, many researchers use 2D views of 3D objects [Rodrigues et al., 2012a, 2014]. However, interest in a truly 3D analysis emerged in the last decade [Bustos et al., 2005, 2012; Tangelder and Veltkamp, 2008; Zhou et al., 2015; Yao et al., 2016; Kazmi et al., 2013]. Similarity search and object categorization are two methods that can be used to deselect unrelated objects from a database, i.e., prior to exact object recognition which can be very tedious because of all possible variations in size, position and orientation that may exist between the object that we want to recognize in a database.

Similarity analysis is not performed in a strict manner. Shape comparison, global or local, is done by computing a set of features (feature vector FV) of a query object and by comparing it with all FVs of the known objects in the database. FVs can be determined by simple methods that use some basic geometric properties, e.g., bounding box, area/volume ratio and eccentricity, or by complex methods which involve mathematical formulations such as the curvature distribution of the sliced volume, spherical harmonics and 3D Fourier coefficients [Vranic and Saupe, 2001; Sijbers and Dyck, 2002; Assfalg et al., 2007; Kazhdan et al., 2003]. The intrinsic nature of an object may disable some methods, and other methods may be more appropriate to extract a FV. Methods based on spherical harmonics or 3D Fourier coefficients are not appropriate for concave objects with certain indentations, which are called non-star-shaped objects. Other methods have problems with open, non-closed objects. To overcome some of the restrictions it is possible to combine two or more methods. Using only one method, or one geometric feature, can produce very similar FVs in case of unrelated objects. However, by applying several methods, the chance that all FVs are similar becomes smaller.

Object categorization is more complex then object recognition. It is more difficult because it is possible to define distinct levels of categorization. For example, recognizing our dog in a pack is fairly easy. An object could be classified as being an animal, with four legs, or as a dog which is very similar to our dog, but *not our dog*. The process by which humans, in infant age, start to discover *things* is typical. If something furry with four legs is a dog, then cats are also dogs until some features are added to create a new category. This early phase, before each category has evolved, is defined as pre-categorization [Rodrigues, 2008].

We believe that studying the process of 3D object categorization can provide some clues

to construct templates, keypoints, saliency maps and segmented surfaces [Gal and Cohen-Or, 2006; Kalogerakis et al., 2010]. Categorization takes place in some kind of hierarchical form, from coarse to fine. Information at coarse scales can be used for a first, quick but rough categorization, after which categorization is refined by using information at progressively finer scales until the object is recognized. Using keypoints and saliency maps determined in multi-scale space can provide useful information for constructing a coarse-to-fine categorization/recognition system.

Mesh smoothing allows to reduce noise. After a smoothing operation it is also possible to reduce the number of vertices (mesh resolution) by a re-triangulation of planar or *quasi*-planar areas [Lam et al., 2001]. Apart from reducing the number of triangles, the smoothing process can eliminate structural information, precisely that information which is typical for a characteristic shape. Smoothing of the principal components was used for 2D shape classification by Glendinning and Herbert [2003]. They worked in the frequency domain, after a circular sub-sampling process, to characterize closed curves. After getting the spectrum of the underlying curve of interest, a smoothing of its principal components was applied to the Fourier spectral coefficients.

In this chapter we will use the smoothing filtering to explore three methods that can extract, at a global level, features that are invariant to pose, size and rotation, suitable to categorization and recognition. In the next section we will briefly describe existing methods for 3D object categorization and recognition. Section 4.3 will present the new methods and experimental results. The final section is a discussion with lines for future research.

4.2 Existing 3D methods

There are many approaches to 3D object recognition, see Tangelder and Veltkamp [2008], Bimbo and Pala [2006]; Bustos et al. [2005] and Kazhdan [2004] for reviews. The interest in this subject remains high, technology will still advance, providing a massive use of 3D models and leading to new methods and algorithms for processing 3D data. Because of the 3D data nature, i.e. the shape representations, the algorithms can work in very diverse ways, with point, surface or volume models. These representations cover a wide range of techniques: (a) parametric or implicit surface representations, for example; or (b) volumetric representations that can be in any partition of three dimensional space, Cartesian grids, BSP (binary space partition) trees, octrees, kd-trees and so on [Botsch et al., 2010]. The majority

of 3D representations used have been conceived for interactive visualization or up to physical rendering with the latest 3D printers. The simplest representations just incorporate geometric information, sometimes with appearance attributes. Generally they are represented by polygonal meshes, in a *soup* form, that can be *watertight* (or not, e.g., surface data from range image sensors). There are many factors that influence the performance of categorization and recognition methods. For example, the point cloud representation requires a pre-processing step in order to determine local relations between points, for computing local curvatures. Despite all the progress achieved there is no geometry-based method which uses shape deformation as features to be used as similarity measures. Below we will describe and explore our approach which is based on shape deformation. Our method can be classified as global and volumetric.

We now succinctly describe the approaches which, to some degree, are related to our own approach. Deformation similarity is one of the methods used to compare planar shapes. Usually they measure the amount of *energy* required to deform a shape until it matches the comparison shape. Assfalg et al. [2006] start with a preprocessing step to smooth and simplify the object mesh. Then the 3D object is projected onto 2D curvature maps, and final comparison is done by using the 2D curvature maps. Mathematical morphology, developed in middle of the 1960s [Serra, 1982; Matheron, 1975], was intended for geometric analyses of shapes and textures. Despite all theoretical developments and a generalization to 3D [Ghosh and Haralick, 1993], the work done in 3D is rather scarse and mostly limited to three-dimensional surfaces. For example, Jackway [1995] developed an approach for the recognition of 3D objects in range data through the matching of local surfaces. Lee et al. [2005b] analyzed the composition of 3D particle aggregates (granularity) by processing range image data with a sphere as structuring element over one hemisphere of the particles.

4.3 New methods

In the development of our new methods we used a set of 40 models, each one represented with four different mesh resolutions. The models were selected from the AIM@SHAPE [2004] database. This database has high-definition objects which can be converted to other mesh resolutions by means of one parameter between 9.9 and 5.5 (maximum and minimum mesh resolutions). The models were downloaded in PLY format. They are 2-manifold, *watertight* and regular. Figures 4.1 and 4.2 show the diversity and examples of different resolutions.

Table 4.1 lists all objects with their mesh resolutions: the first three resolutions are used for creating the training set FVs, the fourth one as test object in similarity search. To ensure invariance under translation and scale (size), the models were normalized to the unit sphere (radius 1.0) after the origin of the models was translated to the center of the normalization sphere. Rotation invariance is ensured by the global nature of our FVs [Vranic, 2004].

4.3.1 Multi-resolution signatures

First experiment:

Our method is based on two sets of features. The first set results from mesh smoothing, the second set is based on a controlled volumetric variation of the mesh. Mesh smoothing allows to reduce noise as well as local mesh resolution, for example by decreasing the mesh size by re-triangulation of planar areas. The idea is related to iterative and adaptive (nonlinear) mesh smoothing in 3D, i.e., smoothing in quasi-planar regions but not at sharp edges [Lam et al., 2001]. In case of controlled volumetric variation, erosion and dilation from mathematical morphology are applied in order to gradually eliminate object details. See Appendix A for a short introduction to mathematical morphology.

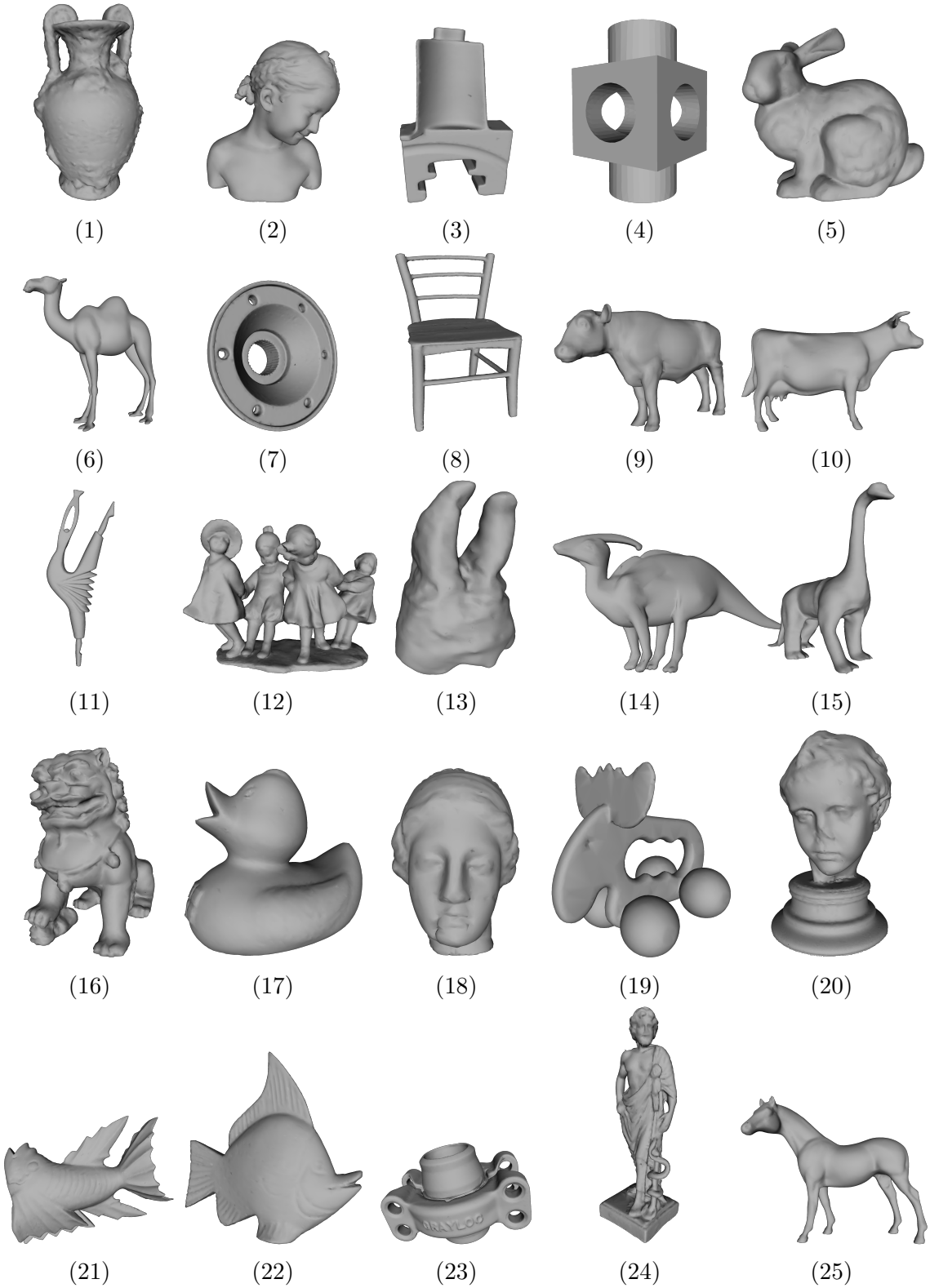
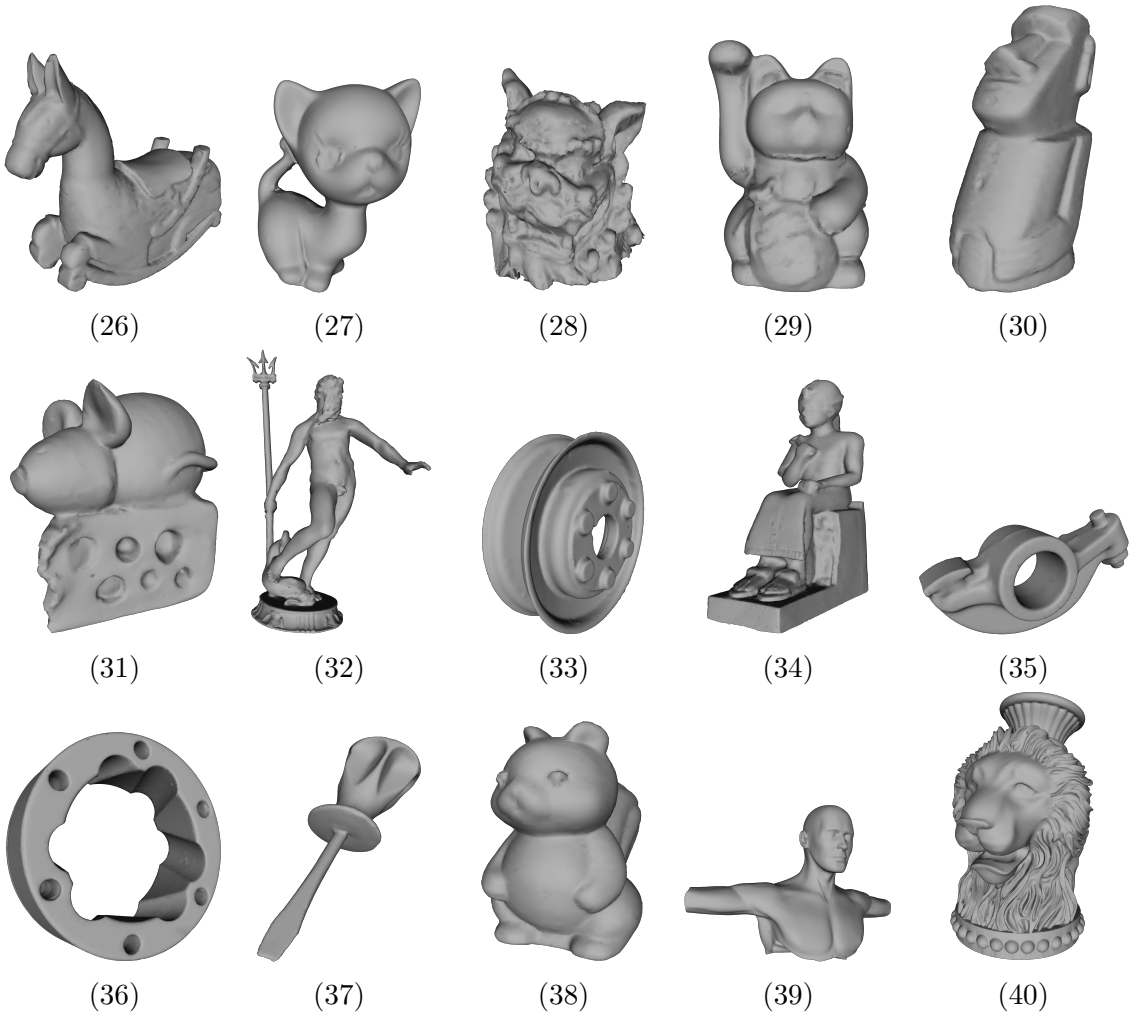


Figure 4.1: Images from objects used in our test experiments. The names of all objects are listed in Tab. 4.1.



(26) (27) (28) (29) (30)

(31) (32) (33) (34) (35)

(36) (37) (38) (39) (40)

Figure 4.1: Continued.

Mesh smoothing

We simply apply a linear filtering which will smooth the mesh at all vertices. It starts by eliminating very sharp object details like in- and protruding dents and bumps. Then, after more iterations, less sharp details are eliminated. The sum of the displacements of all vertices, in combination with the contraction ratio of the surface area, yields a quadratic function which can characterize the model quite well. Let $V_i, i = 1...N$, be the object's vertex list with associated coordinates (x_i, y_i, z_i). The triangle list $T(V)$ can be used to determine the vertices which are at a distance of one edge to another vertex, i.e., all direct neighbor vertices connected to V_i. If all neighbor vertices of V_i are $V_{i,j}, j = 1, n$, the centroid of the neighborhood is $\bar{V}_i = \frac{1}{n}\sum_{j=1}^{n} V_{i,j}$. Each vertex V_i is moved to $\bar{V}_i$, with displacement $\bar{D}_i = ||V_i - \bar{V}_i||$, and the total displacement is $D = \sum_{i=1}^{N} \bar{D}_i$. The entire procedure is repeated only 10 times, because we are mainly interested in the deformation of the object at the start,

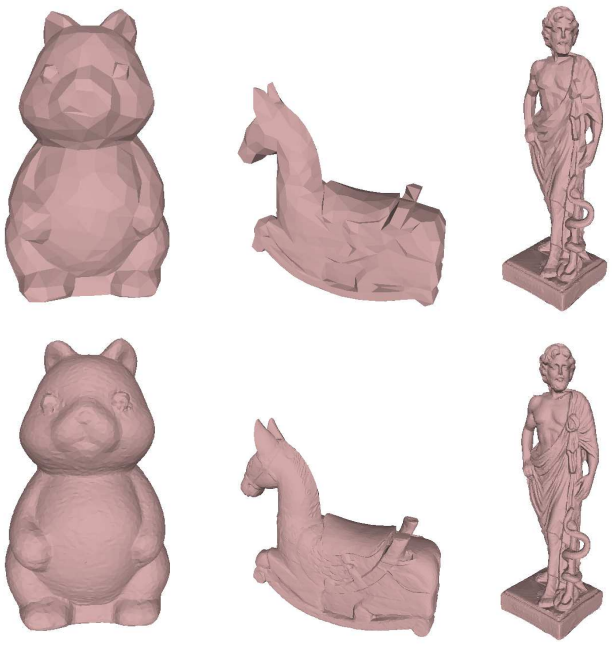


Figure 4.2: Examples of different resolutions. Squirrel (left), IsidoreHorse (center) and GreekSculpture (right). Low resolution at top and high one at bottom.

Table 4.1: All 40 models with their mesh resolutions; the first three are used in resolution-invariant feature extraction, the last one is used in similarity search.

N	Model	Resolutions	N	Model	Resolutions
1	Amphora	6.5; 7.5; 9.5; 8.0	21	Fish	6.0; 7.5; 9.9; 8.0
2	Bimba	6.0; 8.5; 9.5; 8.0	22	FishA	6.0; 7.5; 9.9; 7.0
3	Blade	6.0; 7.5; 9.9; 8.0	23	Grayloc	6.0; 7.5; 9.9; 7.8
4	Block	5.0; 6.5; 8.0; 8.5	24	GreekSculpture	6.5; 7.0; 7.7; 8.5
5	Bunny	6.5; 7.5; 9.9; 8.0	25	Horse	6.0; 7.5; 9.9; 8.0
6	CamelA	6.0; 7.5; 9.9; 7.8	26	IsidoreHorse	6.0; 7.5; 9.9; 7.0
7	Carter	6.0; 7.5; 9.9; 7.3	27	Kitten	6.0; 7.5; 9.9; 7.3
8	Chair	6.0; 7.5; 9.9; 6.9	28	Liondog	6.0; 7.5; 9.9; 8.0
9	Cow2	6.0; 7.5; 9.9; 8.9	29	Maneki	6.0; 8.8; 9.8; 7.5
10	Cow	6.0; 6.4; 9.9; 7.1	30	Moai	6.5; 8.5; 9.5; 9.7
11	Dancer	6.0; 7.5; 9.9; 7.7	31	Mouse	6.0; 7.5; 9.9; 7.8
12	DancingChildren	6.0; 7.5; 9.9; 6.8	32	Neptune	6.0; 7.5; 9.9; 7.6
13	Dente	6.0; 7.5; 9.9; 7.0	33	Pulley	6.0; 7.5; 9.9; 7.0
14	Dilo	6.0; 8.5; 9.6; 7.7	34	Ramesses	6.0; 7.5; 9.9; 8.0
15	Dino	6.0; 8.3; 9.7; 7.7	35	Rocker	6.0; 7.5; 9.9; 7.1
16	Dragon	6.0; 8.0; 9.5; 7.7	36	RStage	6.0; 7.0; 9.0; 9.5
17	Duck	6.0; 7.5; 9.9; 6.7	37	Screwdriver	6.0; 7.5; 9.9; 7.0
18	Egea	7.4; 7.9; 9.5; 8.7	38	Squirrel	6.0; 7.5; 9.9; 7.2
19	Elk	6.0; 7.5; 9.9; 7.9	39	Torso	6.0; 7.5; 9.9; 7.7
20	Eros	6.0; 7.5; 9.9; 6.5	40	Vaselion	6.0; 7.5; 9.9; 7.5

when there still are many object details, and later iterations do not add useful information anymore. Hence, displacements are accumulated by $A_l = \sum_{m=1}^{l} D_m$ with $m = 1...10$. In order to obtain invariance to mesh size, in each iteration m the displacement D_m is corrected using

$$D_m := D_m \cdot \frac{NP_m \cdot N}{A_{10} \cdot S_m}, \tag{4.1}$$

with N the total number of vertices, NP_m the number of participating vertices in non-planar regions which contributed to the displacement, S_m the surface of the object (sum of all triangles) after the smoothing step, and A_{10} the final, maximum accumulated displacement after all 10 iterations. Then the curve of each object and each mesh resolution is further normalized by the total contraction ratio defined by S_{10}/S_0 (final surface and original surface), and the three curves with 10 data points are averaged over the three mesh resolutions. In the last step, the averaged A_l is least-squares approximated by a quadratic polynomial in order to reduce 10 parameters to 3. Figure 4.3 shows representative examples of curves A_l. In contrast to the second method that will be described below, no re-triangulation of the

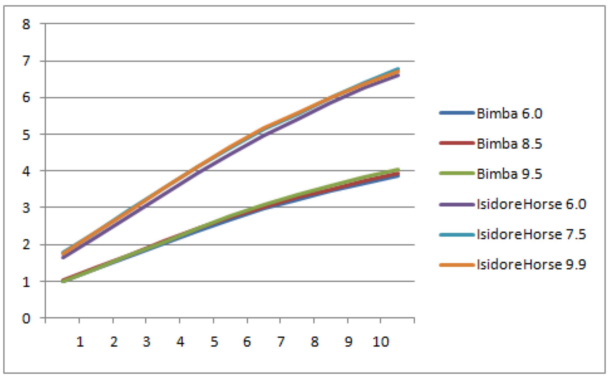


Figure 4.3: Characteristic curves from mesh smoothing of the Bimba and IsidoreHorse objects.

mesh after each iteration is done, i.e., the number of vertices and triangles remains the same. Figure 4.4 shows a model and the influence of mesh smoothing; as the number of iterations increases, the smoothing filtering reduces the level of details.

Dilation and erosion

The second method is based on the concept of 3D fractal dimension by applying a sphere as structural element with increasing radius. A sphere is applied to the model, its origin placed at each vertex. This yields two surfaces: the dilated surface grows and the eroded one shrinks as a function of sphere radius, and both will show less object detail. Instead

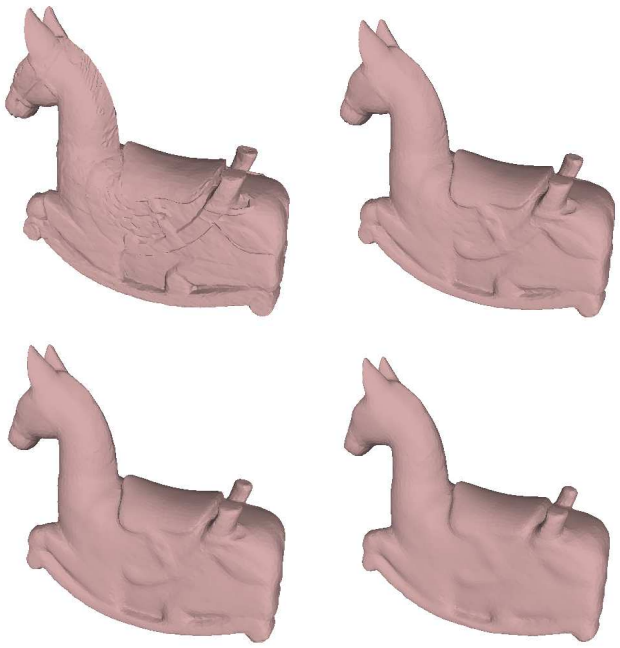


Figure 4.4: Mesh smoothing applied to IsidoreHorse. From left to right and top to bottom: original and smoothed meshes after 3, 6 and 10 iterations.

of computing the fractal dimension, we compute the volume between the two surfaces as a function of sphere radius in order to obtain characteristic curves, like the ones described in the previous section. The rate of growth (increasing radius) of the sphere is related to the mesh resolution of the model. Therefore, before computing the volumes, we eliminate vertices which are inside the neighborhood defined by the sphere with the radius used in the dilation-erosion process. If ΔL is the difference between the maximum and minimum edge length of a model, i.e., $\Delta L = L_{\max} - L_{\min}$, then $\Delta R = 0.05\Delta L$. Hence, the radius at iteration m is $R_m = m\Delta R$, which results in volumes V_0 (the volume of the original model) and V_m (the volume between dilated and eroded models after iteration m). For obtaining invariance to mesh size, we apply

$$V_m := V_m \cdot \frac{V_0}{R_m} \quad ; \quad m = 1, 2, ... \tag{4.2}$$

As can be seen in Fig. 4.5, the dilation-erosion curves are quite similar, for different mesh resolutions, at the start of the process, but then start to diverge when the radius becomes too big and noise is introduced into the model. Therefore we averaged the V_m of the three mesh resolutions and only included in the FV two parameters: V_0 and V_2. The characteristic curves have an almost linear behavior in the first three iterations. Figure 4.6 shows the Bimba model after erosion and dilation (the deformation is biggest at the base and the

head).

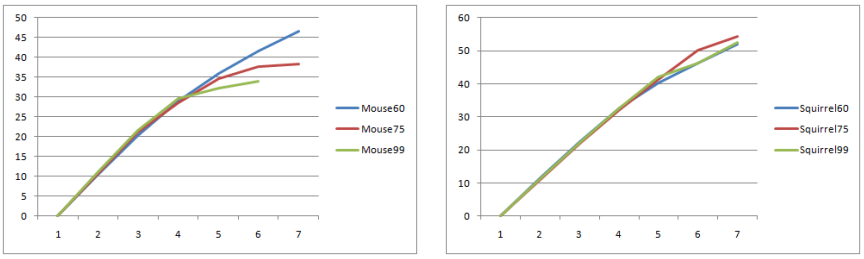


Figure 4.5: Characteristic curves from mesh dilation-erosion of the Mouse and Squirrel models.

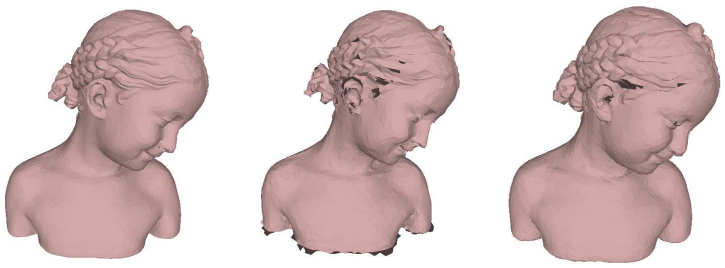


Figure 4.6: Mesh dilation and erosion applied to the Bimba model. From left: original plus eroded and dilated models after 4 iterations.

Results of our first experiment

The 32 models listed in Table 4.2 were used, each with four mesh resolutions as specified in Table 4.1. As explained before, the first three mesh resolutions were used for constructing the FVs of the model, and the last one was used for testing. Each model was characterized by 7 parameters, 5 from the method based on *mesh smoothing* (surface of original model after normalization to unit sphere; contraction ratio after 10 iterations; 3 coefficients of the quadratic approximation of the smoothing curves), and 2 based on *dilation and erosion* (volume of original model after normalization to unit sphere; volume between dilated-eroded surfaces after 2 iterations). The FVs of the objects with the test resolutions were compared with all FVs of the database, and the objects were sorted by using the Euclidean distance between the FVs. Table 4.2 lists the results, starting with the object with the smallest distance, then the object with the next smallest distance, and so forth, until the fifth object where possible. In some cases the contraction ratio was so big that the application of the total number of iterations was impossible and no reliable parameters could be obtained. This requires further research. In any case, Table 4.2 shows that in 26 of 32 cases the correct

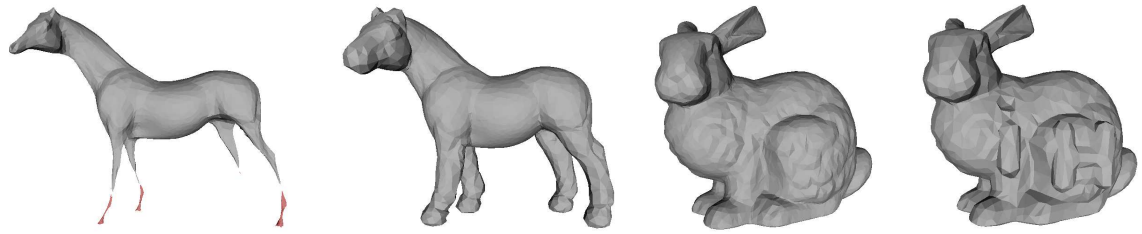


Figure 4.7: Left to right: Horse (res. 7.5) after erosion and dilation, Bunny and Bunny iH.

object was ranked first. The average ranking rate $\bar{R} = (1/32)\sum_{i=1}^{32} R_i$, where R_i is the ranked position (first, second, third,...) of object i, is 1.84. This value reduces to 1.29 if we take out the screwdriver model which was ranked at position 19. This means that the majority of objects is ranked at position 1 or 2, at least within the first 3 or 5 positions. This is extremely useful for narrowing a full object comparison in a big and complex database. It should be stressed that, although 26 objects were ranked first, this does not mean that the correct object has been identified in all cases. The most similar object may have been detected, but in real conditions, i.e., with big and complex databases like protein structures, the search has been narrowed in order to save time for detailed object comparisons.

4.3.2 Improved features based on mathematical morphology

Second experiment:

Using mathematical morphology in an *ad hoc* way may produce inaccurate characteristic curves. In order to get a higher degree of certainty, we decided to use the formal concepts of mathematical morphology. Before describing this method we suggest reviewing the fundamentals of mathematical morphology in Appendix A.

Applying a sphere as structuring element to all vertices leads to a smaller object in case of erosion and a bigger one in case of dilation. In the case of the Horse model, repeated erosions will cause discontinuity of the legs. The processed Horse models as shown in Fig. 4.7 (left) were obtained using by $r = 0.0170$ in erosion and $r = 0.0259$ in dilation at model resolution 7.5. The small stumps, created by erosion and shown in red, were deleted and their volumes were excluded from the computation of the Horse's volume. The same procedure was applied to the other models.

For each model resolution, the difference volume defined as dilated volume minus eroded volume according to Eq. (A.4) yields an approximately linear function of the radius of the

Table 4.2: Results of our first experiment.

N	Testing Model	Ordered output
1	Blade	**1**-6-30-22-12
2	Bimba	**2**-26-32-22-15
3	Block	**3**-27-9-15-32
4	Bunny	**4**-7-23-14-31
5	Cow	12-**5**-6-17-11
6	Cow2	**6**-12-30-22-1
7	DancingChildren	**7**-4-23-14-10
8	Dragon	**8**-32-23-9-2
9	Duck	**9**-32-15-8-2
10	Eros	18-25-**10**-14-26
11	Fish	**11**-6-12-1-30
12	FishA	**12**-6-1-5-11
13	GreekSculpture	**13**-28-20-17-5
14	IsidoreHorse	**14**-25-18-10-26
15	Mouse	**15**-32-9-22-2
16	Pulley	**16**-24-19-3-9
17	Torso	**17**-5-12-20-6
18	CamelA	25-**18**-14-10-26
19	Carter	**19**-24-16-8-9
20	Chair	**20**-17-5-12-1
21	Dancer	11-12-5-6-**21**
22	Dente	**22**-26-30-2-6
23	Elk	**23**-7-4-14-8
24	Grayloc	**24**-19-16-8-9
25	Horse	**25**-14-18-10-26
26	Kitten	**26**-2-22-10-30
27	Lion-dog	**27**-3-9-15-17
28	Neptune	**28**-21-5-20-17
29	Ramesses	**29**-10-14-4-7
30	Rocker	22-**30**-26-2-6
31	Screwdriver	*... -19 th Place*
32	Squirrel	**32**-15-9-2-22

structuring element, see Fig. 4.8. After least-squares line fitting by $Ar + B$, the slope coefficient A reflects the complexity of the surface of the object. Coefficient B also reflects the complexity, but with emphasis on the capacity of the object to be eroded and dilated without self-intersections, i.e., the first step (i) of the two-step process as described in Appendix A. The entire procedure can be summarized as follows. First, in feature extraction, a model is eroded and dilated until the first self-intersection occurs, in the eroded model and the dilated one separately. From these two modified models we compute the difference volume V_i at each mesh resolution i as listed in Table 4.1, from which the average volume V is

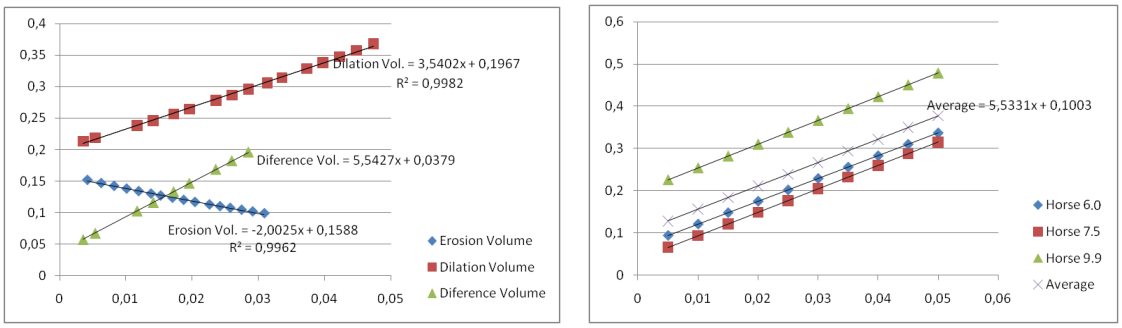


Figure 4.8: Difference volume of Horse model as a function of sphere radius r. Mesh resolution 7.5 (left) and average difference function of resolutions 6.0, 7.5 and 9.9 (right).

computed. Then, for each mesh resolution, both models are further eroded and dilated and the difference volumes as a function of total sphere radius r are computed. From these functions the line parameters A and B are determined by linear regression. Again, these parameters are averaged over the first three mesh resolutions as given in Table 4.1.

The same parameters are computed in the case of the fourth mesh resolution given in Table 4.1. Finally, the parameters of the fourth mesh resolution of each model are compared with the averaged parameters of the first three mesh resolutions of all models, and the models are ranked using the Euclidean distance of the vectors (A, B, V).

Results of the second experiment

Table 4.3 lists the first six ranking positions with increasing Euclidean distance. The six ranking positions are due to object number 13 (Dragon), which gave the worst result (see below). An additional test involved a modified object named "Bunny iH", see Fig. 4.7 (right), which was not part of the dataset. Bunny iH was correctly ranked first as Bunny, object number 4, although the original Bunny (object 4 in Table 4.3) was ranked second after object Blade. Since both Bunny (4) and Blade (1) have the ranking 1-4 in Table 4.3, this probably only means that the feature vectors - with only three parameters - are very close.

If R_i is the correct ranking position of object i, which is shown boldfaced in Table 4.3, then the average ranking rate is $\bar{R} = \frac{1}{32}\sum_{i=1}^{32} R_i$. Our results are summarized by $\bar{R} = 1.47$, with a root-mean-square (RMS) error of 0.973. Although the distribution is asymmetric, this means that most objects were ranked first or second. Indeed, in Table 4.3 we can see that 23 objects of all 32 were ranked first, 6 objects second and only 2 third. Object number 13 (Dragon) was an exception: it was ranked sixth. Fig. 4.9 shows the ranked models

Figure 4.9: Ranked response at Dragon query. From left (first) to right: Eros, Elk, Mouse, DancingChildren , Squirrel and Dragon.

Table 4.3: Results of the second experiment.

N	Test model	Ranking	N	Test model	Ranking
1	Blade	**1**-4-12-22-28-13	17	Fish	27-**17**-2-18-9-31
2	Bimba	**2**-27-18-17-9-21	18	Grayloc	26-6-**18**-3-14-23
3	Block	**3**-14-23-32-30-11	19	GreekSculpture	**19**-8-20-7-31-9
4	Bunny	1-**4**-22-12-28-13	20	Horse	**20**-7-19-25-5-29
5	CamelA	**5**-25-29-7-20-19	21	IsidoreHorse	**21**-2-27-18-17-9
6	Carter	26-**6**-18-3-14-23	22	Kitten	**22**-4-1-12-28-13
7	Chair	**7**-20-25-5-29-19	23	Liondog	14-3-**23**-32-30-16
8	Cow	31-**8**-9-17-18-27	24	Mouse	16-**24**-15-13-11-30
9	Cow2	**9**-17-31-18-27-8	25	Neptune	**25**-5-29-7-20-19
10	Dancer	**10**-29-5-25-7-20	26	Pulley	**26**-6-18-3-14-23
11	DancingChildren	**11**-15-16-30-24-13	27	Ramesses	**27**-18-17-2-9-31
12	Dente	**12**-1-4-22-13-24	28	Rocker	**28**-21-22-4-1-12
13	Dragon	16-15-24-11-30-**13**	29	Screwdriver	5-**29**-25-7-20-19
14	Duck	**14**-23-3-32-30-16	30	Squirrel	**30**-16-11-15-24-13
15	Elk	**15**-11-16-24-13-30	31	Torso	**31**-9-8-17-18-27
16	Eros	**16**-15-24-11-30-13	32	Vaselion	**32**-23-30-14-3-16
	Bunny iH	**4**-22-1-12-28-13			

in the Dragon query. The dataset tested is too small to compute advanced performance measures as used in the SHREC contest, but our correct recognition rate of $23/32 = 0.72$ is at the top of the range between 0.45 and 0.70 achieved in the 2010 contest [Veltkamp et al., 2010]. The results of our first experiment (Section 4.3.1, Table 4.2) was $26/32 = 0.81$. This was achieved by combining features based on mesh smoothing and an *ad hoc* form of mathematical morphology, whereas in this section only mathematical morphology was used.

4.3.3 Extended multi-resolution signatures

Third experiment:

Using only mathematical morphology did not substantially increase the accuracy; ranking rates in the previous experiments were 1.84 and 1.47. The main difference of this third experiment is the replacement of the two volumetric features as used in Section 4.3.1 by three, obtained by formally and structured use of mathematical morphology, see Section 4.3.2. Each model is characterized by 8 parameters, 5 from the method described in Section 4.3.1 (surface A of original model after normalization to unit sphere; contraction ratio C after 10 iterations; 3 coefficients, a_0, a_1 and a_2 of the quadratic approximation of the smoothing curves); and 3 from Section 4.3.2 (volume V of original model after normalization to unit sphere; linear regression coefficients b_0 and b_1 of the approximated difference volume between dilated-eroded surfaces after 10 iterations). The ten iterations used in both methods were defined in order to keep the representative functions of the models well fitting to the models.

Results of the third experiment.

Table 4.4: Ranked objects using all eight parameters. Only three objects (6, 9 and 36) were ranked second.

N	Model	Ordered output	N	Model	Ordered output
1	Amphora	**1**-31-16-29-2	21	Fish	**21**-10-22-3-34
2	Bimba	**2**-13-30-27-29	22	FishA	**22**-10-39-21-3
3	Blade	**3**-22-26-21-10	23	Grayloc	**23**-7-36-33-4
4	Block	**4**-18-17-28-36	24	GreekSculpture	**24**-25-8-10-9
5	Bunny	**5**-27-13-30-1	25	Horse	**25**-6-24-8-9
6	CamelA	25-**6**-24-8-15	26	IsidoreHorse	**26**-3-22-21-10
7	Carter	**7**-23-36-33-4	27	Kitten	**27**-5-30-13-2
8	Chair	**8**-25-6-24-9	28	Liondog	**28**-18-17-4-40
9	Cow2	39-**9**-22-10-3	29	Maneki	**29**-13-2-27-5
10	Cow	**10**-21-9-39-22	30	Moai	**30**-27-2-13-5
11	Dancer	**11**-14-32-15-37	31	Mouse	**31**-38-19-16-1
12	DancingChildren	**12**-19-20-29-31	32	Neptune	**32**-37-15-14-6
13	Dente	**13**-27-5-30-2	33	Pulley	**33**-23-7-36-4
14	Dilo	**14**-15-37-11-32	34	Ramesses	**34**-21-10-22-24
15	Dino	**15**-37-6-32-25	35	Rocker	**35**-30-27-26-5
16	Dragon	**16**-38-31-19-1	36	RStage	7-**36**-23-33-4
17	Duck	**17**-28-18-40-4	37	Screwdriver	**37**-15-32-6-25
18	Egea	**18**-17-28-4-40	38	Squirrel	**38**-19-31-40-16
19	Elk	**19**-12-38-31-40	39	Torso	**39**-9-10-22-21
20	Eros	**20**-12-29-5-15	40	Vaselion	**40**-38-19-12-31

Table 4.4 lists the results, starting again with the object with the smallest distance, then the object with the next smallest distance, and so forth, until the fifth object. The average ranking rate $\bar{R} = (1/40)\sum_{i=1}^{40} R_i$, where R_i is the ranked position of object i, is 1.075. This means that the majority of objects is ranked at position 1 or 2, at least at the first positions. Indeed, Table 4.4 shows that 37 objects of all 40 were ranked first and only 3 second, i.e., when all eight parameters are used. The recognition rate is now $37/40 = 0.925$, which is much better then the previous rates of $23/32 = 0.72$ and $26/32 = 0.81$ in the first two experiments. The best rate of 0.70 in the 2010 SHREC Contest [Veltkamp et al., 2010] is much smaller, but this contest is much more difficult because it is based on a total of 10,000 objects.

Concerning the objects ranked second, CamelA (6) was ranked after Horse (25), and RStage (36) was ranked after Carter (7). These are rather similar objects, i.e., animals and mechanic pieces, but Horse and Carter were correctly ranked first. On the other hand, Cow2 (9) was ranked after Torso (39), but these are quite different objects, and Torso was correctly ranked first.

We performed a few additional tests in order to study the significance of individual parameters and possible parameter combinations. Table 4.5 shows the average ranking rates of all 40 objects when each parameter is used individually. Please note that we call the parameters a_i and b_i for distinguishing smoothing (a) from mathematical morphology (b). In the early Sections (4.3.1 and 4.3.2) we called V_0 and V_2, A and B, respectively. The best parameters are V (ranking rate of 1.75), b_1 (1.8), A (2.0), a_1 (2.5) and b_0 (3.0). The discriminative power of the other three parameters is much poorer. We then did a sequential test. We took the best individual parameter V, and combined it with each of the other seven parameters. Using the best average ranking result, the best couple of parameters was selected and then combined with each of the remaining six parameters, and so on. This is not a full parameter search with all possible combinations, but it gives an impression of the most discriminative parameters. Table 4.6 lists the first five results. Using more than three parameters does not improve performance, i.e., there are always three objects ranked second. On the basis of Table 4.5 one might expect that the couple $[V, b_1]$ would be best, but Table 4.6 shows that the couple $[V, A]$ performs better. However, the triplet $[V, A, b_1]$ includes the best three from Table 4.5. Similarly, the best quadruplet $[V, A, b_1, a_1]$ includes the best four and the quintuple $[V, A, b_1, a_1, b_0]$ the best five. The remaining parameters did

not improve performance, but the set of only 40 objects may be too small to draw final conclusions, apart from the fact that the best result obtained with all eight parameters is equal to that obtained with only three parameters.

Finally, in order to further validate this approach we also tested two deformed objects;

Table 4.5: Average ranking rates using individual parameters.

Smoothing					Morphology		
A	C	a_0	a_1	a_2	V	b_0	b_1
2.0	11.7	6.4	2.5	8.9	1.75	3.0	1.8

Table 4.6: Average ranking rates obtained by a sequential combination of parameters; see text.

Parameters	Ranking rate
[V]	1.75
[V,A]	1.2
[V,A,b_1]	1.075
[V,A,b_1,a_1]	1.075
[V,A,b_1,a_1,b_0]	1.075

see Fig. 4.10. Object Bimba was deformed by applying the algorithm fBM (fractal Brownian Motion, from the Meshlab package [Cignoni et al., 2008]) to all its vertices. Object Bunny-iH exhibits the characters i and H on its left flank; Bunny-iH is part of the AIM@SHAPE database. Both objects were correctly matched (ranked first) with the original objects.

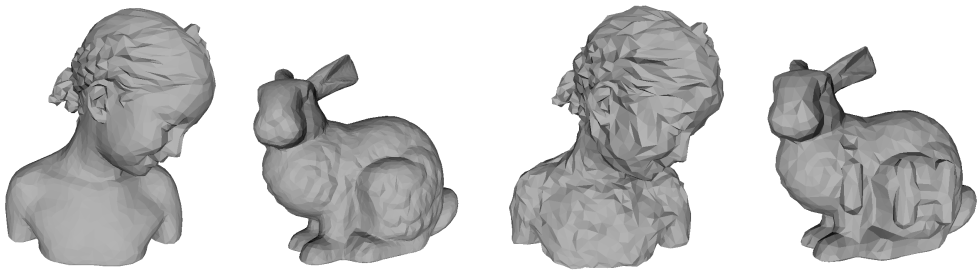


Figure 4.10: Original models (left) and deformed ones (right), Bimba and Bunny-iH.

4.4 Discussion

Taking into account the complexity of the mesh models, the results obtained are quite good. The average ranking rate of 1.47, obtained in Section 4.3.2, was achieved by using only three parameters: the difference volume V of the eroded and dilated objects until the first

self-intersections occurred, plus the linear regression parameters A and B of the difference volumes after additional erosions and dilations. This means that these parameters, which reflect systematic object deformations with decreasing detail as a function of increasing structuring element, are capable of characterizing different shapes. However, this does not mean that similar or other signatures based on mathematical morphology cannot perform even better; it makes sense to experiment with other signatures in the future. In addition, two of our parameters provide multi-scale signatures because of the increasing structuring element, and it makes sense to combine these parameters with other multi-scale signatures, for example those based on mesh smoothing (Section 4.3.1). The latter method achieved on a similar set of objects an average ranking rate of 1.84, but seven instead of three parameters were used. Therefore, the combination of perhaps two multi-scale parameter sets with perhaps as much as six parameters might boost performance.

The extension of Section 4.3.1 by replacing the *ad hoc* volumetric differences by features computed with mathematical morphology is very accurate. Tested signatures—at least three of them—appear to be robust due to their global nature. In addition, small and local deformations of the object's mesh (Fig. 4.10) do not introduce significant modifications of the characteristic signatures, although more types of deformations must be tested with more than two objects. In general, the dataset of 40 objects tested here is too small to compute advanced performance measures as used in the SHREC contest. However, our correct recognition rate of $37/40 = 0.925$ is much better than the range between 0.45 and 0.70 as achieved in the SHREC contest of 2010 [Veltkamp et al., 2010]. Therefore, in future work the number of objects in our database should be increased such that the significance of individual parameters and the best combinations of these can be validated. In parallel, the method should be tested by using other types of objects, such as 3D meshes of complex proteins. A practical problem is that some objects are not available with different mesh resolutions, while others are not 2-manifold or "watertight" and these must be pre-processed. Another problem is that the elimination of disconnected parts after erosions (Fig. 4.7), which has been done manually here using Meshlab, must be automated. The latter problem does not only occur in case of e.g. animals with legs, but can be expected in case of protein structures.

Of special interest is object categorization. One class consists of 4-legged animals, see objects 6, 9, 10, 14, 15 and 25 in Fig. 4.1. In Table 4.4 we can see that Dilo (14) has Dino (15)

at rank 2, but the Dino (15) ranking does not include Dilo (14). Only Dino includes CamelA (rank 3) and Horse (rank 5). Cow (10) includes Cow2 (9) at third rank, and Cow2 includes Cow at rank 4. CamelA (6) has Horse (25) at rank 1, and Horse has CamelA at rank 2. These animals must be distinguished from Chair (8), which includes in its ranking list objects Horse (25) at place 2, CamelA at place 3 and Cow2 at place 5. In contrast, only Horse (25) and CamelA (6) were "linked to" Chair (8), both at place 4. Hence, object categorization appears to be much more complicated than object recognition, and further work must be done to increase categorization performance. From a human perspective, object detection, segregation, categorization and recognition are linked processes which cannot be completely sequential; they must be done in parallel, at least partially, and therefore they are overlapping significantly [Rodrigues, 2008]. Keypoints, extracted from local features, can improve 3D similarity retrieval [Bustos et al., 2012]. Maybe this can be done as in the human visual cortex, in a dynamic and parallel manner, adding to global features an increasing number of local features, just enough to obtain a fair categorization.

Chapter 5

Saliency maps and Keypoints

Abstract: In this chapter we expose the 3D model of a saliency map, which incorporates concepts of low-level human visual attention. The human visual system combines different visual features (color, orientation, movement, etc.) into one single topographic map, the saliency map, for local scene and object conspicuity. Curvatures of the surfaces are good indicators for the shape of 3D objects. The Shape Index and Curvedness measure the type and how curved a surface is. We will use Shape Index to create a more descriptive saliency map and a multi-scale scheme for detecting keypoints.

5.1 Introduction

The concept of a saliency map was introduced in the middle of the 1940's when neuroscience research addressed visual information overload. Our brains take decisions about which information must be chosen for more detailed processing, while other information could be discarded. The selection and importance-ordering process is known as *attention*. Saliency maps were proposed by Koch and Ullman [1985] as a mechanism to model selective attention. They stated that the most salient location, in a bottom-up or data-driven approach applied to a visual scene, is a best choice for directing attention. The proposal of Koch and Ullman was merely conceptual; only in 1996 an implementation of a saliency map based on color, intensity, orientation and motion was published [Niebur and Koch, 1995; Itti et al., 1998].

As for many topics in image analysis, it can be expected that the concept of saliency map can be extended to 3D shape analysis. We stated in Chapter 3 that 3D shape is defined by the way of how intrinsic features can represent the shape. In a more consistent representation,

characteristics must be invariant to rigid transformations and isometric (isotropic) scaling. If the used representation allows for the use of multi-scale methods, then saliency maps can be extended to 3D shape analysis [Lee et al., 2005a]. Yang and Shen [2012] proposed a multi-scale, salient, geometric feature extraction method. It works on a polygonal mesh and uses two circumscribed multi-scale characteristics, Gaussian curvature and local shape descriptors. Salient geometric features are descriptors that highlight non-trivial areas of a surface, saliency being directly related to curvature and its variance [Gal and Cohen-Or, 2006]. Principal curvatures are estimated by PCA on local neighborhoods. For defining the neighborhood, they used a spherical kernel of radius r, that is used as a natural scale. Local surface descriptors are composed by the surface point p and its approximated quadratic surface patch in the vicinity of p. This type of descriptors have the advantage of being independent of mesh triangulation and provide an easy way to cluster non-trivial salient features.

A multi-scale scheme for describing these features, in conformance with human perception, can be used for practical applications as a proper viewpoint selection in 3D object databases. With the growing use of 3D objects, it becomes important to select viewpoints that describe the most salient features. For example, mesh simplification can be driven by saliency: a reduction of the number of triangles can be applied to areas where saliency is low and by preserving them in places which exhibit high values.

One of the main problems in computing curvatures of polygonal meshes is related to mesh resolution. In order to avoid the need of smoothing a surface for an accurate curvature estimation, a 3D version of the SUSAN operator was proposed. The USAN (Univalue Segment Assimilating Nucleus) in 2D pixel areas was redefined as a 3D volumetric sphere, in voxels, by Walter et al. [2008]. They showed the relation between saliency and curvature in 2D. The 3D version of the SUSAN operator defines mean saliency as $\overline{S} = V_{\text{USAN}}/V_{\text{Voxelized Sphere}}$. V_{USAN} is the volume under the surface and inside the sphere, and $V_{\text{Voxelized Sphere}}$ is the sphere volume. Despite the ability of determining three types of saliency (cap, planar and cup), as far we know this operator has not yet been applied in a multi-scale framework.

5.2 Laplacian and Difference of Gaussians

In image processing it is very common to use the Laplacian as edge detector. It measures the second spatial derivative and it is isotropic by nature. Due to sampling noise or outliers

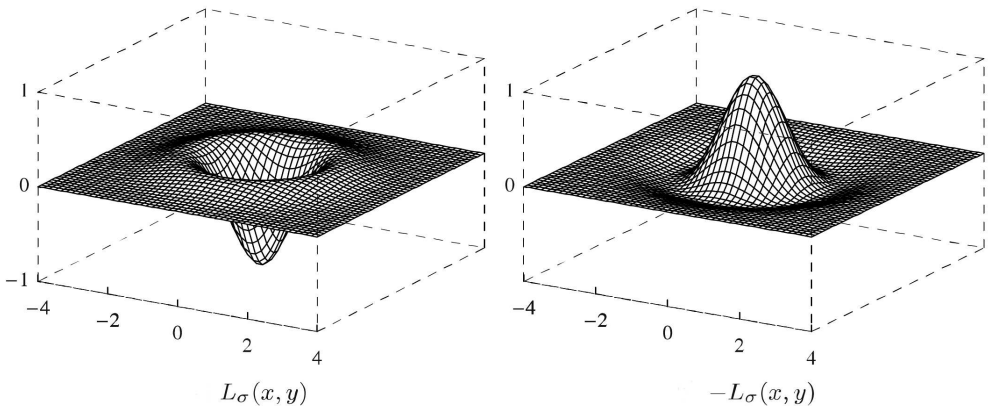


Figure 5.1: At left, Laplacian of Gaussian, $L_\sigma(x,y)$, of a Gaussian with $\sigma = 1$. The right image shows the inverted LoG, $-L_\sigma(x,y)$.

the Laplacian is often preceded by a smoothing filter, usually a Gaussian. A Gaussian filter is defined by the kernel

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \cdot \exp(-\frac{x^2+y^2}{2\sigma^2}), \tag{5.1}$$

with size (standard deviation) σ.

The Laplacian of a continous, 2D function $f(x,y)$ is defined by the sum of the second partial derivatives in x and y directions. We note that the Laplacian is a scalar operator,

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{5.2}$$

By applying this definition to the 2D Gaussian and using equal widths ($\sigma = \sigma_x = \sigma_y$), we obtain the Laplacian-of-Gaussian (LoG) operator,

$$\begin{aligned} L_\sigma(x,y) = \nabla^2 G_\sigma(x,y) &= \frac{\partial^2 G_\sigma(x,y)}{\partial x^2} + \frac{\partial^2 G_\sigma(x,y)}{\partial y^2} \\ &= \frac{1}{\pi\sigma^4} \cdot \frac{x^2+y^2-2\sigma^2}{2\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}. \end{aligned} \tag{5.3}$$

LoGs are commonly used to model the initial processes in biological vision systems, especially to describe the center-surround receptive fields of retinal ganglion cells [Marr and Hildreth, 1980]. In this model an on-center cell is stimulated if the center of the receptive field is exposed to light. When the light hits the surround (off-center) the cell is inhibited. Since L_σ resembles off-center cells, therefore the negative $-L_\sigma$ is taken for on-center cells; see Fig. 5.1.

Despite that the implementation of LoG can be almost separable [Huertas and Medioni, 1986], due to the computational costs when the scale (σ) increases it is very usual to im-

plement LoG as an approximation of a Difference-of-Gaussians (DoG). The problem of implementing a LoG by using a DoG is to establish the correct interval scale between the Gaussians. Considering two Gaussians, the one with parameter σ and the other with $\kappa\sigma$, the DoG is expressed as

$$D_{\sigma,\kappa}(x,y) = G_{\kappa\sigma}(x,y) - G_{\sigma}(x,y), \tag{5.4}$$

with $\kappa \geq 1$ a factor specifying the relative width of the two Gaussians. Using a proper scale factor, the DoG expressed as $D_{\sigma,\kappa}(x,y)$ resembles the LoG expressed as $L_{\sigma}(x,y)$:

$$L_{\sigma}(x,y) \approx \lambda \cdot D_{\sigma,\kappa}(x,y), \quad \lambda = \frac{2\kappa^2}{\sigma^2 \cdot (\kappa^2 - 1)}. \tag{5.5}$$

It has been shown that a κ factor in $]1.1, \ldots, 1.3[$ produces quite precise results in 2D [Burger and Burge, 2013]. In our work, extending this framework to 3D, we will use $\kappa \approx 1.2599$. This value comes from dividing the scale interval of a factor 2 (octave, see Eqn. (5.8)) into 3 equal sub-intervals, $\kappa = 2^{1/3}$.

As stated before, in image processing, for example in SIFT [Lowe, 1999], DoG filters are used instead of normalized LoG filters in the multi-scale domain. This leads to the *continuous or discrete* Gaussian scale space representations. The following axioms are well known in image processing and we will review them here so that later, in Section 5.2.1, we can extend them to 3D. Consider a continuous and real 2D function $F(x,y)$ with $x, y \in R$. Its representation in Gaussian scales is given by the linear convolutions

$$\mathcal{G}(x,y,\sigma) = F(x,y) * G_{\sigma}(x,y). \tag{5.6}$$

The scale space $\mathcal{G}(x,y,\sigma)$ has also a continuous domain. The discrete case is a bit different. With the exception of scale $\sigma = 0$, the *discrete* image $I(u,v)$ can be thought of as already been smoothed with a kernel of size $\sigma_s \geq 0.5$ (in pixels). As a consequence, the Gaussian scale space at scales smaller than σ_s is not defined, it exists only at scales with $\sigma > \sigma_s$. The representation of the Gaussian scales of the discrete image $I(u,v)$ is a set of M images, each generated by applying the (discretized) Gaussian kernel G_{σ}. Each image G_m, $0 \leq m \leq M-1$, is a smoothed version of the original discrete image,

$$G = (G_0, G_1, \ldots, G_{M-1}), \tag{5.7}$$

and G_{m+1} is coarser than G_m. Usually the ratio of the kernel sizes between adjacent scale levels is kept constant

$$\Delta_{\sigma} = \frac{\sigma_{m+1}}{\sigma_m}, \quad m = 0, \ldots, M-1. \tag{5.8}$$

Δ_σ is defined such that σ_m doubles with a number of scale levels Q. The interval that comprises the scale and its double is denoted by *octave*. The scale increment can be expressed as $\Delta_\sigma = 2^{1/Q}$, usually with $Q \in [3, 4, 5, 6]$. Using a base scale $\sigma_0 > \sigma_s$, σ_s being the scale that corresponds to the original discrete image, and a scale interval (Δ_σ) allows to build a discrete, Gaussian scale space. For a given Q and base scale σ_0 the scale at level G_m can be computed by

$$\sigma_m = \sigma_0 \cdot \Delta_\sigma^m = \sigma_0 \cdot 2^{m/Q}, \tag{5.9}$$

with m as in Eqn. (5.8) and $\sigma_0 = 1.6$ as suggested by Lowe [2004]. The scale levels $G_0, G_1, \ldots, G_{M-1}$ obtained by Gaussian filtering with $\sigma_0, \sigma_1, \ldots, \sigma_{M-1}$, see Fig. 5.2, can be seen as a sliced representation of continuous scale space.

One of the major problems in applying 2D Gaussian filters is the computational cost, which becomes excessive at larger scales. To overcome this problem it is possible to reduce the size of the smoothed images, by sub-sampling them. The feasibility of sub-sampling is assured by the fact that a Gaussian filter operation reduces the bandwidth in proportion to the size of the filter. If the image size is kept constant at all scales, the smoothed (filtered) images become over-sampled as the scales become coarser. It is therefore possible to reduce the sampling rate as the scale increases, without loss of significant signal information. Concretely, doubling the scale reduces the bandwidth by a factor of two. Hence, at each octave the sampling rate can be halved without loss of information. The construction of a DoG scale space (as a substitute for LoG) can be done by sub-sampling at each new octave; the decimation to half of the resolution is done by selecting one and rejecting the next pixel (as in a chessboard pattern). The image size reduction allows to construct a hierarchical Gaussian scale space with P octaves, representing almost the same domain as Gaussian scale space which was defined in Eqn. (5.7),

$$G_p = (G_{p,0}, G_{p,1}, \ldots, G_{p,Q}), \quad p \in [0, P-1]. \tag{5.10}$$

In each octave p there are $Q+1$ sub-scale levels $q \in [0, Q]$. The correspondence between a level in hierarchical Gaussian space and the space indexed by m is defined by

$$m = Qp + q. \tag{5.11}$$

Taking as reference the octave base level $G_{p,0}$, the size of each sub-scale (inside the octave) can be computed as

$$\tilde{\sigma}_q = \sigma_0 \cdot \sqrt{2^{2q/Q} - 1}, \; q \in [0, Q]. \tag{5.12}$$

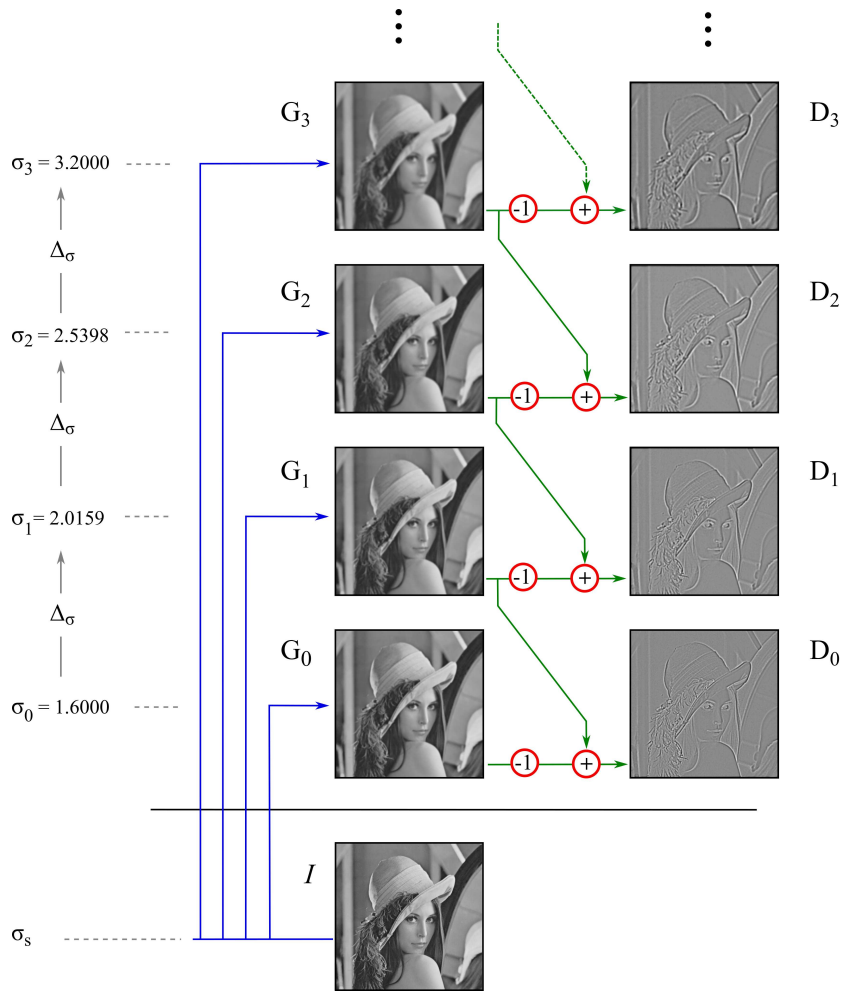


Figure 5.2: Gaussian scale space. The first four blurred images are obtained by Gaussian filtering of the discrete *Lena* image (bottom) with widths $\sigma_0, \sigma_1, \sigma_2, \sigma_3$ (absolute scales or discrete version $\hat{\sigma}_m$, see Burger and Burge [2013]. The blue arrows show the discrete filtering and the green arrows show the construction of DoG images. Figure adapted from Burger and Burge [2013].

Considering an image I with size $M \times N$, taking into consideration that doubling the scale reduces the bandwidth by half, decimation between successive octaves can be achieved. At octave G_p the size of the image is

$$M_p \times N_p = \frac{M}{2^p} \times \frac{N}{2^p}. \tag{5.13}$$

The hierarchical Gaussian scale space is one of the *pyramidal* methods used in image processing [Burt and Adelson, 1987], see Fig. 5.3.

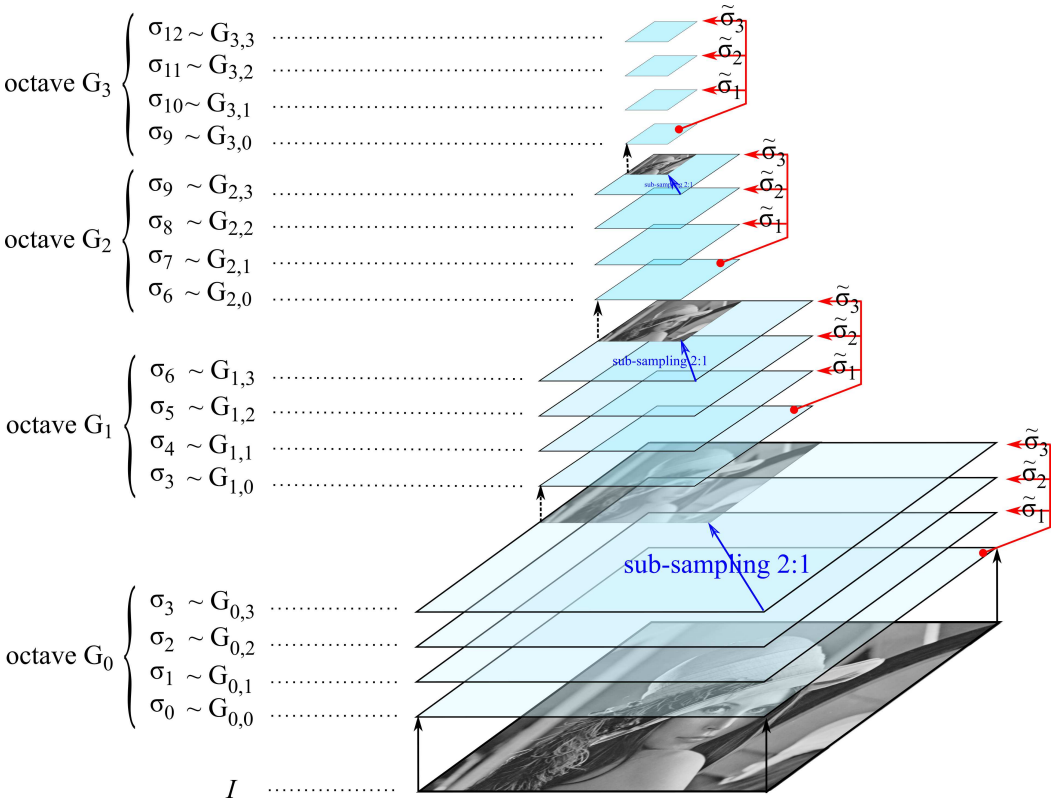


Figure 5.3: Gaussian scale space by octaves and sub-sampling. Each octave is split up into 3 scale intervals, $Q = 3$. The base scale of each octave, with $p > 0$, is acquired by sub-sampling 2:1 the previous scale ($G_{p-1,3}$). The correspondence between the absolute scale σ_m and relative scale, inside octave $G_{p,q}$, is established by $m = Qp + q$. Inside each octave a constant set of Gaussian filters $(\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3)$ is applied to the octave base scale, denoted by $G_{p,0}$. Figure adapted from Burger and Burge [2013].

5.2.1 Extension to 3D

To the best of our knowledge, despite all theoretical foundation, Gaussian scale space of 3D meshes has not yet been applied. One of the problems is due to the representations of 3D models. If a model is represented by a mesh, it must be a regular 3D grid in voxels. Even using the voxel representation, the voxel resolution (size) problem remains to be solved. Intensive computations would be necessary to interpolate, in order to compute the influence of a Gaussian filter to the surface. Here we extend the Gaussian space scale to 3D objects represented by triangle meshes.

First let us extend the mathematical formulation of the Gaussian filter to 3D space. Consider the equation

$$G_\sigma(x, y, z) = \frac{1}{2\sqrt{2}\pi^{3/2}\sigma^3} \cdot e^{-\frac{x^2+y^2+z^2}{2\sigma^2}} \tag{5.14}$$

as a trivial extension from Eqn. 5.1. Its Laplacian (LoG) is:

$$\begin{aligned} L_\sigma(x,y,z) = \nabla^2 G_\sigma(x,y,z) &= \frac{\partial^2 G_\sigma(x,y,z)}{\partial x^2} + \frac{\partial^2 G_\sigma(x,y,z)}{\partial y^2} + \frac{\partial^2 G_\sigma(x,y,z)}{\partial z^2} \\ &= \frac{x^2+y^2+z^2-3\sigma^2}{2\sqrt{2}\pi^{3/2}\sigma^7} \cdot e^{-\frac{x^2+y^2+z^2}{2\sigma^2}}. \end{aligned} \tag{5.15}$$

A Gaussian filter can be applied in many ways. Smoothing can be applied to some specific surface feature. In image processing (grey-scale images) the smoothing is applied to pixel values. Our work will extend the Gaussian filter to 3D by smoothing intrinsic surface features. Empirical psychological and neurophysiological studies of object recognition showed that visual information used during shape recognition is concentrated in regions of higher curvatures, and that negative curvatures (concave surface) are perceptually more relevant than positive ones (convex surface) [Lim and Leek, 2012]. Shape Index and Curvedness can provide a mean to *measure* convex or concave surfaces; see Section 3.2. We choose SI because of its importance to human perception, but for the purpose of extending the concept of Gaussian scale space, which includes octaves and hierarchical sub-sampling, any geometric features could be used.

In order to build the 3D Gaussian scale space we use as base scale the value of $2 \times 0.3\%$ of the bounding box diagonal. This value was chosen after experimenting with different values and objects. In case of the Armadillo model $\sigma_0 = 1.3728$, $Q = 3$ and $\Delta_\sigma = 2^{1/3} \approx 1.2599$. Table 5.1 shows the scales (widths) of the first five octaves, with their intra-scales. Please recall that all scales are Δ_σ apart. Figure 5.4 shows the Armadillo model rendered with Gouraud shading by Meshlab. This model will be used for illustrating the processing scheme.

Figure 5.4: Armadillo mesh model with 40,267 vertices and 80,530 triangles. The bounding box diagonal is 228.7947, hence the basis scale is $2 \times 0.003 \times 228.7947 = 1.3728$

Table 5.1: Size of Gaussian filters at scales m, with $\sigma_0 = 1.3728$, $Q = 3$ and $\Delta_\sigma \approx 1.2599$.
The gray rows show the octave steps $(\sigma_0, \sigma_3, \sigma_6, \sigma_9, \sigma_{12}, \sigma_{12})$.

m	σ_m
0	1.3728
1	1.7296
2	2.1792
3	2.7496
4	3.4592
5	4.3584
6	5.4912
7	6.9185
8	8.7167
9	10.9824
10	13.8370
11	17.4335
12	21.9648
13	27.6739
14	34.8669
15	43.9296

The 16 scales shown in Table 5.1 provide an example selection with two scales between the octave scales (the ones marked in gray). More scales can be selected, but this does not have a clear advantage. If the scale becomes very big, the SI values will converge towards one global, mean value; see Fig. 5.6. Figure 5.5 shows the model surface colored with SI values, before and after Gaussian filtering with different filter sizes; warm colors reflect positive values and cool colors negative ones. Below we only use the first 2 octaves in order to demonstrate the construction of hierarchical Gaussian scale space with mesh decimation. To verify the consistency of hierarchical scale space we applied Gaussian filters with $\tilde{\sigma}_1 = 1.0521, \tilde{\sigma}_2 = 1.6924$ and $\tilde{\sigma}_3 = 2.3778$, based on Eqn.(5.12) on the first 2 octaves. These octaves can also be referred to by $\sigma_0 \ldots \sigma_3$ and $\sigma_3 \ldots \sigma_6$, in the scale domain defined by m, see Eqns. (5.7) to (5.9). The size reduction was applied to the second octave, $\sigma_3 \ldots \sigma_6$, by resizing the model to half of its size and by sub-sampling. Volumetric reduction was done by applying a factor of 0.5 in the scale transform, and sub-sampling was done by iterative vertex-pair contraction. The position of the new vertex is one of the pair or the average coordinates, the new position must minimize the local quadric error. The number of vertices was also reduced to half [Garland and Heckbert, 1997]. In reducing the model, we

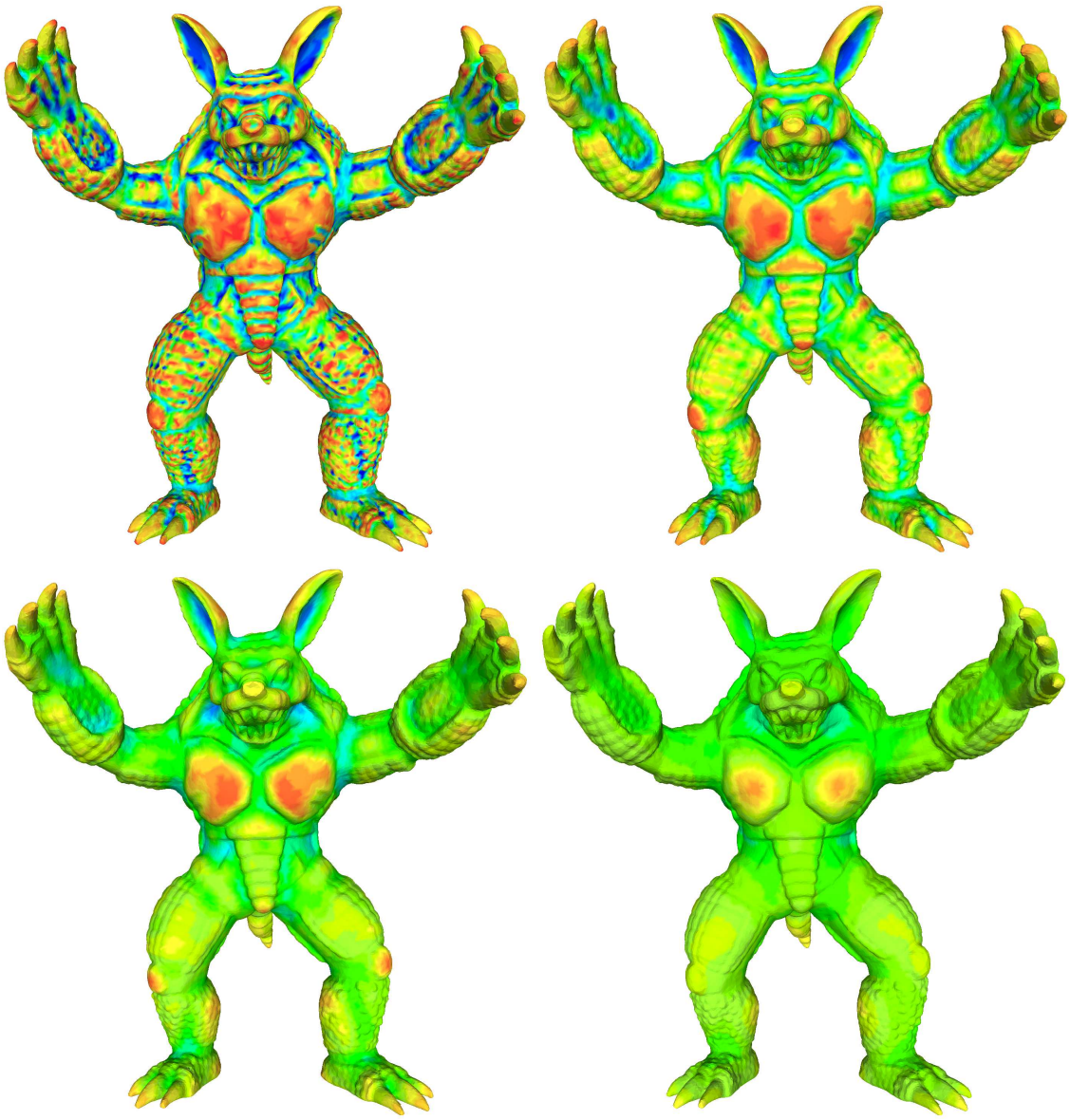


Figure 5.5: From left to right and top to bottom: mesh (40267 vertices) with SI values on the surface. Warm colors reflect positive SI (convex parts) and cool colors negative values (concave parts). First image shows the original SI values. The last three images are after applying Gaussian smoothing with σ_0, σ_3 and σ_6, where σ_0 equals $2 \times 0.3\%$ of the bounding box diagonal ($\sigma_0 = 1.3728$).

selected the SI value from the nearest vertex at the last scale of the previous octave. In this experiment to process the second octave we used the smoothed data from σ_3, but we could have used $\tilde{\sigma}_3$ applied to scale σ_0; we recall that they are equivalent. To compare the data from first and the second scale, smoothed by $\tilde{\sigma}_1, \tilde{\sigma}_2$ and $\tilde{\sigma}_3$ (relative scales), with the data smoothed by scales $\sigma_1 \ldots \sigma_6$, we used their normalized SI histograms, with 200 bins of SI $\in [-1, 1]$ and compared them using the χ^2 distance. The sensitivity of Chi-Square

Figure 5.6: Armadillo, with bounding box diagonal of 228.7947, smoothed by Gaussian kernel $\sigma_{15} = 43.9296$. As kernel size increases, the smoothing process leads to a convergence of SI to a global mean.

Figure 5.7: The left image was obtained by applying $\tilde{\sigma}_3$ to the original model with 40267 vertices. The right image was obtained by applying scale σ_3 for the 2nd octave and then reducing the model to 20134 vertices and half the volumetric size. Both images are shown at the same size in order to illustrate the similarities.

distance to quantization effects is, in our case, unimportant due to the use of normalized histograms with a constant number of bins [Pele and Werman, 2010]. The normalization was done by dividing the number of vertices in an SI bin by the total number of the vertices in the mesh. Figure 5.8 shows a comparison of normalized SI histograms in Gaussian scale space obtained by absolute scale and by decimation (hierarchical, first 2 octaves). The top row, from left to right, shows the histograms of the first octave, i.e., SI values after applying a Gaussian filter with size of σ_1 (red) or $\tilde{\sigma}_1$ (black), σ_2 (green) or $\tilde{\sigma}_2$ (black) and σ_3 (blue)

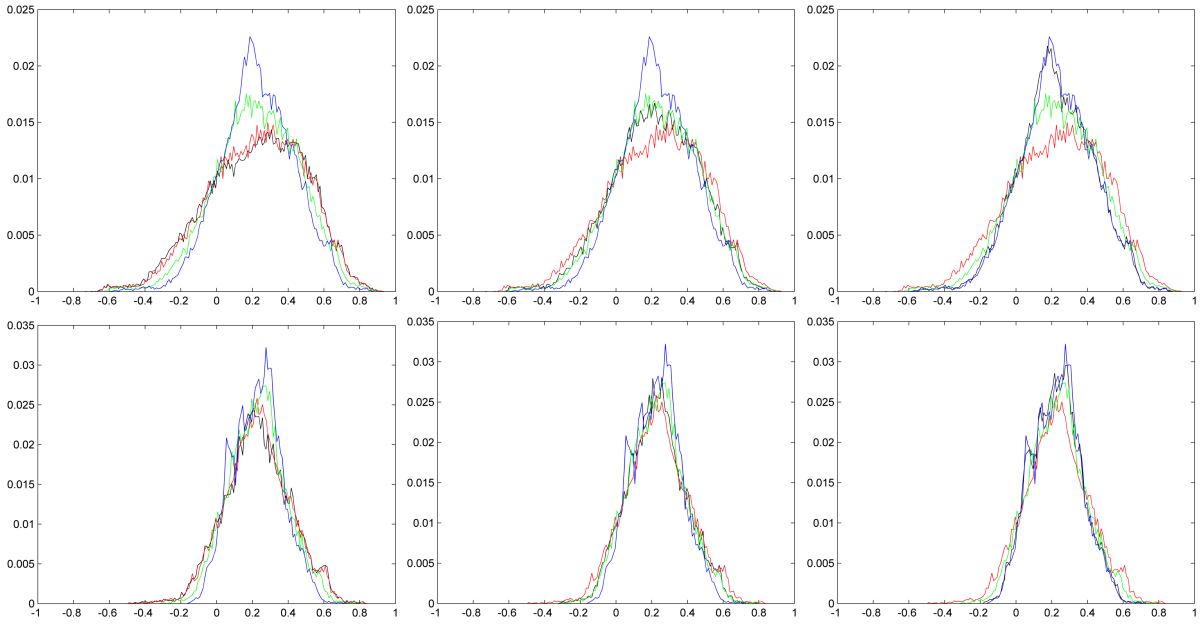


Figure 5.8: Armadillo histogram comparison of first 2 octaves. Top, from left to right: 1st octave SI histograms obtained at absolute scales σ_1 (red), σ_2 (green) and σ_3 (blue), and at relative scales $\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3$ (black). The colored curves are repeated in the three panels. The 1st graph shows $\tilde{\sigma}_1$, the 2nd shows $\tilde{\sigma}_2$, and the last graph shows $\tilde{\sigma}_3$, all in black. Bottom: 2nd octave histograms, after decimation, at absolute scales σ_4 (red), σ_5 (green), σ_6 (blue) and at relative scales within the octave.

or $\tilde{\sigma}_3$ (black). The bottom row, from left to right, shows the histograms after reduction to half and sub-sampling: σ_4 (red), σ_5 (green), σ_6 (blue), and $\tilde{\sigma}_1, \tilde{\sigma}_2$ and $\tilde{\sigma}_3$ (black). It can be seen that the histograms at the relative scales are almost identical to those at the absolute scales. Tables 5.2 and 5.3 show the χ^2 distances between the normalized histograms obtained by using absolute scales and relative scales. $G_{0,1..3}$ are the 3 relative scales in the first octave and $G_{1,1..3}$ are in the second octave. The lowest values (χ^2 distances) are in the diagonal entries which means that the histograms of Fig. 5.8 are closest as observed. They confirm the visual evidences of Figures 5.7 and 5.8. Therefore, the main conclusion is that any further processing can be done in the relative space scale. Because of the mesh reduction by decimation, this will be much faster than working in the space of absolute scales.

Table 5.2: χ^2 distance between the normalized histograms in the 1st octave of the Armadillo model.

	σ_1 (red)	σ_2 (green)	σ_3 (blue)
$G_{0,1}$	0.0045	0.0379	0.1025
$G_{0,2}$	0.0192	0.0041	0.0358
$G_{0,3}$	0.0753	0.0206	0.0041

Table 5.3: χ^2 distance between the normalized histograms in the 2nd octave of the Armadillo model.

	σ_4 (red)	σ_5 (green)	σ_6 (blue)
$G_{1,1}$	0.0063	0.0350	0.0992
$G_{1,2}$	0.0209	0.0069	0.0467
$G_{1,3}$	0.0704	0.0259	0.0060

5.3 Saliency maps

Saliency maps of 3D meshes were introduced by Lee et al. [2005a]. They applied a metric based on perception. The use of only geometric features, such as curvatures, has the disadvantage of missing perceptually important events in patterns: perception indicates a different and therefore important occurrence in the middle of a monotonous pattern. For example, a high local curvature in a large flat area, or a small flat area in a complex modulation pattern are both important. Measuring local importance with a center-surround mechanism, as proposed by Koch and Ullman [1985], has the capacity to identify zones with a different neighborhood context. The resemblance with visual receptive fields can be implemented as the difference between a fine and a coarse scale. Saliency maps of 3D objects are different from those of 2D images. An image with a uniform intensity has zero saliency. In the 3D case, the shape is intrinsic and independent of the light sources; for example, a sphere has zero salience because of its constant curvature. If someone would extract the saliency map from a 3D object, it seems reasonable to drive the process by measuring the changing curvature over the surfaces. Itti et al. [1998] proposed a bottom-up mechanism to aggregate all the local conspicuities, from intensity, orientation and color, in a final saliency map. Local conspicuity is obtained from the multi-scale, difference of scales, process.

There is evidence that human perception is more attracted to high-curvature areas, and more to concave than to convex areas [Lim and Leek, 2012]. In our work we follow the methodology of Lee et al. [2005a], but instead of using mean curvature we will use SI; see Section 3.2. The reason for using SI is the fact that it is based on principal curvatures, and it provides a form of surface classification with extreme values of $[-1, -0.625]$ and of $[0.625, 1]$ for a concave ellipsoid surface (cup) respectively a convex ellipsoid (cap); see

Fig. 3.5. These two types of surfaces have more discriminative power in terms of human perception. Common sense also states that human attention to 3D objects is attracted by prominent (salient) and concave zones. The work of Lee et al. [2005a] was used by Zaharescu et al. [2009] as a basis for their geometric interest point detector, MeshDOG.

To obtain the saliency map proposed by Lee et al. [2005a] we compute the principal curvatures by using the algorithm of Meyer et al. [2003]. Let M_{SI} define a mapping from each vertex v of the mesh to its SI value ($M_{\text{SI}}(v)$ of Equation (3.9)), and the neighborhood defined by $N(v, \sigma) = \{x : \|x - v\| \leq \sigma\}$ with x also being a mesh vertex. As distance function we consider the geodesic distance instead of the Euclidean one, due to possible errors introduced at bigger scales. The multi-scale scheme is build on a set of Gaussian filters, as explained in the previous section. The Laplacian, as second-order derivative, will measure the rate of change in the surface curvature, in our case the SI of the surface. Usually the Laplacian operator is applied after a Gaussian in order to reduce its sensitivity to noise. These two operations can be implemented in a single Laplacian-of-Gaussian (LoG) operator or, due to the high computational cost, by a Difference-of-Gaussians (DoG). The center-surround mechanism, in a mesh around vertex v, is implemented by applying a DoG. Consider $G(M_{\text{SI}}(v), \sigma)$, the Gaussian weighting of SI defined as

$$G(M_{\text{SI}}(v), \sigma) = \frac{\sum\limits_{x \in N(v, 2\sigma)} M_{\text{SI}}(x) \exp(-\|x - v\|^2/(2\sigma^2))}{\sum\limits_{x \in N(v, 2\sigma)} \exp(-\|x - v\|^2/(2\sigma^2))}. \tag{5.16}$$

The Gaussian filter is applied to $M_{\text{SI}}(v)$ and takes into account the 3D surface nature by using a geodesic distance of at most 2σ. Saliency is then computed, at different scales (σ) and for each vertex of the mesh. Figure 5.9 shows Armadillo with a vertex (in black) and its σ_3, σ_6 and σ_{15} geodesic neighborhoods.

Mesh saliency is determined as the absolute value of the DoG responses, at one fine and one coarse scale. As can be seen in Fig. 5.5, as the scale σ increases, the details on the surface disappear. One can see this process as decomposing the object into a set of scale-depending saliency maps (SM_p):

$$\text{SM}_p(v, \sigma_1) = \|G(M_{\text{SI}}(v), \sigma_1) - G(M_{\text{SI}}(v), \sigma_2)\|. \tag{5.17}$$

In general, $\sigma_2 = 2 \times \sigma_1$ because we allways apply octave intervals. Hence,

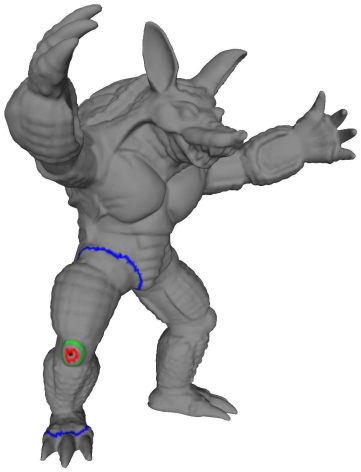


Figure 5.9: The geodesic neighborhoods of a black vertex on the knee cap at scales σ_3, σ_6 and σ_{15}, in red, green and blue. The Gaussian kernel of Eq. 5.16 was applied to twice the scale width ($\sigma_6 = 5.4912$, $\sigma_9 = 10.9824$ and $2\sigma_{15} = 87.8592$).

$$\mathrm{SM}_{p_i}(v, i) = \|G(M_{\mathrm{SI}}(v), \sigma_i) - G(M_{\mathrm{SI}}(v), 2\sigma_i)\|. \tag{5.18}$$

After producing a set of scale-depending saliency maps they are aggregated into one global saliency map SM which codes the most important features. The purpose of a final saliency map is to represent conspicuity by a scalar value, to drive attention to the most conspicuous areas. The blending of all SM_p is done in the same way as proposed by Itti et al. [1998]. The suppression operator

$$\mathrm{SM}_{p_{i(\mathrm{after})}} = \mathrm{SM}_{p_{i(\mathrm{before})}}(M_i - \bar{m}_i)^2 \tag{5.19}$$

is applied which emphasizes locations with a small number of strong peaks and which suppresses locations with numerous and similar peak responses. For computing the global saliency map we need to normalize each SM_{p_i} and apply the suppression operator at each scale. In Eqn. (5.19), M_i is the maximum saliency value in the neighborhood and $\bar{m}_i$ is the mean of all local maxima there (excluding the maximum M_i). Local maxima are detected inside the geodesic neighborhood of each vertex. For this we use the middle radius, between the fine and coarse scales, as limit to the geodesic distance. For example, in SM_{p_3} the scales are σ_3 and σ_6, and the middle radius is $(\sigma_3 + \sigma_6)/2 = 4.1204$[1]. The final SM is the sum of all scale-depending saliency maps, each of them processed by the suppression operator. Figure 5.10 shows the scale-depending saliency maps SM_{p_0}, SM_{p_3}, SM_{p_6} and SM_{p_9}.

[1]Itti et al. [1998] and Lee et al. [2005a] did not define how to compute local maxima, they probably used the maximum size of the octave.

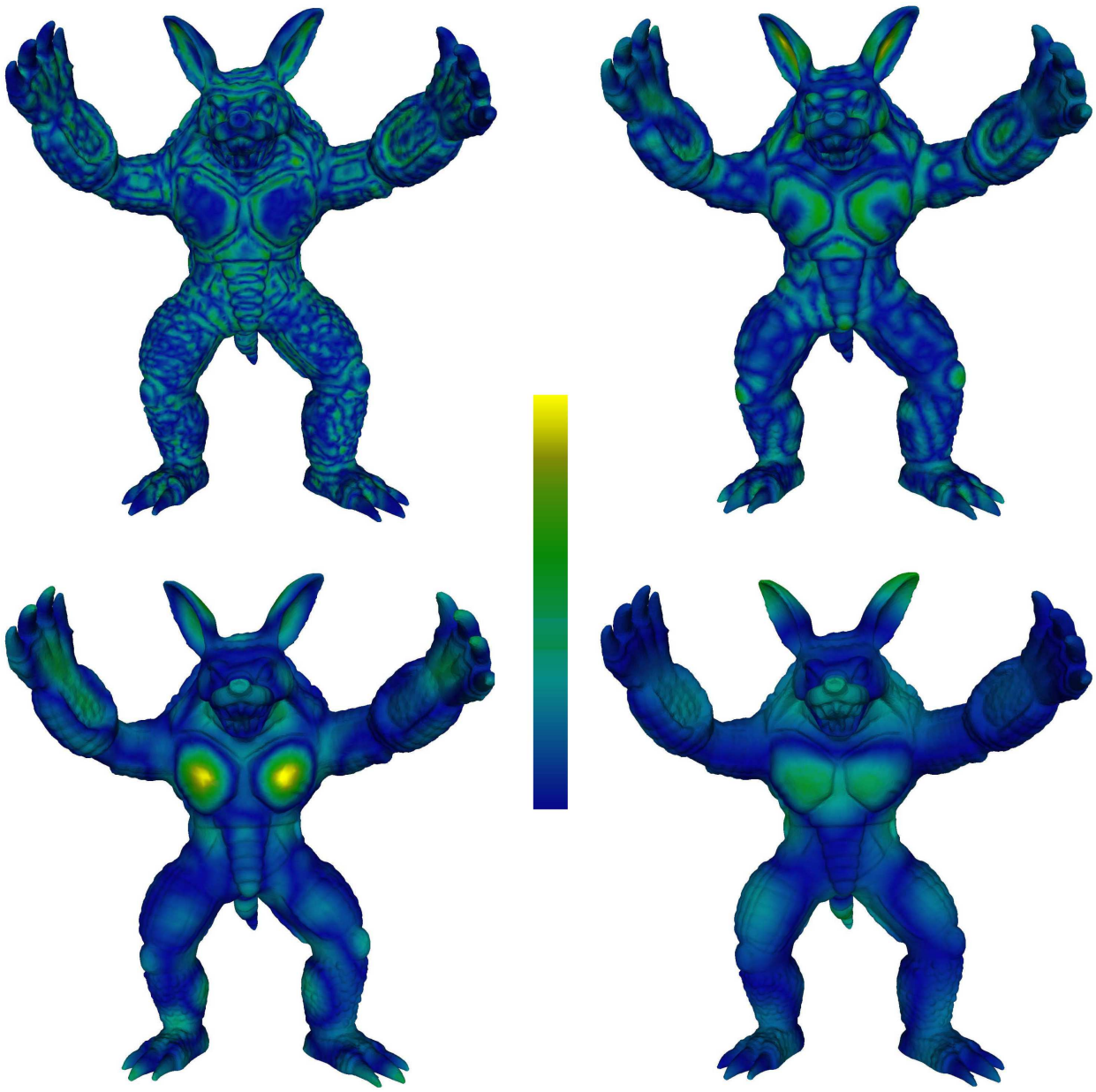


Figure 5.10: Armadillo model with 40267 vertices. From left to right and top to bottom: saliency maps of octave intervals σ_0-σ_3, σ_3-σ_6, σ_6-σ_9 and σ_9-σ_{12}. In the color bar: yellow (top) represents the maximum and blue (bottom) the minimum values in the four saliency maps. In this model these values are 0.5212 and almost zero.

We emphasize that the purpose of applying the SI operator first is the fact that, as the scale grows, the SM_p expose the bigger concave and convex areas. The knees and areas between fingers and toes of the Armadillo model are areas where SI is well defined in terms of convex and concave surfaces. Successive application of Gaussian filters with increasing size will influence surfaces classified as convex ellipsoid and cylindrical. For example, the fingertips are convex. As the filter grows, convex cylinders of the fingers are included in

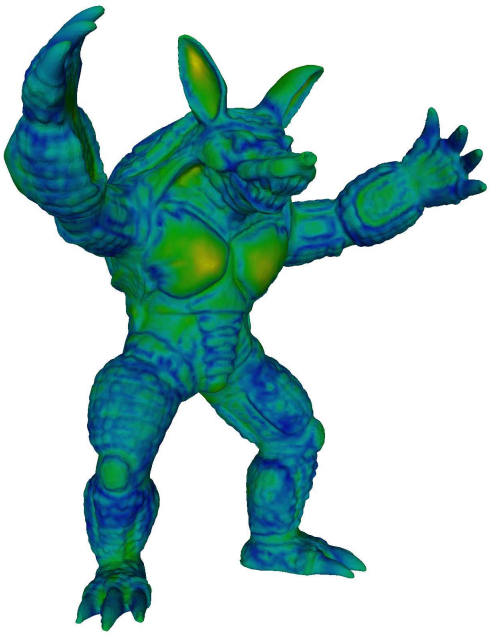


Figure 5.11: Saliency map of Armadillo model (5 octaves). It clearly shows the main concave and convex areas.

the weighting of SI and they will lower the SI values. The final saliency map of 5 octaves $\{\sigma_0\text{-}\sigma_3, \sigma_3\text{-}\sigma_6, \sigma_6\text{-}\sigma_9, \sigma_9\text{-}\sigma_{12}, \sigma_{12}\text{-}\sigma_{15}\}$ is computed as $\mathrm{SM} = \sum_i(\mathrm{SM}_{pi})$; see Fig. 5.11. The images of Fig. 5.12 show that SM has more regular convex and concave areas than the SI-colored mesh.

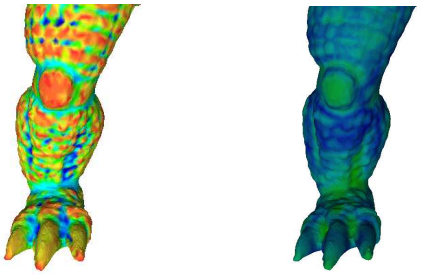


Figure 5.12: The left image shows the original, noisy SI values. The right one shows the final saliency map with more clear concave and convex areas at the knee and between the toes.

A detailed analysis of the evolution of SI values under Gaussian smoothing (G_{σ_i}), at some vertices located at interesting points, shows typical curves, see Fig. 5.13. The SI values of vertices located in valleys (concave ellipsoids) increase as the scale increases. In contrast, vertices in convex ellipsoid zones have decreasing SI values as the scale increases. The toe tip vertex (blue) shows a decreasing SI function, like the knee vertex (red), since both are

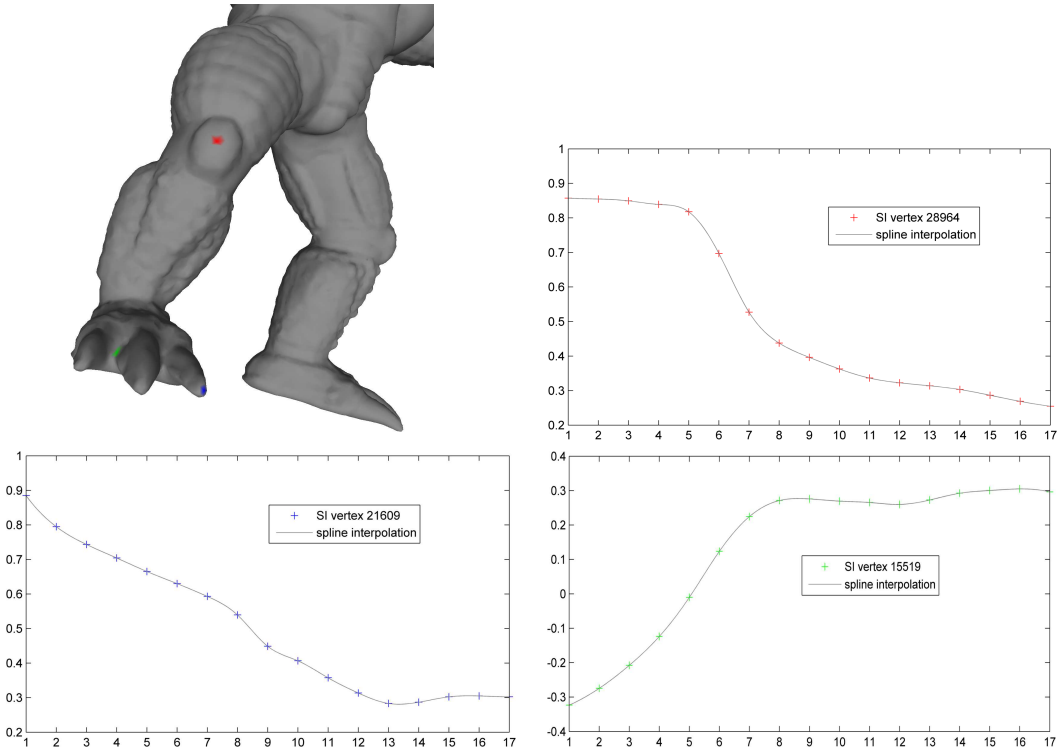


Figure 5.13: Evolution of SI values at characteristic vertices. From left to right and top to bottom: image showing three vertices at interesting points (convex/concave ellipsoid) and three characteristic curves over scales $\{\sigma_s, \sigma_0 ... \sigma_{15}\}$. The first curve belongs to the knee (red), the second (blue) to a toe tip and the last curve (green) is from a saddle point between toes.

protuberances, see Fig. 5.13. However, the shape of the function is different. At the knee, the function decreases with a low rate because its neighborhood vertices have a similar curvature. The SI value at the fingertip decreases faster because the neighbourhood vertices have values in the range $[3/8, 5/8]$ (convex cylinder). Vertices located at protuberances, like fingertips and toes, typically have two kinds of SI surfaces, namely a convex ellipsoids and a convex cylinder.

5.4 Keypoint detection

Saliency maps are used for directing attention to important regions. Other processing often involves detection of precise keypoints: they are used in registration, mesh operations such as simplification and segmentation, and the selection of suitable viewpoints. The ultimate objective is object matching and retrieval. Keypoints, also referred to as interest points, are also salient points. They provide features which are locally distinctive and they remain stable under most variations of the object. A variation of an object can be the result of an invariant

transformation, after which the most significant features are still present. Most of the work done on keypoint extraction addresses the tasks mentioned at the beginning of this section. Zhong [2009] presented a 3D shape descriptor called Intrinsic Shape Signature (ISS), which characterizes a local or semi-local region of a point cloud. To define the ISS descriptor, Zhong started by constructing a local coordinate system, its z-axis being aligned with the surface normal and the y and x axes are determined by eigen analysis of the points in the local vicinity. Then, a feature vector is constructed by using an occupational 3D histogram. The bins are formed using discrete levels of radial distance while building a regular, hexagonal, spherical grid (defining a local spherical neighborhood). The ISS descriptor is composed of two components: intrinsic local coordinate system (from local reference frame) and a feature vector, composed of the sum of weights from all points that fall in each bin.

Zaharescu et al. [2009] proposed two 3D descriptors for uniformly triangulated meshes. They used MeshDOG, based on DoG to capture and detect scale-space extrema. After locating an interest point, the MeshHOG operator (histogram of oriented gradients) is used to construct a descriptor of photometric information of the interest point. For surface representation and object recognition of range images, Chen and Bhanu [2007] introduced a local surface descriptor, LSP, characterized by its centroid, surface type and a 2D histogram (shape index values and angles between the normal of the feature point and its neighbors). Rather than computing LSP descriptors for all surface points, they only calculate them in areas with significant shape variation, using an extension of SI as proposed by Dorai and Jain [1997].

Many 3D keypoint detectors have been developed. In addition, some work has been done. Salti et al. [2011] evaluated the, at that time state-of-the-art, detectors. In their study they divided the detectors into two groups: fixed-scale and scale-invariant. In the first group, keypoints are found by the detectors in a defined spatial neighborhood (constant scale). A *quality* measurement is applied to each keypoint by considering the vertex or region around the vertex. Keypoints are chosen by maximizing the quality within their spatial neighborhoods. For the second group, scale-invariant detectors, an extension of the multi-scale space concept as used in image processing is applied. The keypoint detectors find points at distinct scales, but a characteristic keypoint will be stable over neighboring scales. As in the fixed-scale case, keypoints are selected by their quality in spatial position as well as scale. Final keypoints are chosen by maximizing the quality measurements over

space and over scales.

Dutagaci et al. [2012] proposed a strategy to measure the performance of keypoint detection algorithms by employing human-generated ground truths. Automatically detected keypoints are quantitatively compared to human-marked points. From the six keypoint detectors that they tested, we highlight two because our own work is related to these, i.e., mesh saliency by Lee et al. [2005a] and salient points by Castellani et al. [2008]. Keypoints are extracted from mesh saliency in two steps. First, a set of potential keypoints is selected with saliency values which are highest within a neighborhood (local maxima). From this set, vertices with saliency values which are higher than a threshold value are chosen as keypoints. The threshold is defined as the average of the local maxima. In the salient points method, keypoints are detected by first applying a Gaussian filter. Meshes are decimated at D levels, M^d with $d = 1, ...D$, with different vertex resolutions. Castellani et al. [2008] consider each resolution (M^d) as an *octave* because they see the variation of resolution as a jump of an octave in scale space. 3D saliency is measured by evaluating the displacement of a vertex, from its original position to the new one, after the Gaussian filtering. To detect the peaks, the maps in the multi-scale representation are normalized by using the Itti et al. [1998] approach together with an adaptive inhibition process. The inhibition process, which enhances peaks, is applied to each normalized map. Considering $v \in M^d$, if its value is higher than 85% of those in its neighborhood, than the value is kept, otherwise it is set to 0. The final saliency map is constructed by adding the contributions of all scale maps, after the inhibition process, and the final keypoints are all the points which are local maxima with a value which is higher than 30% of the global maximum [Castellani et al., 2008].

Saliency maps highlight important areas of an object's surface. It is possible to formally classify, according to their principal curvatures, extreme surfaces as concave or convex ellipsoids (cup or cap); Koenderink and van Doorn [1992]. Lim and Leek [2012] and Dutagaci et al. [2012] showed that, in human perception, curvature is very important in the recognition of 3D objects. In pursuing salient keypoints we will propose a method that extends the work of Chen and Bhanu [2007] to triangle meshes. Finding and clustering the SI values can be used for 3D object decomposition [Kalogerakis et al., 2010; Yan et al., 2012]. However, our method only uses vertices with extreme SI values in the ranges $[-1, -0.625[$ and $]0.625, 1]$.

5.4.1 SI keypoints

Our method uses clustered vertices in scale space in order to extract the most prominent vertices of the object. Consider the polygonal (triangle) mesh M defined by sets of vertices V and triangles F, see Section 2.2.1. Computing the SI values for all vertices of M will result in values in the range $[-1, 1]$. For the keypoints we threshold the vertices with SI values below -0.625 or above 0.625; see Fig. 5.14 with cap clusters in red and cup clusters in green, at scale $\sigma_s \approx 0$.

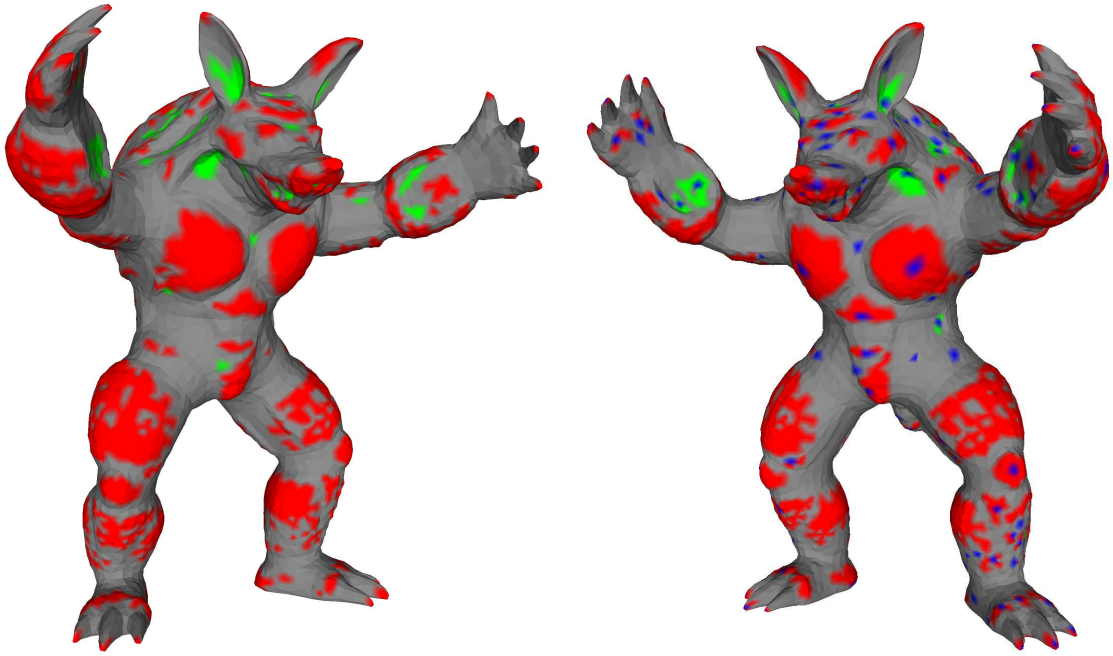


Figure 5.14: The left image shows Armadillo with SI values at scale σ_s (no smoothing). Clusters in red are vertices with SI > 0.625 (caps) and those in green are vertices with SI < -0.625 (cups). The right image shows keypoints, in blue, which are central vertices determined by the "betweenness centrality" of the clusters.

Once grouped, keypoint selection in each cluster is done by choosing the cluster's central vertex. For that, we use "betweenness centrality" of the vertices in a graph which counts the number of paths (shortest paths between any vertex pairs of the graph) that pass through each vertex. The vertex with the highest value is selected as keypoint, but the "betweenness centrality" metric restricts it to clusters with more than 3 vertices (clusters with less than three vertices are ignored). The right image in Fig. 5.14 shows keypoints, the *central* vertices within the clusters, in blue. We are going to apply this keypoint extraction in Gaussian scale space. Increasing the size of the Gaussian kernel, SI values of the keypoints converge to the

Figure 5.15: SI values after applying a Gaussian filter at scales $\sigma_0 = 0.0090$, $\sigma_1 = 0.0113$, $\sigma_2 = 0.0143$ and $\sigma_3 = 0.0180$. In red are vertices with SI values of $> 0.625, 0.6062, 0.5880$ and 0.5704 (caps), and in green are vertices with SI values of $< -0.625, -0.6062, -0.5880$ and -0.5704 (cups).

local mean and, depending on the local surface topology, vertices have small SI variations due to the smoothing introduced by the filtering. We reduce the threshold used to define cap and cup clusters in 3% steps of the initial value of 0.625.

Let $k_{\sigma_s}, k_{\sigma_0}, k_{\sigma_1}, k_{\sigma_2}$ and k_{σ_3} be the sets of keypoints selected at the scales indicated by the subscript. Our final set of keypoints is then

$$K_{M_{\sigma_3}} = k_{\sigma_s} \cup\ k_{\sigma_0} \cup\ k_{\sigma_1} \cup\ k_{\sigma_2} \cup\ k_{\sigma_3}, \tag{5.20}$$

where U is the union set operator. For each keypoint in $k\sigma_3$ and within the neighborhood of σ_3 we check the existence of keypoints of $k\sigma_s$, $k\sigma_0$, $k\sigma_1$ and $k\sigma_2$. The union operator only considers keypoints if there exist more than two keypoints in the σ_3 neighborhood; otherwise no keypoint will be considered. If there are more than two keypoints in σ_3, the one associated with the smallest sigma in $\sigma_0, \sigma_1, \sigma_2$ is selected. The σ_s is not taken into consideration because it is the original mesh which is exposed to noise and outliers.

Figure 5.15 shows the clusters of the Armadillo model after applying $\sigma_0 = 0.0090$, $\sigma_1 = 0.0113$, $\sigma_2 = 0.0143$ and $\sigma_3 = 0.0180$. Vertices are selected with SI greater or less than: $\pm 0.625, \pm 0.6062, \pm 0.5880$ and ± 0.5704. Despite the smoothing process in this small octave some keypoints are stable over scales, i.e., at the fingertips, nose and auricular cavity. If we increase the scale, many keypoints will disappear. The figure shows that the reduction of 3% of the threshold produces clusters with similar sizes (caps and cups), which increases keypoints stability.

In order to validate our algorithm we used the ground-truth set of Dutagaci et al. [2012]. They collected data by using a web page. Participants could mark interest points on a 3D object. Due to the subjective nature of human keypoint selection, the authors selected the most consistent annotations among the participants and eliminated outliers. They used two parameters to define the precision of the keypoint selections: the radius around each keypoint and the number of users who marked a keypoint within the neighborhood. The radius was σd_M, where d_M is the object diameter, the largest Euclidean distance of all vertex pairs, and σ is a scalar value used to define the radius as function of d_M. Keypoints are selected within a group of marked interesting points. The group is defined as all vertices with a longest distance between them which is less than $2\sigma d_M$. The second parameter is for rejecting a group if its number of interest points is less than n, the number of participants who marked the same points. Such a group is not eligible to have a representative keypoint. A keypoint is chosen as the point with the minimum sum of geodesic distances to all points in the group. Overlapping of groups can lead to situations where their representative keypoints are too close. To avoid redundant representation, if the distances between them are less than $2\sigma d_M$, the group with the smaller number of interest points is discarded.

For a particular object M and given ranges of parameter values n and σ it is possible to determine a ground truth set $\mathcal{G}_M(n, \sigma)$. As n grows there are less points because the number of participants that choose small details as interest point will be smaller. A bigger σ yields more points, because a larger variation in the localization will be accepted. On the other hand, this can also lead to the merging of representative keypoints. Figure 5.16 shows Armadillo's ground truth with $\sigma = 0.03$ (of the diagonal of the bounding box) with $n \in 2, 5, 10$. We can see that the number of ground truth points decreases when n increases. This is also due to the subjectivity of the human selection of ground truth points.

After estimating the ground truth, Dutagaci et al. [2012] evaluated six algorithms by

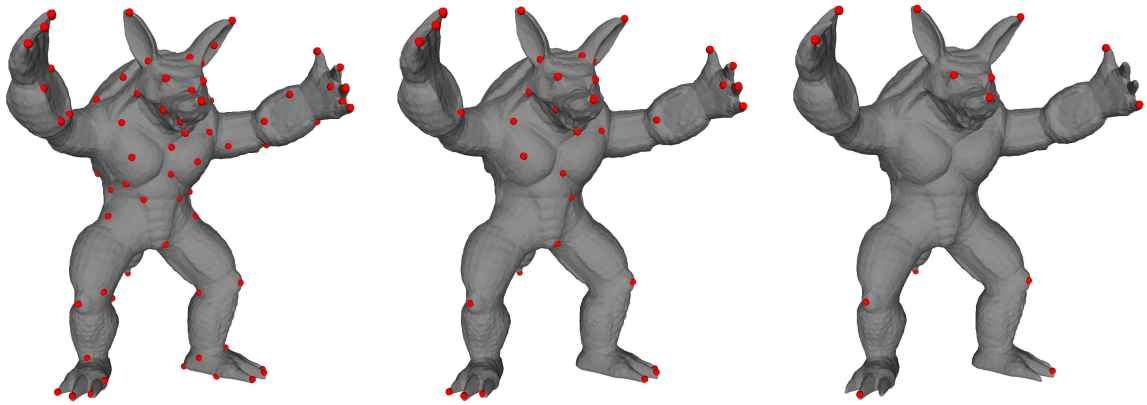


Figure 5.16: From left to right: ground truth points selected by humans at $\sigma = 0.03$ and $n = 2, 5, 10$ (number of participants).

measuring their performance relative to human perception. They computed False Positive, False Negative and Weighted Miss Errors. Denoting $\mathcal{G}_M(n, \sigma)$ as $\mathcal{G}$ and the set of detected keypoints of object M as A, a point $p \in A$ is considered correctly detected if there exists a point $g \in \mathcal{G}$ in a vicinity of radius r such that $d(g, p) \leq r$, with $d(g, p)$ the geodesic distance and r is the parameter that controls the localization tolerance. If N_C is the number of points correctly detected in $\mathcal{G}$, then the False Negative Error as a function of r is

$$\text{FNE}(r) = 1 - \frac{N_C(r)}{N_G}, \tag{5.21}$$

with N_G the number of points in $\mathcal{G}$.

The second measure takes into consideration that all points in A without a close point in $\mathcal{G}$ are false positives, denoted by N_F. If N_A is the number of keypoints detected by the algorithm, then $N_F = N_A - N_C$, and the False Positive Error as a function of r is

$$\text{FPE}(r) = \frac{N_F(r)}{N_A} = \frac{N_A - N_C}{N_A} = 1 - \frac{N_C}{N_A}. \tag{5.22}$$

Normalization of the number of false positives is done by the number of keypoints detected by the algorithm, instead of by the number of true negatives, because the latter depends on the mesh resolution. False Negative Error does not consider the importance of individual ground truth points. If a point has been selected by n participants, it contributes in the same manner as the others. The third measure corrects the False Negative Error by taking into consideration the prominence of an interest point. This is done by introducing the number of hits by the participants as weight. For a ground truth point $g_i \in \mathcal{G}$ in the geodesic vicinity (r) and for the number of participants n_i who marked it as an interest point, the Weighted

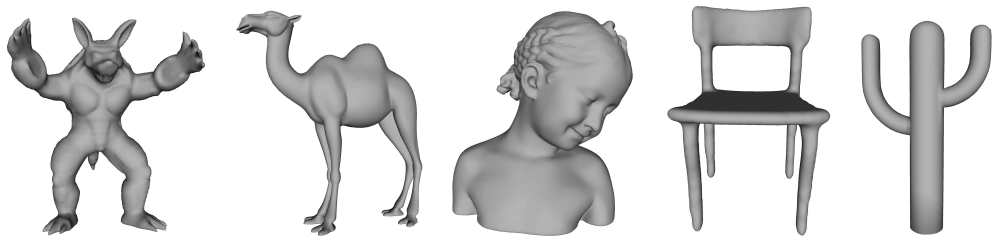


Figure 5.17: From left to right: Armadillo, Camel, Girl, Chair and Cactus.

Miss Error is

$$\mathrm{WME}(r) = 1 - \frac{1}{\sum_{i=1}^{N_G} n_i} \sum_{i=1}^{N_G} n_i \delta_i \;\; ; \;\; \delta_i = \begin{cases} 1 & \text{if } g_i \text{ is detected by the algorithm} \\ 0 & \text{otherwise.} \end{cases} \tag{5.23}$$

This will measure the capability of an algorithm to detect interest points which are significant in terms of human perception, correcting the weakness of FNE by using n as a hard weight factor.

5.4.2 Evaluation and results

In order to evaluate our SI keypoint algorithm, we used the data of Dutagaci et al. [2012] of the following models: Armadillo, Camel, Girl, Chair and Cactus; see Fig. 5.17. We applied our algorithm to these objects and detected keypoints were evaluated by comparing them with the human ground truth. In Dutagaci et al. [2012] the evaluation included the following detection methods: Mesh saliency [Lee et al., 2005a], Salient points [Castellani et al., 2008], 3D-Harris [Sipiran and Bustos, 2010], 3D-SIFT [Godil and Wagan, 2011], SD corners [Novatnack and Nishino, 2007] and Heat Kernel Signature (HKS) [Sun et al., 2009].

Focusing on the Armadillo object, it has different types of interest points: facial, extremities at fine and coarse scales, and other relevant points such as at knees and chest. Our SI-based algorithm has a similar behavior as the Mesh saliency method, and it is a little better than the Salient points method; see Fig. 5.18. The latter method does not detect keypoints at the chest. The SI method is also sensitive to local protuberances and it does not detect keypoints in flat or smooth surfaces. The number of detected SI keypoints is less than Mesh saliency and SD-corners keypoints. This is due to the large scale space interval applied. If we reduce the interval between the maximum and minimum scales, our method will detect a larger number of keypoints and this will increase the number of false positives. If compared to the HKS method, which because of its spectral nature only detects protu-

berance points of the global object structure (nose, finger and toe tips, tail and ear tips), our method will detect all the points detected by HKS plus keypoints at chest, knees and eyes. The points near ear and nose tips are very close to ground-truth points because of the representative vertex of the clusters (chosen by the "between centrality" criterion). When compared with 3D-Harris and 3D-SIFT, our method performs better. 3D-Harris misses the keypoints at the chest and produces too many keypoints at Armadillo's hands and feet, and 3D-SHIFT cannot locate the keypoints at chest and knees and misses some keypoints at finger and toe tips; see Fig. 5.18.

A full, quantitative evaluation of SI keypoints was done by using the introduced measures, namely FNE, WME and FPE; see Fig. 5.20. Each graph shows, with respect to the error tolerance r, the performances of our method and those of the other six methods. A good detection algorithm is expected to have lower errors as r increases. A quick drop of FNE reveals that the algorithm detects keypoints with a low localization error. Over all five objects, the graphs show that our method has almost the same performance as the other methods. On the basis of the WME and FNE values we can infer the precision of our method. If WME is lower than FNE then a method has a tendency to miss most of the salient points as marked by humans.

A careful observation of the graphs shows that our SI method has the 3rd lowest values for the Armadillo, Camel and Girl objects; it ranks 3rd place. However, it has the penultimate rank for the Chair and Cactus objects. This poor performance is caused by the nature of these objects: they contain long cylinders and the SI method tends to detect keypoints at extremities. However, our method performed better than HKS because it can detect keypoints in concave areas; see Fig. 5.19.

5.5 Discussion

We extended the Gaussian scale space to 3D. Some research has been done for range images, where there exists some regularity of the points. Here we focused on *non-uniform* triangle meshes representing 2-manifolds. Moreover, we applied the formal concept of octaves in order to decimate the mesh in the next coarser octave. As far as we know this has not been done yet, and the reduction to half of the number of vertices leads to reduced computation times. Formally, in scale space we can create an almost continuous and linear scaling with hundreds of scales, but due to practical reasons we used 0.3% of a model's bounding box

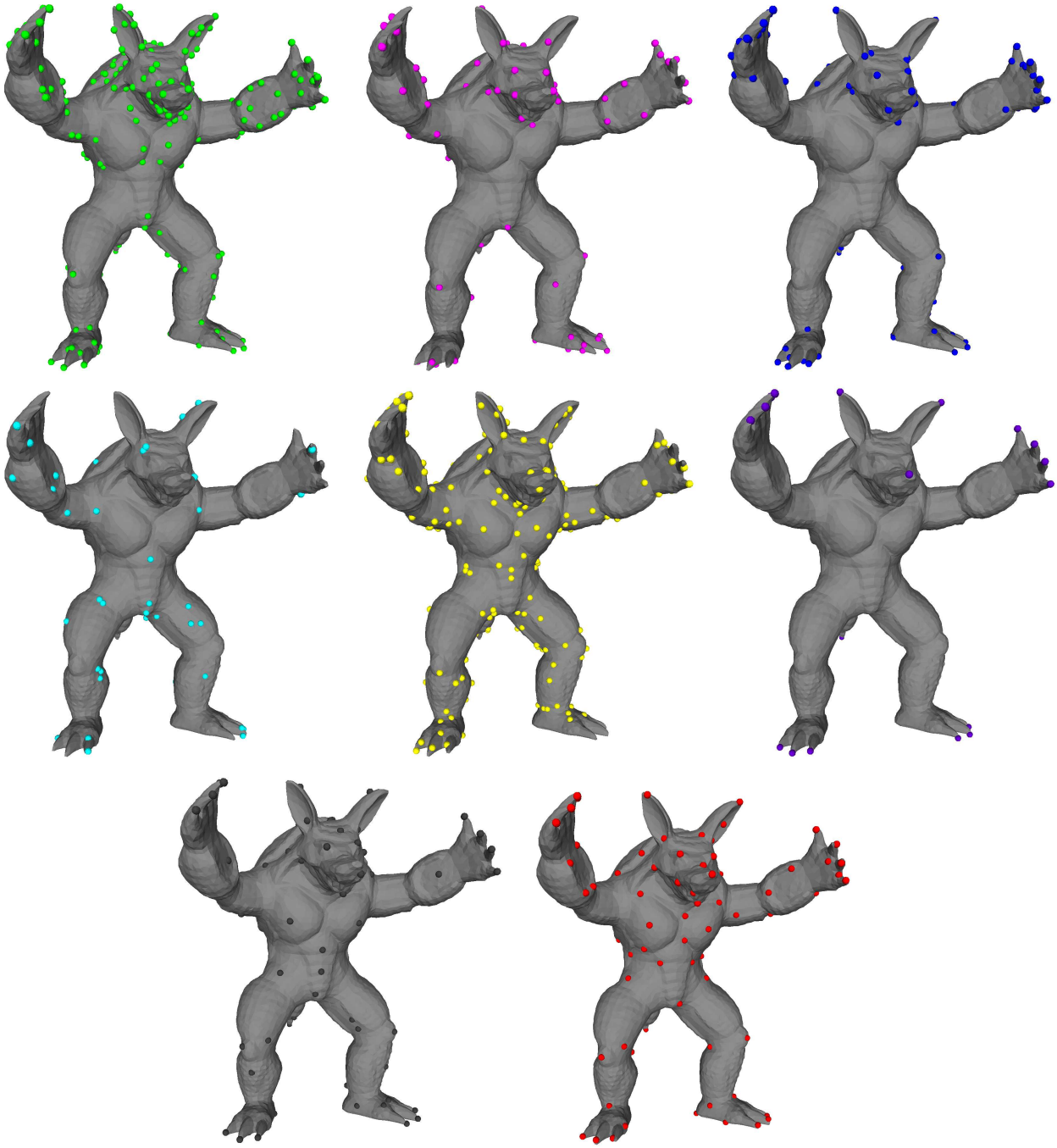


Figure 5.18: Keypoints extracted by the algorithms. From left to right and top to bottom: Mesh saliency, Salient points, 3D-Harris, 3D-SIFT, SD-corners, HKS and SI keypoints. The last, with red keypoints, is the ground truth with $\sigma = 0.03$ and $n = 2$.

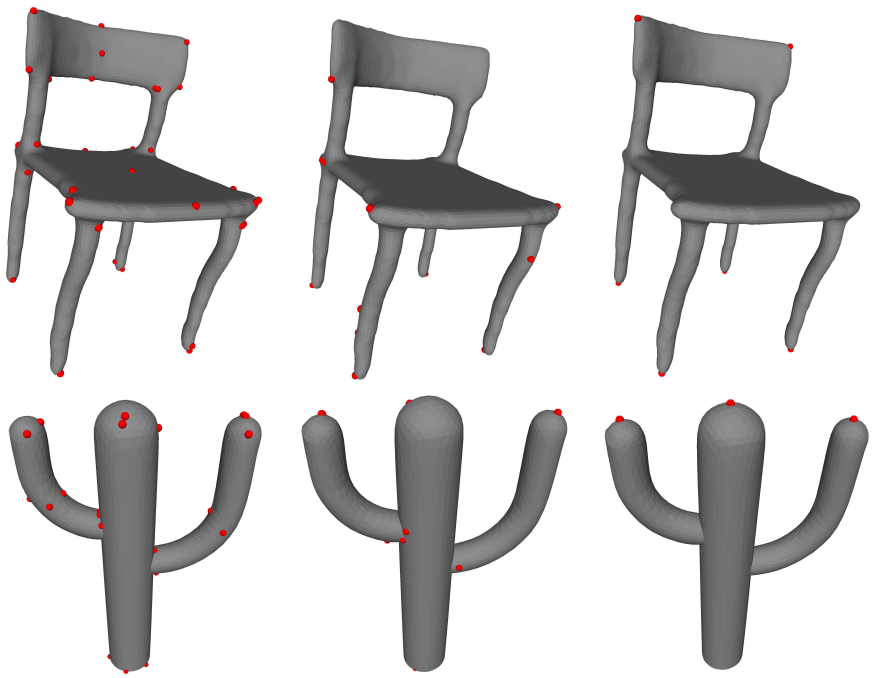
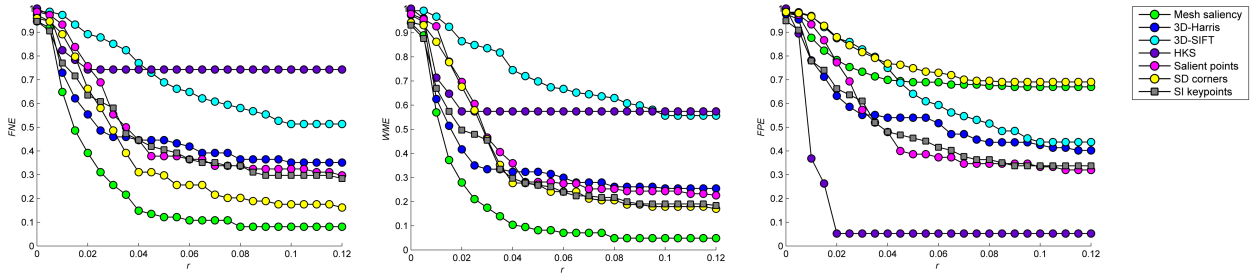
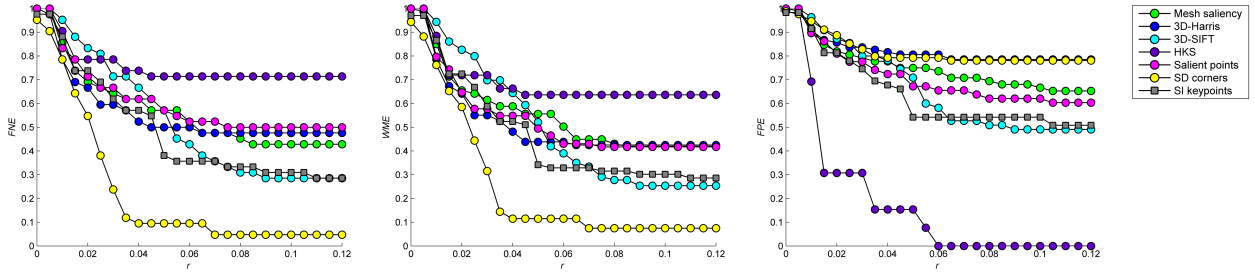


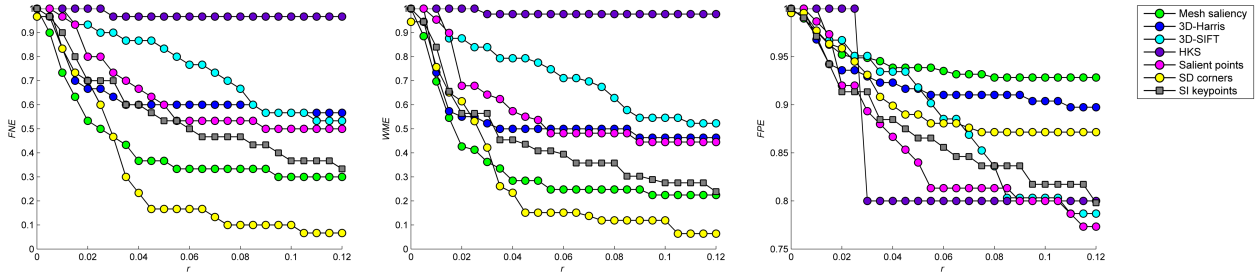
Figure 5.19: Keypoints extracted on models with cylindric areas, Chair (top) and Cactus (bottom). From left to right: ground truth, SI keypoints and HKS keypoints. The ground truth set was defined using $\sigma = 0.03$ and $n = 2$.



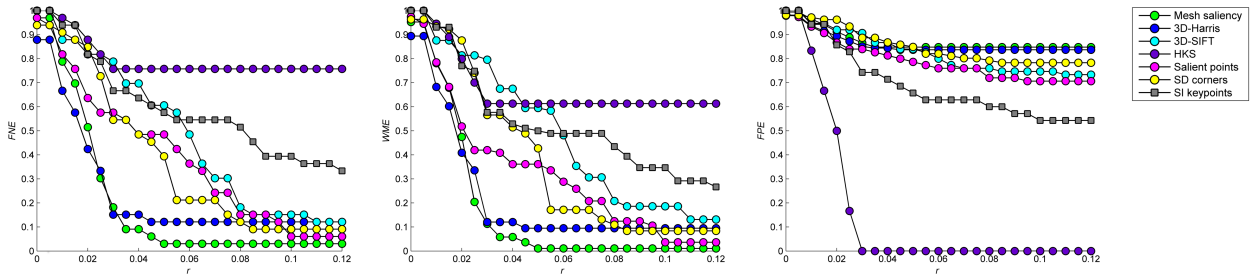
(a) Armadillo



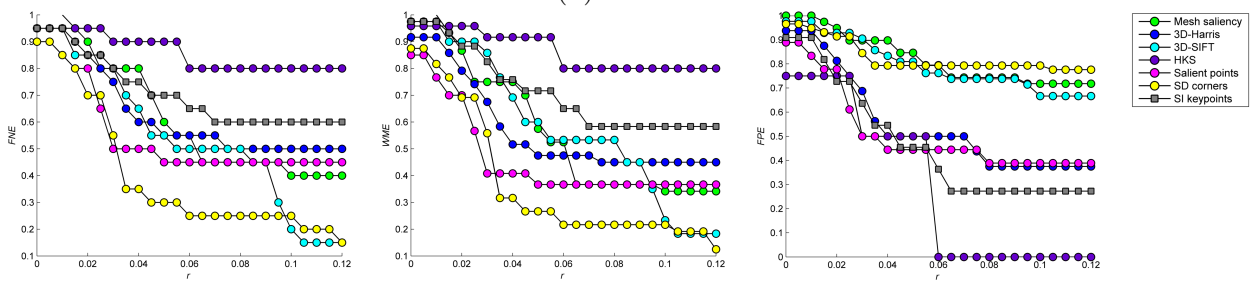
(b) Camel



(c) Girl



(d) Chair



(e) Cactus

Figure 5.20: FNE, WME and FPE error graphs of Armadillo, Camel, Girl, Chair and Cactus as a function of the localization tolerance parameter r. The ground truth set of $\sigma = 0.03$ and $n = 2$ was applied.

diagonal in order to select successive scales. Here we applied only 2 octaves in order to illustrate the implementation (Fig. 5.8).

3D Gaussian scale space can be used to extract saliency maps and keypoints based on Difference-of-Gaussians (DoG). We enhanced the mesh Saliency method of Lee et al. [2005a] by using the Shape Index, because SI indicates surface type and a possible segmentation of the object surface in terms of convex and concave parts. Recently SI was applied in a similarity analysis of 3D protein models [Yao et al., 2016].

Zaharescu et al. [2009] presented MeshDOG, a 3D keypoint detector for *uniformly* triangulated meshes. Because of our 3D Gaussian multi-resolution scale space we could present a new keypoint method which is more simple, fast and stable. We showed that the Shape Index can be applied at different scales and to *non-uniform* triangle meshes. Recent research on 3D data analysis, for instance for object recognition and for shape retrieval, has encouraged new methods. To achieve a good performance, almost all techniques apply keypoint detection to the 3D surface. Only recently appeared studies with performance evaluations of state-of-the-art methods for 3D keypoint detection [Salti et al., 2011; Dutagaci et al., 2012; Filipe and Alexandre, 2014]. Dutagaci *et al.* constructed and made publicly available a ground-truth set to evaluate keypoint detection methods. Their data allowed us to evaluate our own method and to compare it with the other techniques assessed in their paper. A few conclusions can be drawn. First, more research should be devoted to defining the best scale intervals depending on the nature of the 3D object. It seems that for the Armadillo object we used a bigger scale interval than that used by the mesh saliency method. Using a smaller scale interval (more scales in each octave and/or more octaves) may increase the stability of keypoints in the octave and leads to a larger number of keypoints. Second, some other form of choosing representative keypoints in clusters, instead of the "betweenness centrality" criterion, might improve the performance of our method. The keypoint chosen by the higher "betweenness centrality" value in an SI cluster may not yield the best representative vertex. The use of higher SI values after Gaussian filtering could produce better results. Finally, we should not forget that keypoint detection is only the first step in many processing schemes. The next steps are to extract feature vectors at the keypoints and then to apply these vectors to object recognition.

Chapter 6

Concluding remarks

Abstract: This chapter outlines the work presented in the thesis with the contributions and some guidelines for future research.

6.1 Summary

In this work we present a method for retrieving 3D polygonal objects. The categorization task is more complex then recognition. To increase categorization performance we introduced a new insight into saliency map of 3D objects and extraction of keypoints. By extending the concept of pyramidal hierarchy subsampling of image processing to object polygonal mesh we reduced the computational costs associated to the high number of object vertices. Recognition was achieved by using information obtained over the smoothing of the object's shape.

In summary, chapter 1 has a short introduction to the subject and the motivation that drives researchers to object recognition. Afterwards, a brief overview over the aspects of object recognition with especial attention on 3D. Brief review of the state-of-art is also done.

In chapter 2, it is presented an overview of registration and representation of 3D objects. Introductory attention was also given on its features and how can the object be fairly characterized, and how invariance can be achieved.

In chapter 3, the object's shape was studied to stablish the importance of local and global features in three-dimensional objects characterization.

In chapter 4, a new method for categorizing and recognizing 3D objects was proposed. To ensure the method invariance to translation, rotation and size, a process to do it in polygonal mesh was presented. In order to validate the method two experiments were done. In the first experiment, each object is characterized by two sets of multi-scale signatures: (a) the progression of deformation due to iterative mesh smoothing, and also iteratively (b) the volumetric variation between the two meshes, dilated and eroded by use of a sphere with increasing radius. The experiment was validated on a dataset of 32 complex objects, and results were measured in terms of Euclidian distance for ranking those objects. It achieved an overall average ranking rate of 1.84. In the second experiment, we replaced the features obtained on progression of the difference of volumes, computed with incipient dilation and erosion operators (Sect. 4.3.1) by a volumetric difference provided by formal mathematical morphology operators, dilation and erosion. The latter experiment, was conducted on a dataset of 40 complex objects with mesh resolutions different from those used in the feature vector construction. By using all eight features, the average ranking rate obtained was 1.075: 37 objects were ranked first and only 3 objects were ranked second. Additional tests were carried out to determine the significance of individual features and all combinations. The same ranking rate of 1.075 can be obtained by using some combinations of only three features.

In chapter 5, we extended a concept of signal processing, the *octave* to 3D object's polygonal mesh. The proposed multi-scale scheme allows decimation/sub-sampling after smoothing without loss of information. It was also presented, based multi-scale over SI, mechanisms for: a) saliency maps contruction and b) keypoints extraction. The obtained keypoints were tested against the human generated ground-truth set of Dutagaci et al. [2012]. The results are quite similar to the algorithms compared by Dutagaci evaluation study.

6.2 Contributions

The practical use of multi-scale methods in 3D shape analysis is dependent of the type of the representation of the 3D object. Some research has been done with range images, where exists some regularity of the points. Here, we focused on non-uniform triangle meshes representing 2-manifolds. We extended the Gaussian scale space to 3D polygonal meshes. Moreover, we applied the formal concept of octaves in order to decimate the mesh in the next coarser octave. As far as we know this has not been done yet, and the reduction to half

of the number of vertices leads to reduced computation times.

The actual computer vision challenge is to develop reliable methods for classification and recognition of 3D objects or shapes. Mesh smoothing allows to reduce noise and the number of vertices by a re-triangulation of planar or quasi planar areas. Apart from reducing the number of triangles, the smoothing process can eliminate structural information, precisely that information which is typical for characterizing an object's shape. We present a method for retrieving 3D polygonal objects by using two sets of multiresolution signatures. Both sets are based on the progressive elimination of object details by iterative processing of the 3D meshes.

The two main contributions on this thesis are: First, **a new understanding of three dimensional signal processing on 3D object represented polygonal mesh.** Multi-scale scheme used in classical signal processing implemented with smoothing filtering, by using Gaussian kernels. The kernel dimension (*radius*) is normally increased in regular paces. The *octave* concept is formally defined as domain bounded by a frequency and its double, this postulate allows, for instance, to reduce computation times in image processing by using sub-sampling (decimation) the image of the next octave. Extending this concept to 3D objects, brings two main advantages to geometric processing field: *a*) could be applied on objects represented by polygonal meshes, reducing the object mesh size. The sub-sampling (decimation) is ensured by existent algorithm of Garland and Heckbert [1997], and by using the same set of kernels we reduce the object's size in half, *b*) as a consequence it is possible to apply saliency map and keypoints operators in very fast way. In addition, processing the second octave will take half of the time and half of the memory resources used in the first octave, the third octave will take half of the second, or one quarter of the first octave, and so on...

This major contribution leads also to three small contributions: 1) single scale keypoint detection and extraction using Shape Index, and local surface categorization scheme (Shape Index based); 2) a multi-scale keypoints representation, based on the regularity of single scale responses; and 3) saliency maps construction based on multi-scale DoG information (on Shape Index values).

Second main contribution, **improved method for the recognition of three dimensional objects.** Consists in the development of a robust and accurate method that uses area and volumetric variation to define a two component set to be used as a characteris-

tic feature vector. Despite the initial intention to categorization that was not completely achieved. However, recognition was successfully accomplished.

6.3 Directions for further research

Despite the success in the recognition task, we need to pursuit the ability to categorize 3D objects. With the knowledge acquired in chapter 5, on 3D keypoints, saliency maps and the possibility of surface segmentation it is possible to do an exploratory research in gathering the information that *links* global features, providing recognition success to partial (local) features. The relations between local and global features may provide information that allows to assort in a specific class/category. Humans categorize/recognize in a parallel and dynamic way, from primary visual cortex V1 and higher areas, read Sect. 2.4. The integration of global features with multi-scale keypoints, as well as segmented surfaces from the object, must be explored. Knowing how this integration is done may give some clues to shape semantics, and to define the shape ontology. We should remark that retrieving 3D objects under textual searching methods are not effective, special structures with descriptions in geometric and/or topological characteristics seems to be the solution.

One of the Holy Grails in computer vision is the 3D shape canonical representation, which can only be achieved if: *a*) we precisely know how the human visual cortex represents a 3D shape. It is absolutely necessary to understand how this information, obtained in multi-scales (from coarse to fine), achieves its 3D representation from 2D projections of objects; and *b*) we build an ontology of 3D shapes, using it to conceptualise the knowledge of each shape. This allows us to link terms to concepts, and infer potential implied information. When it is possible to achieve this degree of detailed information, it will also be possible to conceive systems that could rival the human capacity to categorize.

"Each problem that I solved became a rule which served afterwards to solve other problems."

Rene Descartes (1596-1650), "Discours de la Methode".

Appendix A

Mathematical morphology approach

Mathematical morphology is based on set-theoretic concepts and its main operators, erosion and dilation, can be seen as operations between two sets. In image processing the sets are defined in R^2 [Serra, 1986], but in general one can work in R^n [Shih, 1991]. If A is a set of points $p = (x_1, x_2, \ldots, x_N)$ in Euclidean space R^3, then the *binary* set A defines the surface and interior of a 3D object. Furthermore, if B is also a set in R^3, $(B)_x$ denotes the translation of B by x and $\hat{B}$ is the reflection of B. The dilation of A by B, denoted $A \oplus B$, is defined by

$$A \oplus B = \{x \in R^3 \mid (\hat{B})_x \cap A \neq \emptyset\}. \tag{A.1}$$

Likewise, the erosion of A by B, denoted $A \ominus B$, is defined by

$$A \ominus B = \{x \in R^3 \mid (B)_x \in A\}. \tag{A.2}$$

The set B is called structuring element. In this work we will only use 3D spheres as structuring elements. If applied to an object A, dilation will lead to a bigger object and erosion to a smaller one, but both new objects will lack detail smaller than the size of the sphere. Such details are also lost when the opening and closing operators are applied, which combine the dilation and erosion operators [Matheron, 1975; Serra, 1982]. Finally, boundary extraction in 2D, see Serra [1982], can be defined by

$$\beta(A) = (A \oplus B) - (A \ominus B) \tag{A.3}$$

if the structuring element (sphere) is very small. In 3D, the boundary β is also calculated on the dilated and eroded versions of object A when B is a sphere with radius r and $r \downarrow 0$. β is the surface of A and this is related to the fractal dimension [Soille and Rivest, 1996]. As for mesh smoothing, Section 4.3.1 and [Lam and du Buf, 2009], the basic idea is to characterize

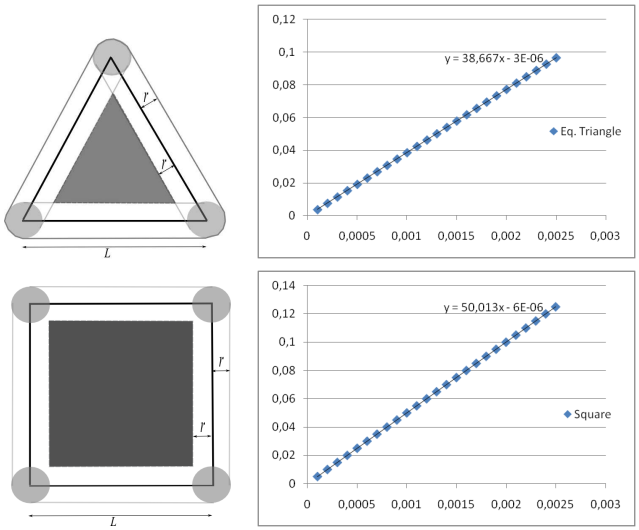


Figure A.1: Left: 2D erosion and dilation of an equilateral triangle and a square. Right: area β as a function of radius r (x axis) of the structuring element.

3D objects by a controlled elimination of detail. This is illustrated in 2D in Fig. A.1. The left shows a triangle and a square with the structuring element, a circle with radius r, on the corners of the original objects. The dilated objects are bigger (only the contours are shown) and the eroded objects (in grey) are smaller. The surface β between both as a function of radius r is shown to the right: the two curves are linear but have different slopes. This effect is exploited in the 3D case, in Section 4.3.2.

There are a few important issues when applying mathematical morphology to 3D objects. One is associated with the type of representation: voxel or mesh [Campbell and Flynn, 2001; Shih, 1991]. The voxel representation involves 3D arrays with, depending on the object's resolution, very big dimensions, although the voxels themselves are binary: object vs. background. An advantage is that many algorithms from mathematical morphology have been developed for 2D image processing, and these can easily be adapted to 3D. Polygonal meshes, on the other hand, have a more complex data structure. After applying the erosion and dilation operators, the new meshes must be determined, very close vertices can be collapsed, and self-intersecting facets must be detected and removed. In our method we extend boundary extraction (A.3) from 2D to 3D. Due to the fact that we use polygonal meshes we can apply a similar solution. If $A^c = 1 \backslash A$ is the set outside A, then

$$\beta(A) = A^c \cap (A \oplus B) + A \cap (A \ominus B)^c \tag{A.4}$$

is the sum of the expanded and shrunken volumes, i.e., the difference volume.

In order to limit distortions in the transformations, we iteratively apply a sphere of which

the radius r is a function of edge lenght. To avoid inconsistencies between different mesh resolutions, we select $r = \hat{L}/20$, where $\hat{L}$ is an object's edge length with the maximum occurrence. This can be easily determined by filling a length histogram with 50 equal bins from $L_{\min}$ to $L_{\max}$ of each object.

Because of the mesh representation, after applying the erosion and dilation operators, (i) vertices within the neighborhood defined by the structuring element should be merged, and (ii) self-intersecting facets must be detected and removed. All this is done without introducing distortions and by keeping the mesh closed, 2-manifold and without degenerated facets.

Dilations are obtained by displacing all vertices a distance r (the radius) in the direction of the normal vector. Since normal vectors always point outside, this is $-r$ in the case of erosions. Both operators are applied in two distinct steps. The first one is intented to acquire the volumes of the objects after the erosion/dilation process. Each operator is repeatedly applied until the first self-intersection occurs. In this step we do not remove any element of the mesh, vertex nor facet. In the second step we use the dilated (biggest) and the eroded (smallest) objects, generated in the first step, as a new starting point. The operators are repeatedly applied to the corresponding object: erosion to the smallest and dilation to the biggest object. After each erosion/dilation, we search the mesh for vertices that have a neighbor vertex in their vicinity, i.e., in the sphere with radius r centered at the vertex being processed, V_p. If there is a candidate vertex, V_c, it must be connected to V_p by at most 3 edges but it may not possess a direct edge to V_p. These restrictions must be satisfied in order to keep the mesh 2-manifold. The search for the vertices with the shortest path from V_p to V_c is done by using Dijkstra's algorithm. Vertices V_p and V_c are merged by removing all edges and vertices, which causes a gap in the mesh, and then by inserting a new vertex, V_f, with coordinates equal to the average of the removed vertices. In the last step V_f is connected to the vertices forming the gap; see Fig. A.2 (left). The elimination of self-intersecting facets is also necessary in situations where the nearest vertex is outside the vicinity sphere, the structuring element. Figure A.2 (right) shows two situations which both lead to a self-intersection. Elimination is done using the TransforMesh Library [Zaharescu et al., 2007], i.e., the method `ensureEdgeSizes` with the flags `fixDegeneracy=Yes` and `smoothing=No`.

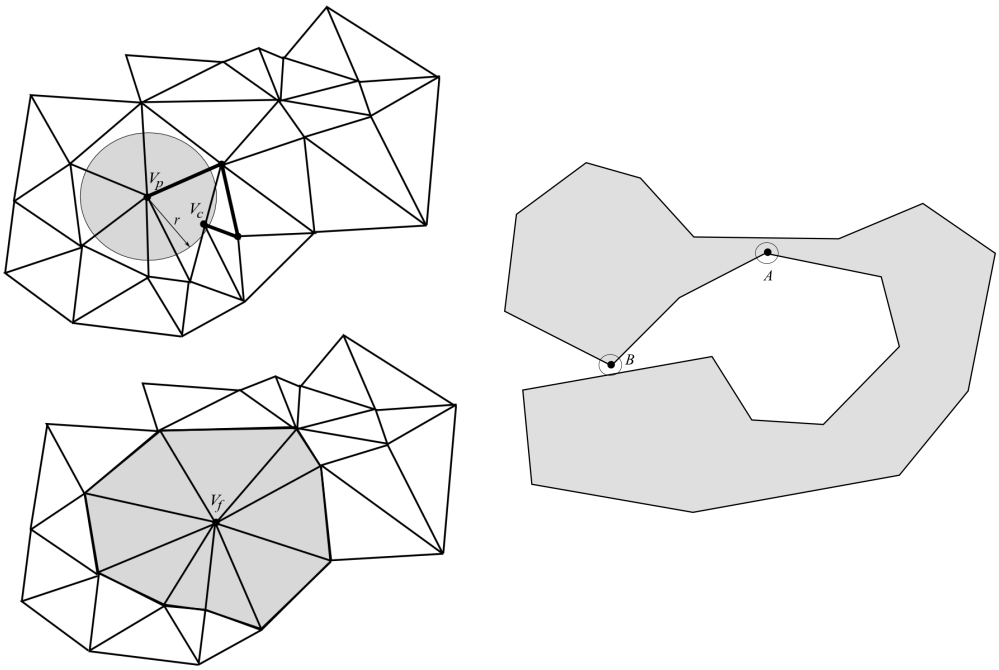


Figure A.2: Left: merging neighboring vertices, before (top) and after (bottom). Right: triangles at vertex A will self-intersect during erosions; those at B during dilations.

Appendix B

List of publications

Papers in refereed journals and book chapters

[**1**] Lam, R. and J.M.H. du Buf (2011), Retrieval of 3D Polygonal Objects Based on Multiresolution Signatures, Advances in Visual Computing, LNCS 6939 (II), Springer Berlin Heidelberg, 136-147. DOI: 10.1007/978-3-642-24031-7.

[**2**] Lam R. and du Buf, J.M.H. (2011) Using Mathematical Morphology for similarity search of 3D objects, Pattern Recognition and Image Analysis, LNCS 6669, Springer Berlin Heidelberg, 411-419. DOI: 10.1007/978-3-642-21257-4_51.

[**3**] Rodrigues, J., Lam R. and du Buf, J.M.H. (2012) Cortical 3D Face and Object Recognition Using 2D Projections, Int. J. Creat. Interfaces Comput. Graph., 3 (1), 45-62. DOI: 10.4018/jcicg.2012010104.

[**4**] Rodrigues, J.M.F., Martins, J., Lam R. and du Buf, J.M.H. (2012) Cortical Multiscale Line-Edge Disparity Model, Image Analysis and Recognition, LNCS 7325 (I), Springer Berlin Heidelberg, 296-303. DOI: 10.1007/978-3-642-31295-3_35.

[**5**] J.M.F. Rodrigues, R. Lam, K. Terzić and J.M.H. du Buf (2014), Face and Object Recognition Using Biological Features and Few Views, Contemporary Advancements in Information Technology Development in Dynamic Environments, Chap 4, 58-77. DOI: 10.4018/978-1-4666-6252-0.ch004.

[**6**] Sousa, L., Rodrigues, J.M.F.., Monteiro, J. Cardoso, P.J.S. and Lam, R.,(2016) GyGSLA: A Portable Glove System for Learning Sign Language Alphabet. In M. Antona and C. Stephanidis (Eds.): Universal Access in Human-Computer Interaction 2016, Part III, LNCS 9739, pp. 159-170. DOI: 10.1007/978-3-319-40238-3_16.

[**7**] Rodrigues, J.M.F., Alves, R., Sousa, L., Negrier, A., Cardoso, P.J.S., Monteiro, J.,

Felisberto, P., Figueiredo, M., Silva, B., Lam, R., Martins, J., Gomes, M. and Bica, P. (2016) PRHOLO - 360º Interactive Public Relations, Chapter 7 of the Handbook of Research on Human-Computer Interfaces, Developments, and Applications, pp. 166-191 , IGI Global. DOI: 10.4018/978-1-5225-0435-1.ch00.

Refereed conferences proceedings

[**1**] Lam R., Loke R. and du Buf, J.M.H. (2001) Smoothing and reduction of triangle meshes, Proc. 10th Portuguese Computer Graphics Meeting (10th EPCG), Lisbon, Portugal, pp. 97-107.

[**2**] Lam, R., Rodrigues, J. and du Buf, J.M.H. (2005) Artistic rendering of the visual cortex, Proc. 2º Workshop Luso-Galaico de Artes Digitais (ARTECH2005), Vila-Nova-Cerveira, Portugal, August 27, pp. 65-78.

[**3**] Rodrigues, J., Almeida, D., Nunes, S., Lam R. and du Buf, J.M.H. (2005) Building the what and where systems: multi-scale lines, edges and keypoints, Proc. Workshop on Active Vision VII (WAVVII2005), University of Reading, Reading, UK, September 12, pp. 8.

[**4**] Lam, R., Rodrigues, J. and du Buf, J.M.H. (2006) Looking through the eyes of the painter: from visual perception to non-photorealistic rendering, Proc. 14th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG2006), Plzen, Czech Republic, Jan 30 - Feb 3, pp. 147-154.

[**5**] Lam, R. and du Buf, J.M.H. (2009) Invariant Categorisation of Polygonal Objects using Multi-resolution Signatures, Proc. International Conference on Knowledge Discover and Information Retrieval (KDIR 20069), Madeira, Portugal, October 6-8, pp. 168-173.

[**6**] Rodrigues, J.M.F., Lam, R. and du Buf, J.M.H. (2011) Cortical 3D Face Recognition Framework, Proc. V Ibero-American Symposium in Computer Graphics (SIACG 2011), Faro, Portugal.

[**7**] Lam, R. and du Buf, J.M.H. (2011) Retrieval of 3D Polygonal Objects Based on Multiresolution Signatures, Proc. 7th International Symposium on Advances Visual Computing (ISVC 2011), Las Vegas 26-28.

[**8**] Lam, R. and du Buf, J.M.H. (2011) Retrieval of 3D Polygonal Objects Based on Multiresolution Signatures, Proc. 7th International Symposium on Advances Visual Computing (ISVC 2011), Las Vegas 26-28.

[**9**] Lam R. and du Buf, J.M.H. (2011) Using Mathematical Morphology for similarity search of 3D objects, Proc. of 5th Iberian Conf. on Pattern Recognition and Image Analysis, Spain Las Palmas, June 8-10, pp. 411-419.

[**10**] Rodrigues, J.M.F., Martins, J., Lam R. and du Buf, J.M.H. (2012) Cortical Multi-scale Line-Edge Disparity Model, Proc. 9th Int. Conf. Image Analysis and Recognition (ICIAR2012), Aveiro, Portugal, June 25-27, pp. 296-303.

[**11**] Martins, D., Lam, R., Rodrigues, J.M.F., Cardoso, P.J.S. and Serra, F. (2015) A Web Crawler Framework for Revenue Management, In Proc. 14th Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED '15), in Advances in Electrical and Computer Engineering, Tenerife, Canary Islands, Spain, 10-12 Jan, pp. 88-97. ISBN: 978-1-61804-279-8.

[**12**] Martins, D., Ramos, C.M.Q, Rodrigues, J.M.F., Cardoso, P.J.S., Lam, R. and Serra, F. (2015) Challenges in Building a Big Data Warehouse Applied to the Hotel Business Intelligence, In Proc. 6th Int. Conf. on Applied Informatics and Computing Theory (AICT15), in Recent Research in Applied Informatics, Salerno, Italy, 27-29 June, pp. 110-117. ISBN: 978-1-61804-313-9

[**13**] Ramos, C.M.Q., Martins, D., Serra, F., Lam, R., Cardoso, P.J.S., Correia, M.B. and Rodrigues, J.M.F. (2017) Framework for Hospitality Big Data Warehouse: the implementation of an efficient Hospitality Business Intelligence System, Accepted in Int. J. of Information Systems in the Service Sector (IJISSS) 9(2).

Papers to conference/journal in preparation

[**1**] Lam, R. and du Buf, J.M.H. (2016) Beyond classical signal processing: Fast multi-scale processing of Polygonal meshes.

[**2**] Lam, R. and du Buf, J.M.H. (2016) Categorization of 3D objects by dynamic mixing local and global features.

Bibliography

AIM@SHAPE, 2004. Advanced and innovative models and tools for the development of semantic-based systems for handling, acquiring, and processing knowledge embedded in multidimensional digital objects, project funded by the european commission under grant agreement 506766.
URL http://cordis.europa.eu/project/rcn/71204_en.html

Al-Osaimi, F. R., Bennamoun, M., Mian, A., Mar. 2008. Integration of local and global geometrical cues for 3D face recognition. Pattern Recogn. 41 (3), 1030–1040.
URL http://dx.doi.org/10.1016/j.patcog.2007.07.009

Alboul, L., Echeverria, G., 2005. Polyhedral gauss maps and curvature characterisation of triangle meshes. In: Proceedings of the 11th IMA International Conference on Mathematics of Surfaces. IMA'05. Springer-Verlag, Berlin, Heidelberg, pp. 14–33.
URL http://dx.doi.org/10.1007/11537908_2

Ankerst, M., Kastenmüller, G., Kriegel, H.-P., Seidl, T., 1999. 3D shape histograms for similarity search and classification in spatial databases. In: Güting, R., Papadias, D., Lochovsky, F. (Eds.), Advances in Spatial Databases. Vol. 1651 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 207–226.
URL http://dx.doi.org/10.1007/3-540-48482-5_14

Assfalg, J., Bertini, M., Bimbo, A., Pala, P., April 2007. Content-based retrieval of 3-D objects using spin image signatures. Multimedia, IEEE Transactions on 9 (3), 589–599.

Assfalg, J., Bimbo, A. D., Pala, P., 2006. Content-based retrieval of 3D models through curvature maps: a CBR approach exploiting media conversion. Multimedia Tools and Applications 31, 29–50.
URL http://dx.doi.org/10.1007/s11042-006-0034-2

Assfalg, J., Del Bimbo, A., Pala, P., 2004. Spin images for retrieval of 3D objects by local and global similarity. In: ICPR 2004. Proc. of the 17th International Conference on Pattern Recognition. Vol. 3. pp. 906–909.

Axenopoulos, A., Daras, P., Papadopoulos, G. T., Houstis, E., 2013. SP-Dock: Protein-protein docking using shape and physicochemical complementarity. IEEE/ACM Trans. Comput. Biology Bioinform. 10 (1), 135–150.

Bajaj, C. L., Coyle, E. J., Lin, K.-N., 1996. Arbitrary topology shape reconstruction from planar cross sections. Graphical Models and Image Processing 58 (6), 524 – 543.
URL http://www.sciencedirect.com/science/article/pii/S1077316996900441

Barra, V., Biasotti, S., 2013. Learning Kernels on Extended Reeb Graphs for 3D Shape Classification and Retrieval. Eurographics Association, Girona, Spain, pp. 25–32.
URL http://diglib.eg.org/EG/DL/WS/3DOR/3DOR13/025-032.pdf

Bengio, Y., 2009. Learning deep architectures for ai. Foundations and Trends in Machine Learning 2 (1), 1–127.
URL http://dx.doi.org/10.1561/2200000006

Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Levine, J. A., Sharf, A., Silva, C. T., 2014. State of the Art in Surface Reconstruction from Point Clouds. In: Lefebvre, S., Spagnuolo, M. (Eds.), Eurographics 2014 - State of the Art Reports. The Eurographics Association.

Berman, H., Kleywegt, G., Nakamura, H., Markley, J., 2013. How community has shaped the protein data bank. Structure 21 (9), 1485 – 1491.
URL http://www.sciencedirect.com/science/article/pii/S0969212613002608

Besl, P., 1995. Triangles as a primary representation. In: Hebert, M., Ponce, J., Boult, T., Gross, A. (Eds.), Object Representation in Computer Vision. Vol. 994 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 191–206.
URL http://dx.doi.org/10.1007/3-540-60477-4_14

Besl, P. J., Jain, R. C., 1985. Three-dimensional object recognition. ACM Comput. Surv. 17 (1), 75–145.
URL http://doi.acm.org/10.1145/4078.4081

Besl, P. J., McKay, N. D., Feb. 1992. A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. 14 (2), 239–256.
URL http://dx.doi.org/10.1109/34.121791

Biederman, I., 1987. Recognition-by-components: A theory of human image understanding. Psychological Review 94, 115–147.

Bimbo, A. D., Pala, P., Feb. 2006. Content-based retrieval of 3D models. ACM Trans. Multimedia Comput. Commun. Appl. 2 (1), 20–43.
URL http://doi.acm.org/10.1145/1126004.1126006

Bonde, U., Badrinarayanan, V., Cipolla, R., 2013. Multi scale shape index for 3D object recognition. In: Kuijper, A., Bredies, K., Pock, T., Bischof, H. (Eds.), Scale Space and Variational Methods in Computer Vision. Vol. 7893 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 306–318.
URL http://dx.doi.org/10.1007/978-3-642-38267-3_26

Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., Lévy, B., 2010. Polygon mesh processing. CRC press.

Botsch, M., Pauly, M., Rossl, C., Bischoff, S., Kobbelt, L., 2006. Geometric modeling based on triangle meshes. In: ACM SIGGRAPH 2006 Courses. SIGGRAPH '06. ACM, New York, NY, USA.
URL http://doi.acm.org/10.1145/1185657.1185839

Boyer, E., Berger, M.-O., Mar. 1997. 3D surface reconstruction using occluding contours. Int. J. Comput. Vision 22 (3), 219–233.
URL http://dx.doi.org/10.1023/A:1007978616082

Braude, I., Marker, J., Museth, K., Nissanov, J., Breen, D., Mar. 2007. Contour-based surface reconstruction using MPU implicit models. Graphical Models 69 (2), 139–157.
URL http://dx.doi.org/10.1016/j.gmod.2006.09.007

brickisland.net, 2015. DDG: Discrete differential geometry. http://brickisland.net/cs177/?p=67, accessed: 2015-07-19.

Bu, S., Han, P., Liu, Z., Han, J., Lin, H., 2015. Local deep feature learning framework for 3D shape. Computers & Graphics 46 (0), 117 – 129.
URL http://www.sciencedirect.com/science/article/pii/S0097849314000958

Bulthoff, H. H., 1990. Shape-from-X: psychophysics and computation, 235–246.
URL http://spiedigitallibrary.org/proceedings/resource/2/psisdg/1383/1/235_1

Bulthoff, H. H., Edelman, S. Y., Tarr, M. J., 1995. How are three-dimensional objects represented in the brain? Cerebral Cortex 5 (3), 247–260.
URL http://cercor.oxfordjournals.org/content/5/3/247.abstract

Burger, W., Burge, M., 2013. Principles of Digital Image Processing - Advanced Methods. Undergraduate Topics in Computer Science. Springer.

Burt, P. J., Adelson, E. H., 1987. Readings in computer vision: Issues, problems, principles, and paradigms. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Ch. The Laplacian Pyramid As a Compact Image Code, pp. 671–679.
URL http://dl.acm.org/citation.cfm?id=33517.33571

Bustos, B., Keim, D. A., Saupe, D., Schreck, T., Vranic, D., 2005. Feature-based similarity search in 3D object databases. ACM Computing Surveys 37 (4), 345–387.

Bustos, B., Schreck, T., Walter, M., Barrios, J. M., Schaefer, M., Keim, D., May 2012. Improving 3D similarity search by enhancing and combining 3D descriptors. Multimedia Tools Appl. 58 (1), 81–108.
URL http://dx.doi.org/10.1007/s11042-010-0689-6

Campbell, R. J., Flynn, P. J., 2001. A survey of free-form object representation and recognition techniques. Computer Vision and Image Understanding 81 (2), 166 – 210.
URL http://www.sciencedirect.com/science/article/pii/S1077314200908890

Cantzler, H., Fisher, R., 2001. Comparison of HK and SC curvature description methods. In: Proc. Third International Conference on 3-D Digital Imaging and Modeling. pp. 285–291.

Castellani, U., Cristani, M., Fantoni, S., Murino, V., 2008. Sparse points matching by combining 3D mesh saliency with statistical descriptors. Computer Graphics Forum 27 (2), 643–652.
URL http://dx.doi.org/10.1111/j.1467-8659.2008.01162.x

Chen, H., Bhanu, B., 2007. 3D free-form object recognition in range images using local surface patches. Pattern Recogn. Lett. 28 (10), 1252–1262.
URL http://dx.doi.org/10.1016/j.patrec.2007.02.009

Chen, X., Schmitt, F., 1992. Intrinsic surface properties from surface triangulation. In: Sandini, G. (Ed.), Computer Vision - ECCV'92. Vol. 588 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 739–743.
URL http://dx.doi.org/10.1007/3-540-55426-2_83

Chen, Y., Medioni, G., 1992. Object modelling by registration of multiple range images. Image Vision Comput. 10 (3), 145–155.
URL http://dx.doi.org/10.1016/0262-8856(92)90066-C

Cignoni, P., Corsini, M., Ranzuglia, G., April 2008. Meshlab: an open-source 3D mesh processing system. ERCIM News, 45–46.
URL http://vcg.isti.cnr.it/Publications/2008/CCR08

Clerc, M., Mallat, S., Apr. 2002. The texture gradient equation for recovering shape from texture. IEEE Trans. Pattern Anal. Mach. Intell. 24 (4), 536–549.
URL http://dx.doi.org/10.1109/34.993560

Curless, B., 1999. From range scans to 3D models. SIGGRAPH Comput. Graph. 33 (4), 38–41.
URL http://doi.acm.org/10.1145/345370.345399

Curran, T., Tanaka, J. W., Weiskopf, D. M., 2002. An electrophysiological comparison of visual categorization and recognition memory. Cognitive, Affective, and Behavioral Neuroscience 2, 201–216.

Dahl, G., Yu, D., Deng, L., Acero, A., jan. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. Audio, Speech, and Language Processing, IEEE Transactions on 20 (1), 30 –42.

Devernay, F., Faugeras, O. D., 1994. Computing differential properties of 3-D shapes from stereoscopic images without 3-D models. Proceedings CVPR '94., IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 208–213.

Díez, Y., Roure, F., Lladó, X., Salvi, J., 2015. A qualitative review on 3D coarse registration methods. ACM Comput. Surv. 47 (3), 45:1–45:36.
URL http://doi.acm.org/10.1145/2692160

do Carmo, M. P., 1976. Differential geometry of curves and surfaces. Prentice Hall.

Dorai, C., Jain, A., 1997. Cosmos-a representation scheme for 3D free-form objects. IEEE Trans. on Pattern Analysis and Machine Intelligence 19 (10), 1115–1130.

Dutagaci, H., Cheung, C., Godil, A., 2012. Evaluation of 3D interest point detection techniques via human-generated ground truth. The Visual Computer 28, 901–917.
URL http://dx.doi.org/10.1007/s00371-012-0746-4

Eck, M., Hoppe, H., 1996. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In: Proc. of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '96. ACM, New York, NY, USA, pp. 325–334.
URL http://doi.acm.org/10.1145/237170.237271

Fang, Y., Xie, J., Dai, G., Wang, M., Zhu, F., Xu, T., Wong, E., June 2015. 3D deep shape descriptor. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2319–2328.

Farin, G., 1996. Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, 4th Edition. Academic Press Professional, Inc., Orlando, FL, USA.

Faugeras, O., 1993. Three-dimensional Computer Vision: A Geometric Viewpoint. MIT Press, Cambridge, MA, USA.

Favaro, P., Mennucci, A., Soatto, S., 2003. Observing shape from defocused images. Int. J. Comput. Vision 52 (1), 25–43.
URL http://dx.doi.org/10.1023/A:1022366408068

Fehr, J., 2009. Local invariant features for 3D image analysis. Ph.D. thesis, Albert-Ludwigs-Universität Freiburg, Institut für Informatik.

Filipe, S., Alexandre, L. A., 2014. A comparative evaluation of 3D keypoint detectors in a RGB-D object dataset. In: Battiato, S., Braz, J. (Eds.), Proceedings of the 9th International Conference on Computer Vision Theory and Applications VISAPP, Volume 1, Lisbon, Portugal, 5-8 January, 2014. SciTePress, pp. 476–483.
URL http://dx.doi.org/10.5220/0004679904760483

Foley, J. D., van Dam, A., Feiner, S. K., Hughes, J. F., 1990. Computer Graphics: Principles and Practice (2nd Ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Foote, J., Jan. 1999. An overview of audio information retrieval. Multimedia Syst. 7 (1), 2–10.
URL http://dx.doi.org/10.1007/s005300050106

Forsyth, D., Mundy, J., Zisserman, A., Coelho, C., Heller, A., Rothwell, C., 1991. Invariant descriptors for 3D object recognition and pose. Pattern Analysis and Machine Intelligence, IEEE Transactions on 13 (10), 971–991.

Forsyth, D. A., Ponce, J., 2012. Computer Vision: A Modern Approach, 2nd Edition. Prentice Hall Professional Technical Reference.

Gal, R., Cohen-Or, D., Jan. 2006. Salient geometric features for partial shape matching and similarity. ACM Trans. Graph. 25 (1), 130–150.
URL http://doi.acm.org/10.1145/1122501.1122507

Garland, M., Heckbert, P. S., 1997. Surface simplification using quadric error metrics. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '97. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 209–216.
URL http://dx.doi.org/10.1145/258734.258849

Gatzke, T., Grimm, C., Garland, M., Zelinka, S., June 2005. Curvature maps for local shape comparison. In: Shape Modeling and Applications, 2005 International Conference. pp. 244–253.

Geng, J., Jun 2011. Structured-light 3D surface imaging: a tutorial. Adv. Opt. Photon. 3 (2), 128–160.
URL http://aop.osa.org/abstract.cfm?URI=aop-3-2-128

Ghosh, P. K., Haralick, R. M., 1993. Mathematical morphology of three-dimensional objects: the slope diagrammatic approach.
URL http://dx.doi.org/10.1117/12.164987

Glendinning, R. H., Herbert, R. A., 2003. Shape classification using smooth principal components. Pattern Recognition Letters Volume 24(12), 2021–2030.

Godil, A., Wagan, A. I., 2011. Salient local 3D features for 3D shape retrieval. In: 3D Image Processing (3DIP) and Applications II. SPIE Electronic Imaging. International Society for Optics and Photonics, pp. 78640S–78640S.

Goshtasby, A. A., 2005. 2-D and 3-D Image Registration: For Medical, Remote Sensing, and Industrial Applications. Wiley-Interscience.

Gray, A., Abbena, E., Salamon, S., 2006. Modern Differential Geometry of Curves and Surfaces with Mathematica, Third Edition (Studies in Advanced Mathematics). Chapman & Hall/CRC.

Hamann, B., 1993. Curvature approximation for triangulated surfaces. In: Farin, G., Noltemeier, H., Hagen, H., Knödel, W. (Eds.), Geometric Modelling. Vol. 8 of Computing Supplementum. Springer Vienna, pp. 139–153.
URL http://dx.doi.org/10.1007/978-3-7091-6916-2_10

Hamel, P., Eck, D., 2010. Learning features from music audio with deep belief networks. In: Downie, J. S., Veltkamp, R. C. (Eds.), ISMIR 2010. International Society for Music Information Retrieval, pp. 339–344.

Hartley, R., Zisserman, A., 2003. Multiple View Geometry in Computer Vision, 2nd Edition. Cambridge University Press, New York, NY, USA.

Heider, P., Pierre-Pierre, A., Li, R., Grimm, C., 2011. Local shape descriptors, a survey and evaluation. In: Proceedings of the 4th Eurographics Conference on 3D Object Retrieval. 3DOR '11. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 49–56.
URL http://dx.doi.org/10.2312/3DOR/3DOR11/049-056

Hiebert-Treuer, B., Nashashibi, S. A., Scharstein, D., 2006. Middlebury stereo vision - datasets. Accessed: January 29, 2015.
URL http://vision.middlebury.edu/stereo/data/

Hinton, G. E., Osindero, S., Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. Neural Comput. 18 (7), 1527–1554.
URL http://dx.doi.org/10.1162/neco.2006.18.7.1527

Hinton, G. E., Salakhutdinov, R. R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507.

Hoppe, H., 1996. Progressive meshes. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '96. ACM, New York, NY, USA, pp. 99–108.
URL http://doi.acm.org/10.1145/237170.237216

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W., 1993. Mesh optimization. In: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '93. ACM, New York, NY, USA, pp. 19–26.
URL http://doi.acm.org/10.1145/166117.166119

Horn, B. K., 1970. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. Tech. rep., Cambridge, MA, USA.

Hubeli, A., Gross, M., 2000. A survey of surface representations for geometric modeling. CS Technical Report 335, Swiss Federal Institute of Technology Zurich, Institute of Scientific Computing.

Huertas, A., Medioni, G., 1986. Detection of intensity changes with subpixel accuracy using laplacian-gaussian masks. IEEE Trans. Pattern Anal. Mach. Intell. 8 (5), 651–664.
URL http://dx.doi.org/10.1109/TPAMI.1986.4767838

Hung, Y.-P., Chen, C.-S., Hsieh, I.-B., Fuh, C.-S., 1999. Reconstruction of complete 3D object model from multiview range images. Proc. SPIE 3640, 138–145.
URL http://dx.doi.org/10.1117/12.341055

Huttenlocher, D., Klanderman, G., Rucklidge, W., 1993. Comparing images using the hausdorff distance. Pattern Analysis and Machine Intelligence, IEEE Transactions on 15 (9), 850–863.

Itti, L., Koch, C., Niebur, E., Nov 1998. A model of saliency-based visual attention for rapid scene analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on 20 (11), 1254–1259.

Jackway, P. T., 1995. Morphological scale-space with application to three-dimensional object recognition. Ph.D. thesis, Queensland University of Technology (Australia).

Johnson, A., August 1997. Spin-images: A representation for 3-D surface matching. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Johnson, A. E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell. 21 (5), 433–449.
URL http://dx.doi.org/10.1109/34.765655

Kalogerakis, E., Hertzmann, A., Singh, K., 2010. Learning 3D mesh segmentation and labeling. ACM Trans. Graph. 29 (4), 102:1–102:12.
URL http://doi.acm.org/10.1145/1778765.1778839

Kazhdan, M., Funkhouser, T., Rusinkiewicz, S., 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. In: Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing. SGP '03. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 156–164.
URL http://dl.acm.org/citation.cfm?id=882370.882392

Kazhdan, M. M., 2004. Shape representations and algorithms for 3D model retrieval. Ph.D. thesis, Princeton University, Department of Computer Science.

Kazmi, I., You, L., Zhang, J. J., 2013. A survey of 2D and 3D shape descriptors. In: Computer Graphics, Imaging and Visualization (CGIV), 2013 10th International Conference. pp. 1–10.

Kendall, D. G., 1977. The diffusion of shape. In: Advances in Applied Probability. Vol. 9. pp. 428–430.
URL http://www.jstor.org/stable/1426091

Kobbelt, L., 2003. Freeform shape representations for efficient geometry processing. Computer Graphics Forum 22 (3), xviii–xviii.
URL http://dx.doi.org/10.1111/1467-8659.00667

Koch, C., Ullman, S., Jan. 1985. Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry. Human Neurobiology 4, 219–227.

Koenderink, J., 1984. The structure of images. Biological Cybernetics 50 (5), 363–370.
URL http://dx.doi.org/10.1007/BF00336961

Koenderink, J. J., van Doorn, A. J., 1992. Surface shape and curvature scales. Image Vision Comput. 10 (8), 557–565.
URL http://dx.doi.org/10.1016/0262-8856(92)90076-F

Lam, R., du Buf, J., 2011a. Retrieval of 3D polygonal objects based on multiresolution signatures. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Wang, S., Kyungnam, K., Benes, B., Moreland, K., Borst, C., DiVerdi, S., Yi-Jen, C., Ming, J. (Eds.), Advances in Visual Computing. Vol. 6939 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 136–147.
URL http://dx.doi.org/10.1007/978-3-642-24031-7_14

Lam, R., du Buf, J. M. H., 2009. Invariant categorisation of polygonal objects using multi-resolution signatures. In: Proc. 1st Inter. Conference on Knowledge Discovery and InformationRetrieval (KDIR). pp. 168–173.

Lam, R., du Buf, J. M. H., 2011b. Using mathematical morphology for similarity search of 3D objects. In: Proc. of Fifth Iberian Conference on Pattern Recognition and Image Analysis. LNCS 6669. Springer, pp. 411–419.

Lam, R., Loke, R., du Buf, H., 2001. Smoothing and reduction of triangle meshes. Proc. 10th Portuguese Computer Graphics Meeting, 97–107.

Lee, C. H., Varshney, A., Jacobs, D. W., 2005a. Mesh saliency. ACM Trans. Graph. 24 (3), 659–666.
URL http://doi.acm.org/10.1145/1073204.1073244

Lee, J. R. J., Smith, M. L., Smith, L. N., Midha, P. S., 2005b. A mathematical morphology approach to image based 3D particle shape analysis. Machine Vision and Applications 16 (5), 282–288.
URL http://dx.doi.org/10.1007/s00138-005-0181-x

Lee, T. S., Mumford, D., Romero, R., Lamme, V. A., 1998. The role of the primary visual cortex in higher level vision. Vision Research 38 (15-16), 2429 – 2454.
URL http://www.sciencedirect.com/science/article/pii/S0042698997004641

Li, X., Godil, A., Wagan, A., 2008. 3D part identification based on local shape descriptors. In: Proc. of the 8th Workshop on Performance Metrics for Intelligent Systems. PerMIS '08. ACM, New York, NY, USA, pp. 162–166.
URL http://doi.acm.org/10.1145/1774674.1774700

Li, X., Guskov, I., 2005. Multi-scale features for approximate alignment of point-based surfaces. In: Proceedings of the Third Eurographics Symposium on Geometry Processing. SGP '05. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland.
URL http://dl.acm.org/citation.cfm?id=1281920.1281955

Li, X., Iyengar, S. S., 2014. On computing mapping of 3D objects: A survey. ACM Comput. Surv. 47 (2), 34:1–34:45.
URL http://doi.acm.org/10.1145/2668020

Lim, I. S., Leek, E. C., 2012. Curvature and the visual perception of shape: Theory on information along object boundaries and the minima rule revisited. Psychological Review 119, 668–677.

Lindeberg, T., 1996. Scale-space theory : A framework for handling image structures at multiple scales. In: Proc. CERN School of Computing, Egmond aan Zee, The Netherlands, 8-21 September, 1996. pp. 27–38, qC 20111101.

Lipschutz, M., 1969. Theory and Problems of Differential Geometry. Schaum's Outline Series.

Liter, J. C., Bülthoff, H. H., 1998. An introduction to object recognition. Zeitschrift für Naturforschung C 53, 610–621.

Liu, S., Kang, K., Tarel, J.-P., Cooper, D. B., Jan. 2008. Free-form object reconstruction from silhouettes, occluding edges and texture edges: A unified and robust operator based on duality. IEEE Trans. Pattern Anal. Mach. Intell. 30 (1), 131–146.
URL http://dx.doi.org/10.1109/TPAMI.2007.1143

Loke, R. E., 2006. Progressive visualization of incomplete sonar-data sets: from sea-bottom interpolation and segmentation to geometry extraction. Ph.D. thesis, Delft University of Technology.

Loncaric, S., 1998. A survey of shape analysis techniques. Pattern Recognition 31 (8), 983 – 1001.
URL http://www.sciencedirect.com/science/article/pii/S0031202397001222

Lorensen, W. E., Cline, H. E., 1987. Marching cubes: A high resolution 3D surface construction algorithm. SIGGRAPH Comput. Graph. 21 (4), 163–169.
URL http://doi.acm.org/10.1145/37402.37422

Lowe, D., 1999. Object recognition from local scale-invariant features. In: Proc. of the Seventh IEEE Int. Conference on Computer Vision. Vol. 2. pp. 1150–1157.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2), 91–110.
URL http://dx.doi.org/10.1023/B%3AVISI.0000029664.99615.94

Luffel, M., Gurung, T., Lindstrom, P., Rossignac, J., 2014. Grouper: A compact, streamable triangle mesh data structure. IEEE Trans. Vis. Comput. Graph. 20 (1), 84–98.

Ma, Y., Soatto, S., Kosecka, J., Sastry, S. S., 2003. An Invitation to 3-D Vision: From Images to Geometric Models. SpringerVerlag.

Mahmoudi, M., Sapiro, G., Jan. 2009. Three-dimensional point cloud recognition via distributions of geometric distances. Graph. Models 71 (1), 22–31.
URL http://dx.doi.org/10.1016/j.gmod.2008.10.002

Malik, J., Rosenholtz, R., 1997. Computing local surface orientation and shape from texture for curved surfaces. Int. J. Comput. Vision 23 (2), 149–168.
URL http://dx.doi.org/10.1023/A:1007958829620

Marr, D., Hildreth, E., 1980. Theory of edge detection. Proceedings of the Royal Society of London B: Biological Sciences 207 (1167), 187–217.
URL http://rspb.royalsocietypublishing.org/content/207/1167/187

Marsh, D., 2005. Applied Geometry for Computer Graphics and CAD, 2nd Edition. Springer-Verlag, London, UK.

Marturi, N., Dembélé, S., Piat, N., Jan. 2013. Depth and Shape Estimation from Focus in Scanning Electron Microscope for Micromanipulation. In: International Conference on Control, Automation, Robotics & Embedded Systems, CARE'13. India, pp. 1–6.
URL https://hal.archives-ouvertes.fr/hal-00876253

Matheron, G., 1975. Random sets and integral geometry. John Wiley & Sons, New York.

Meyer, M., Desbrun, M., Schröder, P., Barr, A., 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In: Hege, H.-C., Polthier, K. (Eds.), Visualization and Mathematics III. Mathematics and Visualization. Springer Berlin Heidelberg, pp. 35–57.
URL http://dx.doi.org/10.1007/978-3-662-05105-4_2

Microsoft, D., 2009. Kinect for windows, Microsoft. Accessed: January 13, 2015.
URL https://developer.microsoft.com/en-us/windows/kinect

Mortara, M., Patané;, G., Spagnuolo, M., Falcidieno, B., Rossignac, J., Oct. 2003. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. Algorithmica 38 (1), 227–248.
URL http://dx.doi.org/10.1007/s00453-003-1051-4

Morvan, J., Thibert, B., 2002a. On the approximation of a smooth surface with a triangulated mesh. Computational Geometry 23 (3), 337 – 352.
URL http://www.sciencedirect.com/science/article/pii/S0925772102000974

Morvan, J., Thibert, B., 2002b. Smooth surface and triangular mesh: Comparison of the area, the normals and the unfolding. In: Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications. SMA '02. ACM, New York, NY, USA, pp. 147–158.
URL http://doi.acm.org/10.1145/566282.566306

Mousa, M., Chaine, R., Akkouche, S., Galin, E., Oct 2007. Efficient spherical harmonics representation of 3D objects. In: 15th Pacific Conf. on Computer Graphics and Applications. PG '07. pp. 248–255.

Mumford, D., Fogarty, J., Kirwan, F., Nov. 1994. Geometric Invariant Theory, 3rd Edition. Springer.

Nayar, S. K., Nakagawa, Y., Aug. 1994. Shape from focus. IEEE Trans. Pattern Anal. Mach. Intell. 16 (8), 824–831.
URL http://dx.doi.org/10.1109/34.308479

Newman, T. S., Yi, H., 2006. A survey of the marching cubes algorithm. Computers & Graphics 30 (5), 854 – 879.
URL http://www.sciencedirect.com/science/article/pii/S0097849306001336

Niebur, E., Koch, C., 1995. Control of selective visual attention: Modeling the where pathway. In: Touretzky, D. S., Mozer, M., Hasselmo, M. E. (Eds.), NIPS. MIT Press, pp. 802–808.

Novatnack, J., Nishino, K., Oct 2007. Scale-dependent 3D geometric features. In: IEEE 11th Int. Conf. on Computer Vision, ICCV 2007. pp. 1–8.

Novotni, M., Klein, R., 2003. 3D zernike descriptors for content based shape retrieval. In: Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications. SM '03. ACM, New York, NY, USA, pp. 216–225.
URL http://doi.acm.org/10.1145/781606.781639

Ohbuchi, R., Otagiri, T., Ibato, M., Takei, T., 2002. Shape-similarity search of three-dimensional models using parameterized statistics. In: Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on. pp. 265–274.

Oliva, A., Torralba, A., 2006. Building the gist of a scene: The role of global image features in recognition. Visual Perception, Progress in Brain Research 155, 23–36.

Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D., 2002. Shape distributions. ACM Trans. Graph. 21 (4), 807–832.
URL http://doi.acm.org/10.1145/571647.571648

Papadakis, P., Pratikakis, I., Perantonis, S., Theoharis, T., Sep. 2007. Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. Pattern Recogn. 40 (9), 2437–2452.
URL http://dx.doi.org/10.1016/j.patcog.2006.12.026

Paquet, E., Rioux, M., Murching, A., Naveen, T., Tabatabai, A., 2000. Description of shape information for 2-D and 3-D objects. Signal Processing: Image Communication 16 (1-2), 103–122.
URL http://www.sciencedirect.com/science/article/pii/S0923596500000205

Pears, N., Liu, Y., Bunting, P. (Eds.), 2012. 3D Imaging, Analysis and Applications. Springer London.

Pele, O., Werman, M., 2010. The quadratic-chi histogram distance family. In: Daniilidis, K., Maragos, P., Paragios, N. (Eds.), Computer Vision - ECCV 2010. Vol. 6312 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 749–762.
URL http://dx.doi.org/10.1007/978-3-642-15552-9_54

Phillips, Flip ; Todd, J. T., 1996. Perception of local three-dimensional shape. Journal of Experimental Psychology: Human Perception and Performance Vol. 22 (4), 930–944.

Polthier, K., Schmies, M., 1998. Straightest Geodesics on Polyhedral Surfaces. Springer Berlin Heidelberg.
URL http://dx.doi.org/10.1007/978-3-662-03567-2_11

Pottmann, H., Wallner, J., Huang, Q.-X., Yang, Y.-L., 2009. Integral invariants for robust geometry processing. Computer Aided Geometric Design 26 (1), 37 – 60.
URL http://www.sciencedirect.com/science/article/pii/S0167839608000095

Qiao, F., Wang, C., Zhang, X., Wang, H., 2013. Large scale near-duplicate celebrity web images retrieval using visual and textual features. The Scientific World Journal.

Qin, F.-w., Li, L.-y., Gao, S.-m., Yang, X.-l., Chen, X., 2014. A deep learning approach to the classification of 3D cad models. Journal of Zhejiang University, SCIENCE C (Computers & Electronics) 15 (2), 91–106.
URL http://dx.doi.org/10.1631/jzus.C1300185

Restrepo, M., Mayer, B., Ulusoy, A., Mundy, J., 2012. Characterization of 3-D volumetric probabilistic scenes for object recognition. Selected Topics in Signal Processing, IEEE Journal of 6 (5), 522–537.

Rodrigues, J., 2008. Integrated multi-scale architecture of the cortex with application to computer vision. Ph.D. thesis, Universidade do Algarve.

Rodrigues, J., Lam, R., du Buf, H., Jan. 2012a. Cortical 3D face and object recognition using 2D projections. Int. J. Creat. Interfaces Comput. Graph. 3 (1), 45–62.
URL http://dx.doi.org/10.4018/jcicg.2012010104

Rodrigues, J., Lam, R., du Buf, K. T. J., 2014. Face and Object Recognition Using Biological Features and Few Views. IGI Global, pp. 58–77.

Rodrigues, J. M. F., Martins, J. A., Lam, R., du Buf, J. M. H., 2012b. Cortical multiscale line-edge disparity model. In: Proceedings of the 9th International Conference on Image Analysis and Recognition - Volume Part I. ICIAR'12. Springer-Verlag, Berlin, Heidelberg, pp. 296–303.
URL http://dx.doi.org/10.1007/978-3-642-31295-3_35

Salti, S., Tombari, F., Di Stefano, L., 2011. A performance evaluation of 3D keypoint detectors. In: 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on. pp. 236–243.

Scharstein, D., Szeliski, R., 1998. Stereo matching with nonlinear diffusion. International Journal of Computer Vision 28 (2), 155–174.
URL http://dblp.uni-trier.de/db/journals/ijcv/ijcv28.html#ScharsteinS98

Scharstein, D., Szeliski, R., Apr. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vision 47 (1-3), 7–42.
URL http://dx.doi.org/10.1023/A:1014573219977

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks 61 (0), 85 – 117.
URL http://www.sciencedirect.com/science/article/pii/S0893608014002135

Schroth, G., Hilsenbeck, S., Huitl, R., Schweiger, F., Steinbach, E., Dec. 2011. Exploiting text-related features for content-based image retrieval. In: IEEE International Symposium on Multimedia (ISM).

Serna, S. P., Stork, A., Fellner, D. W., 2011. Considerations toward a dynamic mesh data structure. In: SIGRAD 2011. Linköping Electronic Conference Proceedings; 65. SIGRAD, Linköping University Electronic Press, Linköping, pp. 83–90.

Serra, J., 1982. Image Analysis and Mathematical Morphology. Academic Press: New York.

Serra, J., 1986. Introduction to mathematical morphology. Comput. Vision, Graphics and Image Processing 35 (3), 283–305.

Shih, F., 1 1991. Object representation and recognition using mathematical morphology model. Journal of Systems Integration 1 (123), 235–256.

Sijbers, J., Dyck, D. V., 2002. Efficient algorithm for the computation of 3D Fourier descriptors. 3D Data Processing Visualization and Transmission, International Symposium on, 640–643.

Sipiran, I., Bustos, B., 2010. A robust 3D interest points detector based on Harris operator. In: Proceedings of the 3rd Eurographics Conference on 3D Object Retrieval. 3DOR '10. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 7–14.
URL http://dx.doi.org/10.2312/3DOR/3DOR10/007-014

Socher, R., Huval, B., Bath, B., Manning, C. D., Ng, A. Y., 2012. Convolutional-recursive deep learning for 3D object classification. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp. 656–664.

Soille, P., Rivest, J.-F., 1996. On the validity of fractal dimension measurements in image analysis. Journal of Visual Communication and Image Representation 7 (3), 217 – 229.

Spivak, M., 1999. A Comprehensive Introduction to Differential Geometry, 3rd Edition. Vol. 2. Publish of Perish, Inc., Houston, TX.

Structure Sensor, 2014. Structure sensor, occipital.com. Accessed: January 13, 2015.
URL http://structure.io/

Sun, J., Ovsjanikov, M., Guibas, L., 2009. A concise and provably informative multi-scale signature based on heat diffusion. In: Proceedings of the Symposium on Geometry Processing. SGP '09. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 1383–1392.
URL http://dl.acm.org/citation.cfm?id=1735603.1735621

Sun, Y., Paik, J. K., Koschan, A., Page, D. L., Abidi, M. A., 2002. Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing. In: ICIP (3). pp. 825–828.

Szeliski, R., 2010. Computer Vision: Algorithms and Applications, 1st Edition. Springer-Verlag New York, Inc., New York, NY, USA.

Tam, G. K. L., Cheng, Z.-Q., Lai, Y.-K., Langbein, F. C., Liu, Y., Marshall, D., Martin, R. R., Sun, X.-F., Rosin, P. L., 2013. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. IEEE Transactions on Visualization and Computer Graphics 19 (7), 1199–1217.

Tang, S., Godil, A., 2012. An evaluation of local shape descriptors for 3D shape retrieval. Vol. 8290. pp. 82900N–82900N–15.
URL http://dx.doi.org/10.1117/12.912153

Tangelder, J. W., Veltkamp, R. C., 2008. A survey of content based 3D shape retrieval methods. Multimedia Tools Appl. 39 (3), 441–471.

Tarr, M. J., Bülthoff, H. H., 1995. Is human object recognition better described by geon-structural-descriptions or by multiple views? Journal of Experimental Psychology: Human Perception and Performance 21 (6), 1494–1505.

Taubin, G., 1995. Estimating the tensor of curvature of a surface from a polyhedral approximation. In: Proceedings of the Fifth International Conference on Computer Vision. ICCV '95. IEEE Computer Society, Washington, DC, USA, pp. 902–.
URL http://dl.acm.org/citation.cfm?id=839277.840020

Trucco, E., Verri, A., 1998. Introductory Techniques for 3-D Computer Vision. Prentice Hall PTR, Upper Saddle River, NJ, USA.

Tulving, E., Thomson, D. M., 1973. Encoding specificity and retrieval processes in episodic memory. Psychological Review 80 (5), 352–373.

Turk, G., Levoy, M., 1994. Zippered polygon meshes from range images. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '94. ACM, New York, NY, USA, pp. 311–318.
URL http://doi.acm.org/10.1145/192161.192241

Umeyama, S., Apr. 1991. Least-squares estimation of transformation parameters between two point patterns. IEEE Trans. Pattern Anal. Mach. Intell. 13 (4), 376–380.
URL http://dx.doi.org/10.1109/34.88573

Velho, L., Gomes, J., de Figueiredo, L. H., 2002. Implicit Objects in Computer Graphics. Springer-Verlag New York, Inc.

Veltkamp, R. C., 2001. Shape matching: Similarity measures and algorithms. In: Proceedings of the International Conference on Shape Modeling & Applications. SMI '01. IEEE Computer Society, Washington, DC, USA, pp. 188–.
URL http://dl.acm.org/citation.cfm?id=882486.884078

Veltkamp, R. C., Giezeman, G.-J., Bast, H., Baumbach, T., Furuya, T., Giesen, J., Godil, A., Lian, Z., Ohbuchi, R., Saleem., W., 2010. SHREC'10 track: Large scale retrieval. In: Proc. of the Eurographics/ACM SIGGRAPH Symp. on 3D Object Retrieval. pp. 63–69.

Visionair, 2011. Visionair, project funded by the european commission under grant agreement 262044.
URL http://www.infra-visionair.eu/index.php

Vogel, J., Schwaninger, A., Wallraven, C., Bülthoff, H. H., Nov. 2007. Categorization of natural scenes: Local versus global information and the role of color. ACM Trans. Appl. Percept. 4 (3).
URL http://doi.acm.org/10.1145/1278387.1278393

Vranic, D., 2004. 3D model retrieval. Ph.D. thesis, University of Leipzig.

Vranic, D. V., Saupe, D., Sep. 2001. 3D Shape Descriptor Based on 3D Fourier Transform. In: Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001) (editor K. Fazekas). Budapest, Hungary, pp. 271–274.

Walter, N., Aubreton, O., Laligant, O., 2008. Salient point characterization for low resolution meshes. In: Image Processing, ICIP 2008. 15th IEEE International Conference on. pp. 1512–1515.

Wang, X., Wang, Y., 2014. Improving content-based and hybrid music recommendation using deep learning. In: Proceedings of the ACM International Conference on Multimedia. MM '14. ACM, New York, NY, USA, pp. 627–636.
URL http://doi.acm.org/10.1145/2647868.2654940

Watt, A., 1993. 3D Computer Graphics, 2nd Edition. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Weinshall, D., 1993. Model-based invariants for 3-D vision. International Journal of Computer Vision 10 (1), 27–42.
URL http://dx.doi.org/10.1007/BF01440845

Weiss, I., 1993. Geometric invariants and object recognition. International Journal of Computer 11263on 10 (3), 207–231.
URL http://dx.doi.org/10.1007/BF01539536

Wildes, R., 1991. Direct recovery of three-dimensional scene geometry from binocular stereo disparity. IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (8), 761–774.

Woodham, R. J., 1989. Shape from shading. MIT Press, Cambridge, MA, USA, Ch. Photometric Method for Determining Surface Orientation from Multiple Images, pp. 513–531.
URL http://dl.acm.org/citation.cfm?id=93871.93888

Wu, J., Kobbelt, L., 2003. Piecewise Linear Approximation of Signed Distance Fields. In: Vision Modeling and Visualization. pp. 513–520.

Wu, Z., Song, S., Khosla, A., Zhang, L., Tang, X., Xiao, J., 2015. 3D shapenets: A deep representation for volumetric shape modeling. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, USA.

Xie, J., Fang, Y., Zhu, F., Wong, E., June 2015. Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1275–1283.

Yan, D.-M., Wang, W., Liu, Y., Yang, Z., Nov. 2012. Variational mesh segmentation via quadric surface fitting. Comput. Aided Des. 44 (11), 1072–1082.
URL http://dx.doi.org/10.1016/j.cad.2012.04.005

Yang, Y.-L., Shen, C.-H., 2012. Multi-scale salient feature extraction on mesh models. In: Hu, S.-M., Martin, R. (Eds.), Computational Visual Media. Vol. 7633 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 122–129.
URL http://dx.doi.org/10.1007/978-3-642-34263-9_16

Yao, B., Li, Z., Ding, M., Chen, M., 2016. Three-dimensional protein model similarity analysis based on salient shape index. BMC Bioinformatics 17 (1), 1–10.
URL http://dx.doi.org/10.1186/s12859-016-0983-z

Zaharescu, A., Boyer, E., Horaud, R., 2007. Transformesh: a topology-adaptive mesh-based approach to surface evolution. In: Proc. of Eighth Asian Conf. on Computer Vision. Vol. 2 of LNCS 4844. Springer, pp. 166–175.

Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R., 2009. Surface feature detection and description with applications to mesh matching. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. pp. 373–380.

Zhang, C., Chen, T., 2001. Efficient feature extraction for 2D/3D objects in mesh representation. In: Image Processing, 2001. Proceedings. 2001 International Conference on. Vol. 3. pp. 935–938.

Zhang, R., Tsai, P.-S., Cryer, J. E., Shah, M., 1999. Shape from shading: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 21 (8), 690–706.
URL `http://dx.doi.org/10.1109/34.784284`

Zhong, Y., 2009. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. pp. 689–696.

Zhou, C., Güney, F., Wang, Y., Geiger, A., Dec. 2015. Exploiting object similarity in 3D reconstruction. In: International Conference on Computer Vision (ICCV).

Zitová, B., Flusser, J., 2003. Image registration methods: a survey. Image and Vision Computing 21, 977–1000.

Zorin, D., Schröder, P., 2000. Subdivision for Modeling and Animation. Tech. rep., SIGGRAPH 2000, course Notes.